



Capacitación Desarrollo y Consumo de WebServices

Ing. José Fernando Ramos

Contenido

PARTE TEÓRICA

¿Qué son los WebServices?

Beneficios de utilizar WS

Arquitectura

¿Qué es XML ?

- Características
- Estructura
- Ejemplos

¿Qué es WSDL?

- Elementos
- Ejemplos

Contenido

PARTE TEÓRICA

¿Qué es WSDL?

- Elementos

¿Qué es SOAP?

- Ventajas

- Anatomía

- Modelo de Intercambio

¿Qué es REST?

- Características

- Modelo

- Representación URI

- Beneficios

Contenido

PARTE PRÁCTICA

Servicio Web SOAP (ASMX)

- Participantes

Servicios Web SOAP (WCF)

- Operaciones Financieras

Servicio Web REST

- Operaciones Financieras

Consumo de Servicio Web con MVC

¿Qué son los WebServices?

Un web service es una vía de intercomunicación e interoperabilidad entre máquinas conectadas en Red. En el mundo de Internet se han popularizado enormemente, ya se trate de web services públicos o privados. Generalmente, la interacción se basa en el envío de solicitudes y respuestas entre un cliente y un servidor, que incluyen datos

Estándares Empleados



XML

Extensible Markup Language



SOAP

Simple Object Access Protocol



WSDL

Web Services Description Language



REST

Representational State Transfer

Beneficios de Utilizar WS

- Aportan interoperabilidad entre aplicaciones de software independientemente de la plataforma.
- Fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permite la intercomunicación sin importar la aplicación y lenguaje de programación empleada.

