



# Servicios Web

## Introducción

# Visión General Servicios Web



- ¿Qué son los Servicios Web ?
  - Un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios.

# Visión General Servicios Web



- Un servicio web es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.
- La interoperabilidad se consigue mediante la adopción de estándares abiertos.

# Visión General Servicios Web



- Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

# Visión General Servicios Web



- Ventajas de los servicios Web
  - Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
  - Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
  - Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad

# Visión General Servicios Web



- Ventajas de los servicios web
  - Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
  - Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones. Es el futuro de la aplicaciones

# Visión General Servicios Web



- Inconvenientes
  - Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA
  - Su rendimiento es bajo si se compara con otros modelos. Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.
  - Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall

# Visión General Servicios Web



- La principal razón para usar servicios Web
  - Es que se basan en HTTP sobre TCP en el puerto 80. Dado que las organizaciones protegen sus redes mediante firewalls - que filtran y bloquean gran parte del tráfico de Internet-, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web se enrutan por este puerto, por la simple razón de que no resultan bloqueados.
  - Otra razón es que, antes de que existiera SOAP, no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red.

# Visión General Servicios Web



- Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más acusada.

# Visión General Servicios Web



- ¿ Cómo funcionan ?
  - Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.

# Visión General Servicios Web

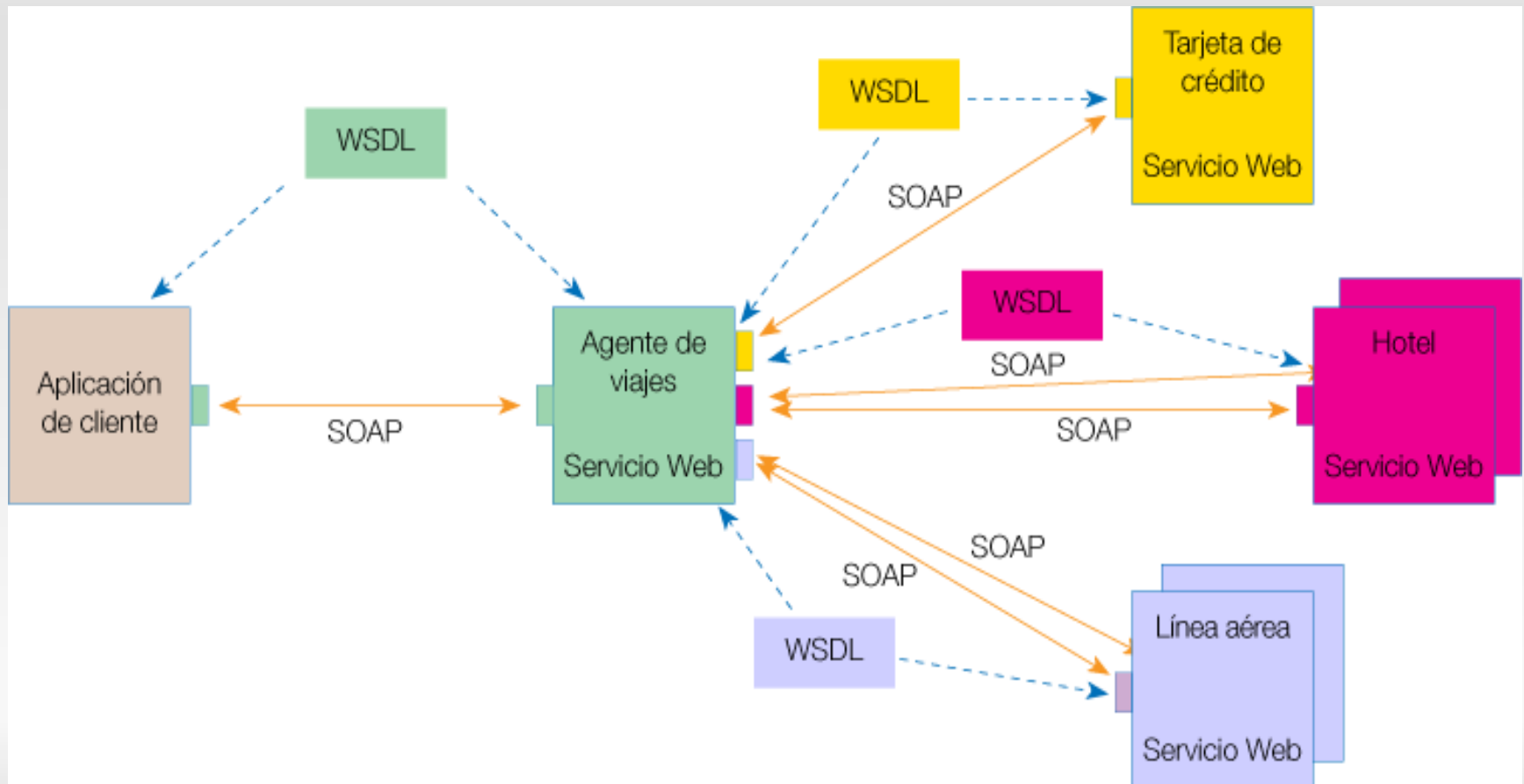


- ¿Para qué sirven?
  - Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario.
  - Es necesario una arquitectura estándar para proporcionar estos servicios