Arquitectura



SERVICE DISCOVERY

Centraliza los servicios web en punto común



SERVICE DESCRIPTION

Describir el servicio web para su consumo



SERVICE INVOCATION

Transmitir el mensaje de petición y respuesta



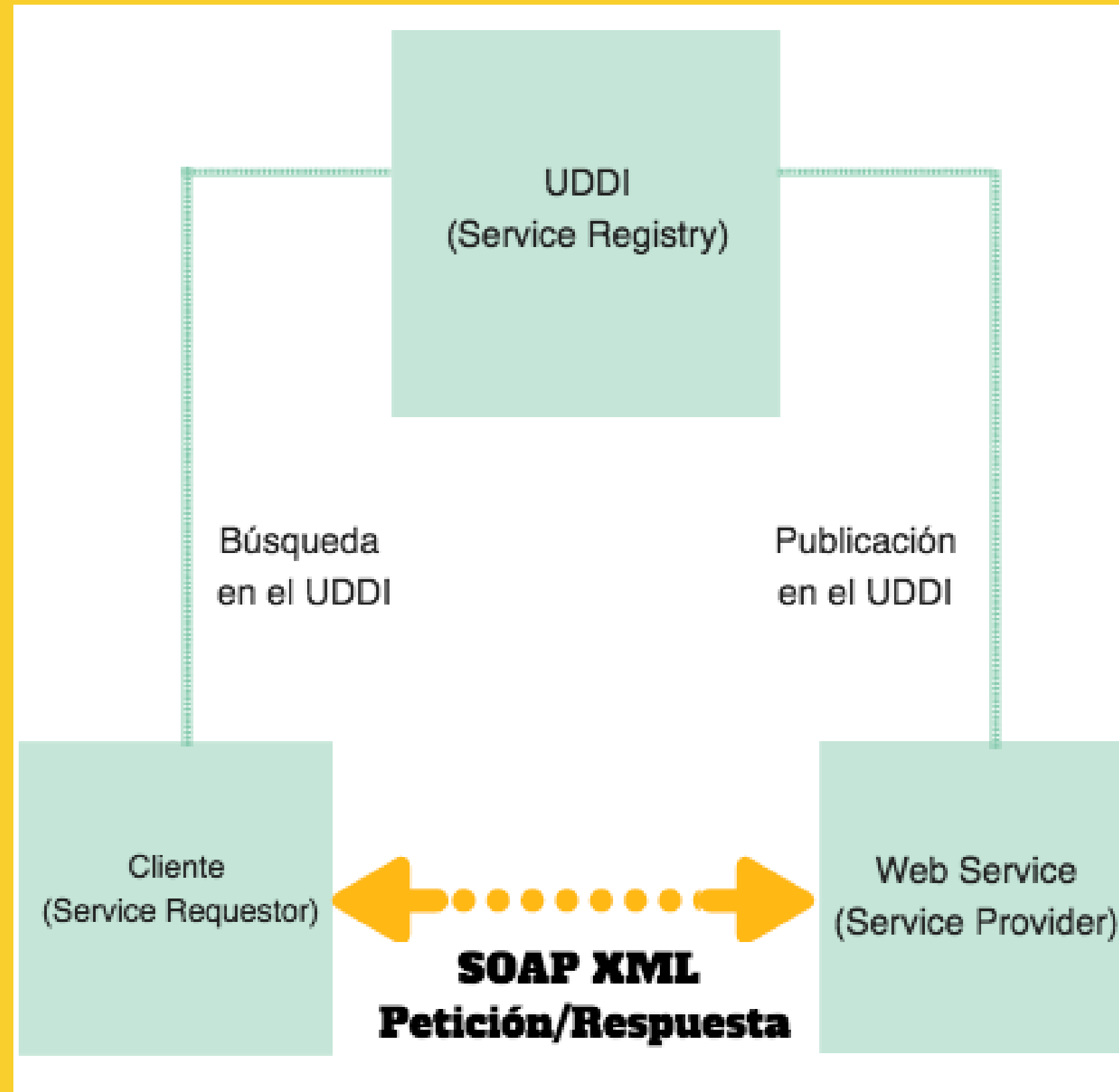
TRANSPORT

Medio de transmisión, se utiliza HTTP

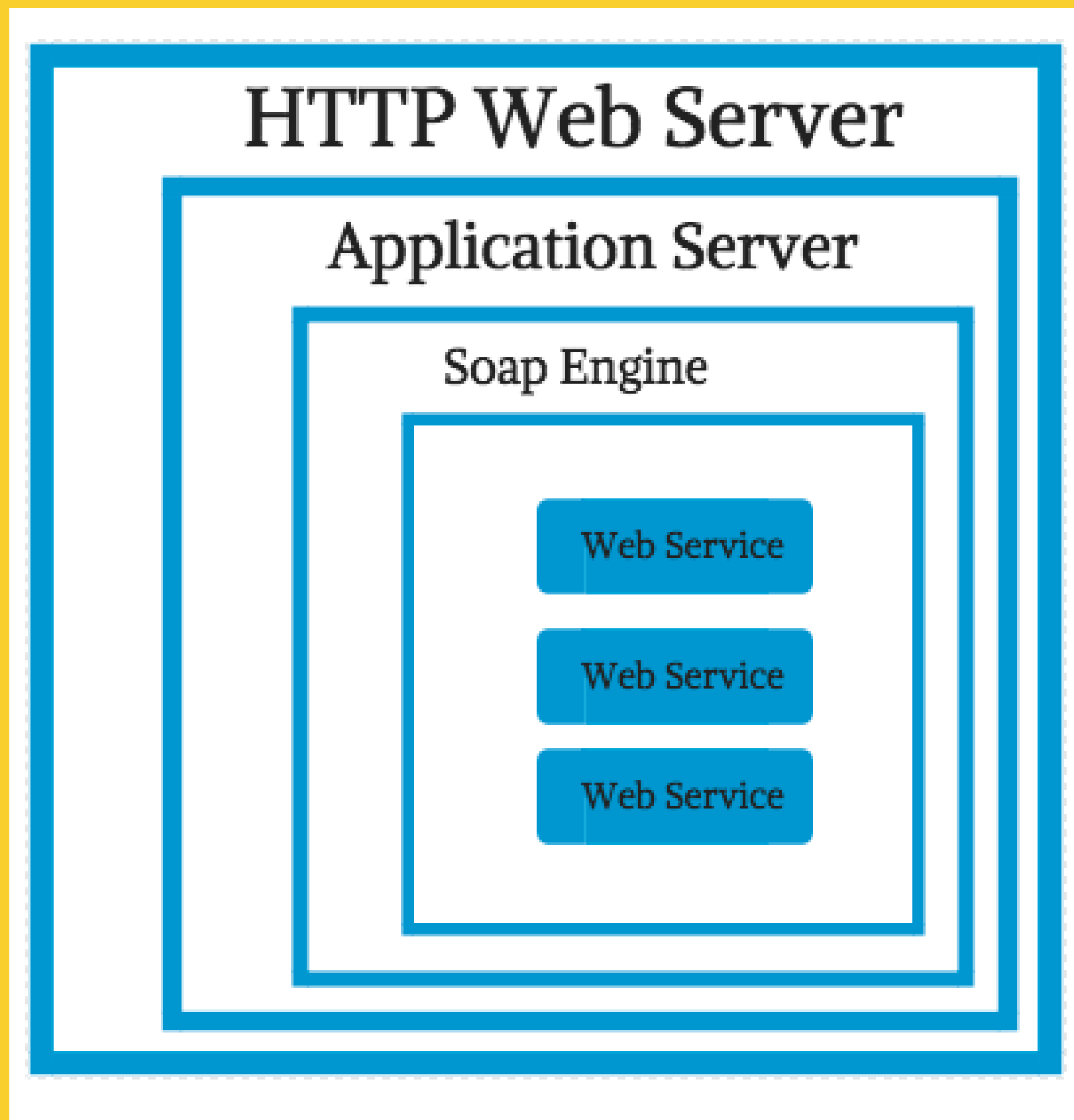
¿Cómo funciona?

1. El Service Provider genera el WSDL describiendo el Web Service y registra el WSDL en el directorio UDDI o Service Registry.
2. El Service Request o la aplicación del cliente requiere un Web Service y se pone en contacto con el UDDI para localizar el Web Service.
3. El cliente, basándose en la descripción descrita por el WSDL, envía un request para un servicio particular al Web Service Listener, que se encarga de recibir y enviar los mensajes en formato SOAP.
4. El Web Service analiza el mensaje SOAP del request e invoca una operación particular en la aplicación para procesar el request. El resultado se escribe de nuevo en SOAP en forma de respuesta y se envía al cliente.
5. El cliente analiza el mensaje de respuesta SOAP y lo interpreta o genera un error si ha habido alguno.

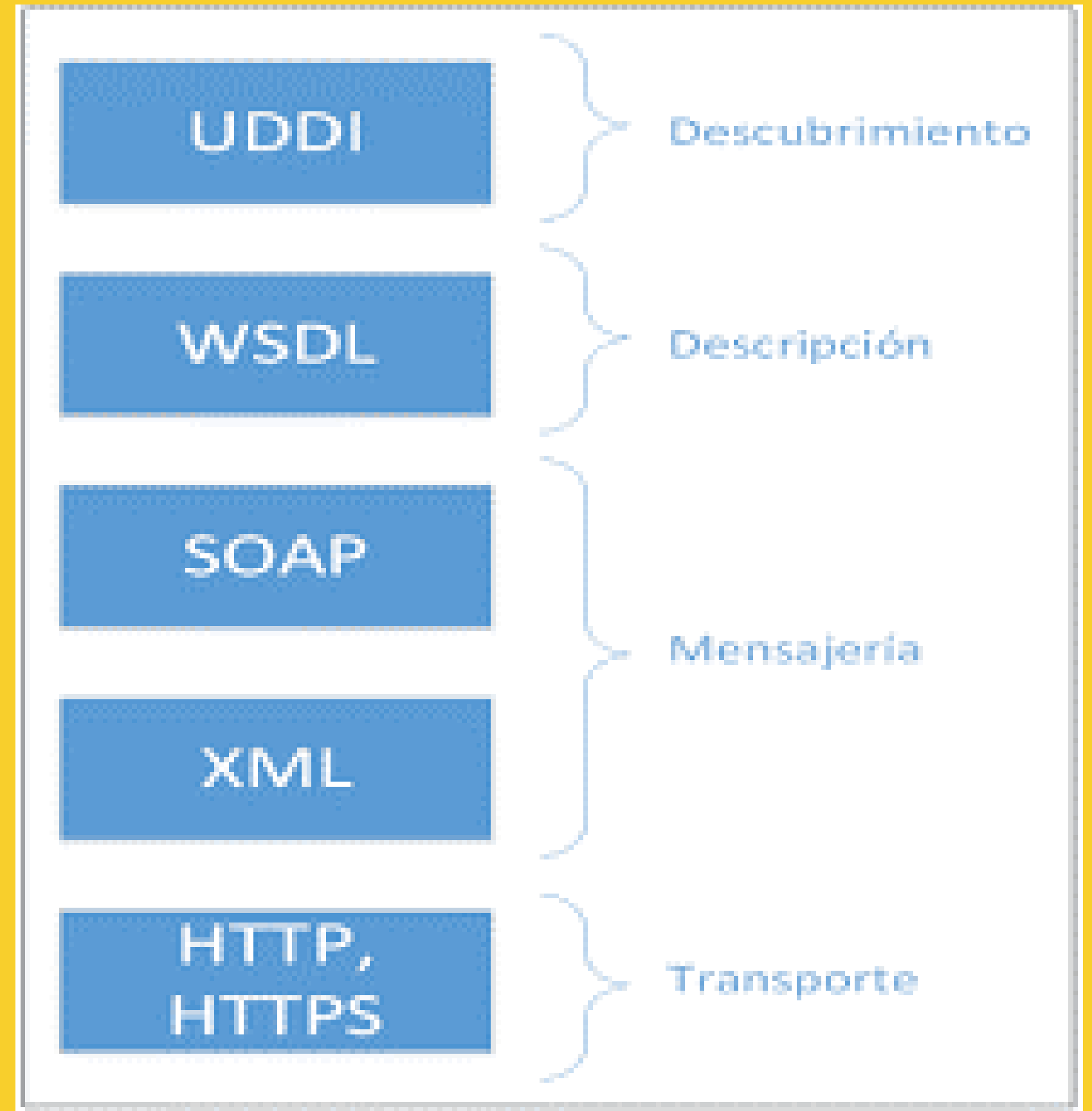
¿Cómo funciona?



Componentes



Estructura



¿Qué es XML?

XML, es el estándar de Extensible Markup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados

Características

- Arquitectura abierta y extensible.
- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML.
- Permite el intercambio entre cualquier tipo de aplicación.
- Permite gestionarse y manipularse para integrarse a cualquier entorno de aplicaciones.

Partes del XML

PRÓLOGO

```
<?xml version="1.0" encoding="UTF-8"?>
```

CUERPO

```
<Edit_Mensaje>  
  (...)  
</Edit_Mensaje>
```

ELEMENTOS

Los elementos XML pueden tener contenido (más elementos, caracteres o ambos), o bien ser elementos vacíos.

Partes del XML

```
<Edit_Mensaje>  
<Participante>  
<Nombre></Nombre>  
</Participante>  
</Edit_Mensaje>
```

ATRIBUTOS

```
<Participante  
  Nacionalidad="Hondureña">Información</Participante>
```


¿Qué es WSDL?

WSDL (Web Services Description Language) es un protocolo basado en XML que describe los accesos al Web Service. Podríamos decir que es el manual de operación de este, porque nos indica cuáles son las interfaces que provee el Servicio web y los tipos de datos necesarios para su utilización.