# Visión General Servicios Web



- Gráfico



# Visión General Servicios Web



- Según el grafico anterior
  - Un usuario solicita la información a una agencia de viajes que ofrece sus servicios por internet. La agencia de viajes ofrece a su cliente la información requerida

# Visión General Servicios Web



- ¿ Como se proporciona es información?
  - Para proporcionar dicha información la agencia de viajes solicita información a otros recursos (servicios Web)
  - Lo que convierte a la agencia de viajes a su vez en cliente de otros servicios web que le van a proporcionar información sobre el hotel y la tarjeta de crédito

# Visión General Servicios Web



- Tecnologías Implicadas
  - Las tecnologías que hacen posible esta circulación son:
    - SOAP (Protocolo simple de acceso a Objetos)
    - WSDL (Lenguaje de descripción de Servicios)

# Visión General Servicios Web



- ¿ Qué es Soap ?
  - Se trata de un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de HTTP , SMTP
  - Soap especifica el formato de los mensajes

# Visión General Servicios Web



- Formato de los mensajes Soap
  - Sobre (envelope)
  - Cabecera (header)
  - Cuerpo (body)

# Visión General Servicios Web



- Envelope
  - Header
    - Bloque header cliente banco
  - Body
    - Body sub-elemento transferencia

# Visión General Servicios Web



- Definición detallada
  - SOAP es un protocolo de mensajería XML extensible que forma la base de los Servicios Web. SOAP proporciona un mecanismo simple y consistente que permite a una aplicación enviar mensajes XML a otra aplicación. Un mensaje SOAP es una transmisión de una vía desde un emisor SOAP a un receptor SOAP, y cualquier aplicación puede participar en este intercambio como emisor o receptor. Los mensajes SOAP se pueden combinar para soportar muchos comportamientos de comunicación, incluyendo, solicitud/respuesta, respuesta solicitada, mensajería asíncrona de una-vía, o incluso notificación. SOAP es un protocolo de alto nivel que sólo define la estructura del mensaje y unas pocas reglas para su procesamiento. Es completamente independiente del protocolo de transporte subyacente, por eso los mensajes SOAP se pueden intercambiar sobre HTTP, JMS o protocolos de transporte de e-mail. Actualmente el protocolo HTTP es el más utilizado para los mensajes HTTP.

# Visión General Servicios Web



- ¿ Para qué sirve WSDL ?
  - Permite que un servicio y un cliente establezcan un acuerdo en lo que se refiere a los detalles de transporte de mensajes y su contenido, a través de un documento
  - Representa una especie de contrato entre el proveedor y el que solicita.

# Visión General Servicios Web



- WSDL es un documento XML que contiene un conjunto de definiciones que describen un Servicio Web. Proporciona toda la información necesaria para acceder y utilizar un Servicio Web. Un documento WSDL describe qué hace el Servicio Web, cómo se comunica, y dónde reside. Usamos el documento WSDL en el momento del despliegue para crear nuestros interfaces de servicio.

# Visión General Servicios Web



- Evolución
  - Las aplicaciones basadas en Servicios Web de las grandes aplicaciones han ido evolucionando a lo largo del tiempo y experimentando un enriquecimiento de la estructura semántica



- Introducción
  - Podemos bien en cliente o servidor realizar peticiones mediante http a un servidor web. Los mensajes deben tener un formato determinado empleando XML para encapsular los parámetros de la petición. Con el paso del tiempo el proyecto interesó a importantes multinacionales entre las que se encuentran IBM y Microsoft y de este interés por XML-RPC se desarrollo SOAP

# Soap



- Soap aporta flexibilidad y características a los servicios web.
- En el núcleo de los servicios Web se encuentra el protocolo simple de acceso a datos SOAP, que proporciona un mecanismo estándar de empaquetar mensajes



- Características
  - **No esta asociado con ningún lenguaje:** los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista pero los desarrolladores responsables de mantener antiguas aplicaciones podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación, como en Java, y la plataforma como Microsoft .Net.



- **No se encuentra fuertemente asociado a ningún protocolo de transporte:**
  - La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.

# Soap



- **Aprovecha los estándares existentes en la industria:** Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y como ya se ha mencionado SOAP no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.