Elementos del WSDL

<DEFINITIONS>

Comienzo del documento, este tag agrupa a todos los demás elementos

<TYPES>

Se definen los tipos de datos utilizados en los mensajes

<MESSAGE>

Se definen los métodos y parámetros para realizar la operación

<PORTTYPE>

Esta sección es la más importante, ya que definen las operaciones que pueden ser realizadas

Elementos del WSDL

<BINDING>

Se definen el formato del mensaje y detalles del protocolo para cada portType

<SERVICE>

Colección de puntos finales donde se encuentra alojado el servicio

¿Qué es SOAP?

Simple Object Access Protocol es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por Dave Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros. Está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

Ventajas

- No está asociado a ningún lenguaje.
- No se encuentra fuertemente asociado a ningún protocolo de transporte.
- No está atado a ninguna infraestructura de objeto distribuido.
- Aprovecha los estándares existentes en la industria.
- Permite la interoperabilidad entre múltiples entornos

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reserva xmlns:m="[http://www.example.org]"
      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>
    <n:pasajero xmlns:n="http://miempresa.example.com/empleados"
      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>
    <p:itinerario xmlns:p="http://empresaviajes.example.org/reserva/viaje">
      <p:ida>
        <p:salida>Nueva York</p:salida>
        <p:llegada>Los Angeles</p:llegada>
        <p:fechaSalida>2001-12-14</p:fechaSalida>
        <p:horaSalida>última hora de la tarde</p:horaSalida>
        <p:preferenciaAsiento>pasillo</p:preferenciaAsiento>
      </p:ida>
      <p:vuelta>
        <p:salida>Los Angeles</p:salida>
        <p:llegada>Nueva York</p:llegada>
        <p:fechaSalida>2001-12-20</p:fechaSalida>
        <p:horaSalida>media-mañana</p:horaSalida>
        <p:preferenciaAsiento />
      </p:vuelta>
    </p:itinerario>
    <q:alojamiento xmlns:q="http://empresaviajes.example.org/reserva/hoteles">
      <q:preferencia>ninguna</q:preferencia>
    </q:alojamiento>
  </env:Body>
</env:Envelope>
```

Modelo de Intercambio

- Los mensajes SOAP son transmisiones unidireccionales desde un emisor a un receptor.
- Se suelen combinar mensajes para implementar patrones, como petición/respuesta.
- Las implementaciones SOAP se pueden optimizar para explotar las características específicas de sistemas de red concretos.

Ejemplo: Petición

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productId>827635</productId>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

Ejemplo: Respuesta

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Toptimate 3-Piece Set</productName>
        <productId>827635</productId>
        <description>3-Piece luggage set.  Black Polyester.</description>
        <price>96.50</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```


¿Qué es REST?

Representational State Transfer o traducido a "Transferencia de presentación de estado" es lo que se domina a REST. ¿Y eso es?, una técnica de arquitectura de software usada para construir APIs que permitan comunicar a nuestro servidor con sus clientes usando el protocolo HTTP mediante URIs lo suficientemente inteligentes para poder satisfacer la necesidad del cliente.

Tipos de Peticiones

GET

Utilizado para obtener datos a través de consultas "select".

POST

Utilizado para enviar datos, típicamente para operaciones de "insert".

PUT

Utilizado para actualizar datos, típicamente para operaciones de "update".

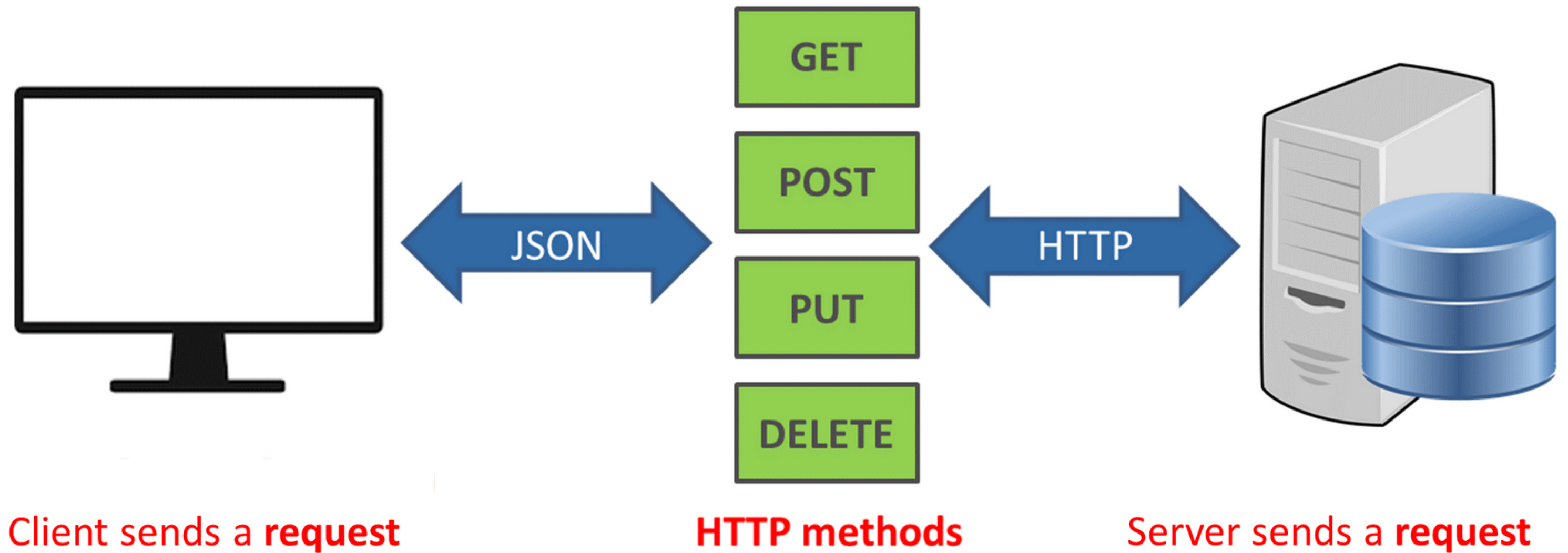
DELETE

Utilizado para borrar datos, típicamente para operaciones de "delete".

Características del REST

- Devuelven las respuestas a las peticiones en formatos de documentos XML o JSON, ya que es el lenguaje de intercambio de información más usado.
- Los objetos en REST siempre se manipulan a partir de la URI. La URI nos facilita acceder a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.

Modelo del REST



Representación URI

My Blog API

A simple demonstration of a Flask RestPlus powered API

blog/categories : Operations on blog categories

[Show/Hide](#)[List Operations](#)[Expand Operations](#)

GET	/blog/categories/	Returns list of blog categories
POST	/blog/categories/	Creates a new blog category
DELETE	/blog/categories/{id}	Deletes blog category
GET	/blog/categories/{id}	Returns a category with a list of posts
PUT	/blog/categories/{id}	Updates a blog category

blog/posts : Operations related to blog posts

[Show/Hide](#)[List Operations](#)[Expand Operations](#)

[BASE URL: /api , API VERSION: 1.0]

Beneficios

SEPARACIÓN ENTRE EL CLIENTE Y EL SERVIDOR

El protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos

VISIBILIDAD, FIABILIDAD Y ESCALABILIDAD

La separación entre cliente y servidor tiene una ventaja evidente y es que cualquier equipo de desarrollo puede escalar el producto sin excesivos problemas

NO IMPORTA LA TECNOLOGÍA

REST siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando; JAVA, .NET, PHP, PYTHON, NodeJS, etc

Desarrollo Práctico

WS EN ASMX

WS CON WCF

WS CON REST

CONSUMO CON MVC