- **Permite la interoperabilidad entre múltiples entornos:**
  - SOAP se desarrollo sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dicho estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas. Por ejemplo, una aplicación de escritorio que se ejecute en una PC puede comunicarse con una aplicación del ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP.

# Soap



- Formato Soap

```
<?xml version="1.0"?>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soap:Header>
```

```
<!--Optional header information goes here. -->
```

```
<To>Scott</To>
```

```
<From>Suzanne</From>
```

```
</soap:Header>
```

```
<soap:Body>
```

```
<!--Message goes here. -->
```

```
Please pick up some milk on your way home from work.
```

```
</soap:Body>
```

```
</soap:Envelope>
```

# Soap



- El elemento raíz del documento es el elemento Envelope. El ejemplo contiene dos subelementos, Body y Header. Un ejemplo de SOAP válido también puede contener otros elementos hijo en el sobre

# Soap



- El sobre puede contener un elemento Header opcional que contiene información sobre el mensaje. En el ejemplo anterior, la cabecera contiene dos elementos que describen a quien compuso el mensaje, y posible receptor del mismo.

El sobre debe contener un elemento body el elemento body (cuerpo) contiene la carga de datos del mensaje. En el ejemplo el cuerpo contiene una simple cadena de caracteres.

# Soap



- Un mensaje debe estar dentro de sobre de SOAP bien construido. Un sobre se compone de un único elemento envelope el sobre puede contener un elemento Header y puede contener un elemento body. Si existe, la cabecera debe ser el elemento hijo inmediato del sobre, con el cuerpo siguiendo inmediatamente a la cabecera.

# Soap



- El cuerpo contiene la carga de datos del mensaje y la cabecera contiene los datos adicionales que no pertenecen necesariamente al cuerpo del mensaje.  
Además de definir un sobre de SOAP, la especificación de SOAP define una forma de codificar los datos contenidos en un mensaje. La codificación de SOAP proporciona un mecanismo estándar para serializar tipos de datos no definidos en la parte 1 de la especificación del esquema de XML.

# Soap



- **Descripción el protocolo de comunicación SOAP**

- `<?xml version="1.0"?>`  
`<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"`  
`Soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">`  
`<soap:Header>`  
`...`  
`</soap:Header>`  
`<soap:Body>`  
`...`  
`<soap:Fault>`  
`...`  
`</soap:Fault>`  
`</soap:Body>`  
`</soap:Envelope>`

# Soap



- `<?xml version="1.0"?>` Como podemos ver en esta línea SOAP es un documento XML, y como tal, debe comenzar con el tag `<?xml....?>` y la versión correspondiente.
- `<soap:Envelope` Aquí se indica que comienza el envelope (sobre) del mensaje

# Soap



- `xmlns:soap = "http://www.w3.org/2001/12/soap-envelope"` Un mensaje SOAP debe contener siempre un elemento envelope asociado con el namespace (espacio de nombres)  
`http://www.w3.org/2001/12/soap-envelope`

# Soap



- `Soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">` En esta línea lo que se hace es indicar donde se encuentran definidos los tipos de datos utilizados en el documento.



- `<soap:Header>` Esta línea indica el comienzo del Header (encabezado). En esta sección se incluye información específica del mensaje, como puede ser la autenticación, descripción.
- `</soap:Header>` Como todo documento XML los tags que son abiertos deben ser cerrados, esta línea indica la finalización del Header (encabezado).

# Soap



- `<soap:Body>` Aquí comienza el cuerpo del mensaje, en esta sección se incorpora toda la información necesaria para el nodo final. Por ejemplo, los parámetros para la ejecución, o la respuesta a una petición.

# Soap



- `<soap:Fault>` Cualquier tipo de fallo que se produzca será notificado en esta sección. La cual esta contenida dentro del cuerpo del mensaje.
- `</soap:Fault>` Cierre de la sección Fault.

# Soap



- `</soap:Body>` Indica el final del cuerpo del mensaje.
- `</soap:Envelope>` Fin del mensaje SOAP

# Soap



- `<?xml version='1.0' ?>`
- `<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">`
- `<env:Header>`
- `<m:reserva xmlns:m="http://empresaviajes.example.org/reserva" env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:mustUnderstand="true">`  
`<m:referencia>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:referencia>`  
`<m:fechaYHora>2001-11-29T13:20:00.000-05:00</m:fechaYHora> </m:reserva>`

# Soap



- `<n:pasajero  
xmlns:n="http://miempresa.example.com/emple  
ados"  
env:role="http://www.w3.org/2003/05/soap-  
envelope/role/next"  
env:mustUnderstand="true">`
- `<n:nombre>Åke Jógvan Øyvind</n:nombre>  
</n:pasajero> </env:Header> <env:Body>`