

Tema 8

Servicios Web

SW Corporativo

Curso 2018-19

Objetivos del Tema 8

- Conocer los conceptos básicos en torno a Web Services aplicado a los ERP (Software Corporativo)
- Entender de importancia de Servicios Web dentro de los procesos operacionales de la Empresa.
- Relación con buena parte de los procesos operativos en los que se requiere involucrar a terceros (Clientes, proveedores, administraciones públicas...), mejorando las relaciones y creando valor.
- Se presupone conocimiento previo del alumnado

Servicios Web

Un servicio web (en inglés, Web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos (comunicarse) 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. (Ref. wiki).

Las aplicaciones “hablan” entre ellas: B2B

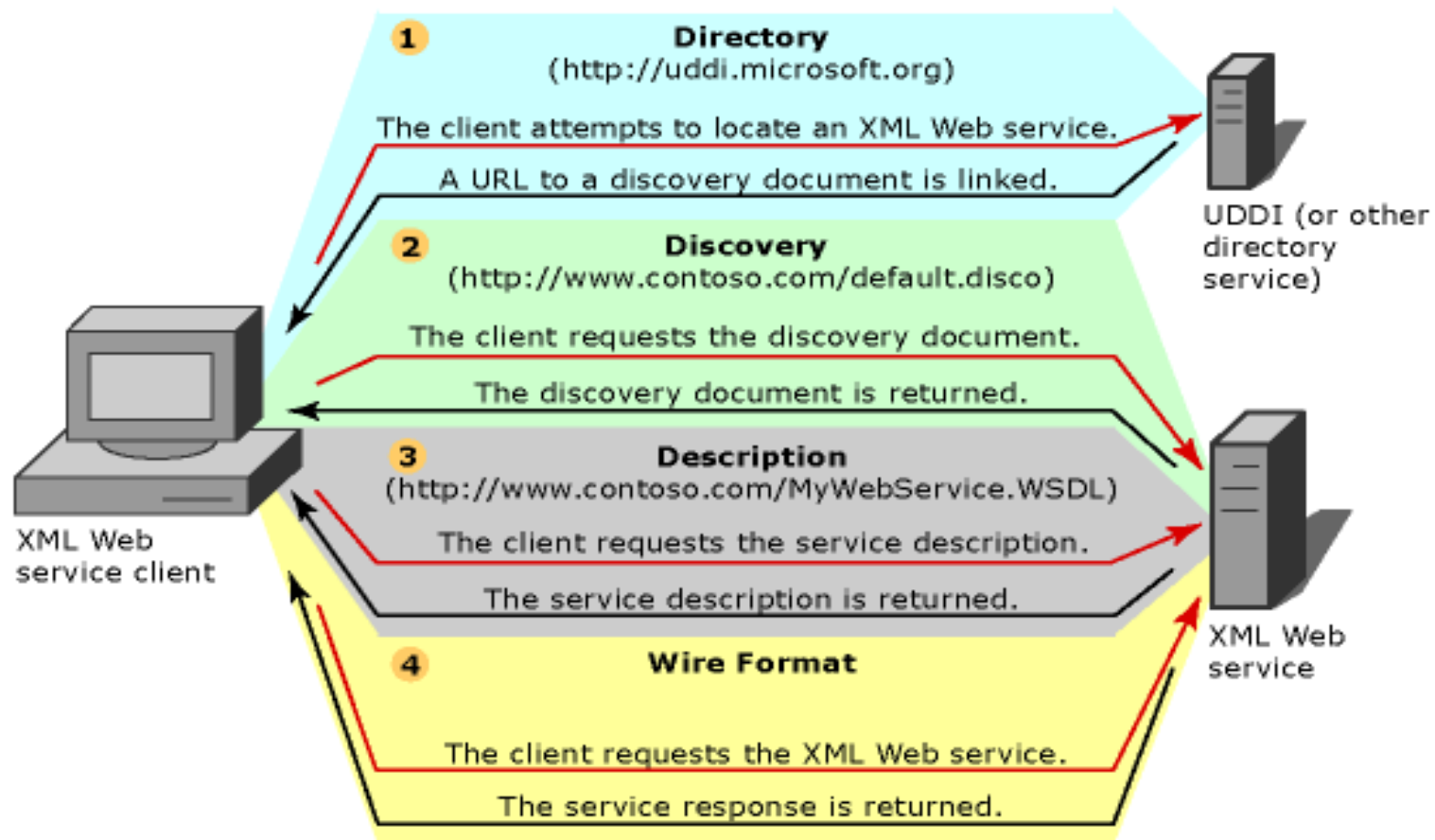
Características de los servicios web

- Un servicio debe poder ser **accesible a través de la Web**. Para ello debe utilizar **protocolos de transporte estándares como HTTP**, y codificar los mensajes en un lenguaje estándar que pueda conocer cualquier cliente que quiera utilizar el servicio.
- Un servicio debe contener una **descripción de sí mismo**. De esta forma, **una aplicación podrá saber cuál es la función de un determinado Servicio Web, y cuál es su interfaz, de manera que pueda ser utilizado de forma automática por cualquier aplicación, sin la intervención del usuario.**
- Debe poder **ser localizado**. Deberemos tener algún mecanismo que nos permita encontrar un Servicio Web que realice una **determinada función**. De esta forma tendremos la posibilidad de que una aplicación localice el servicio que necesite de forma automática, sin tener que conocerlo previamente el usuario.

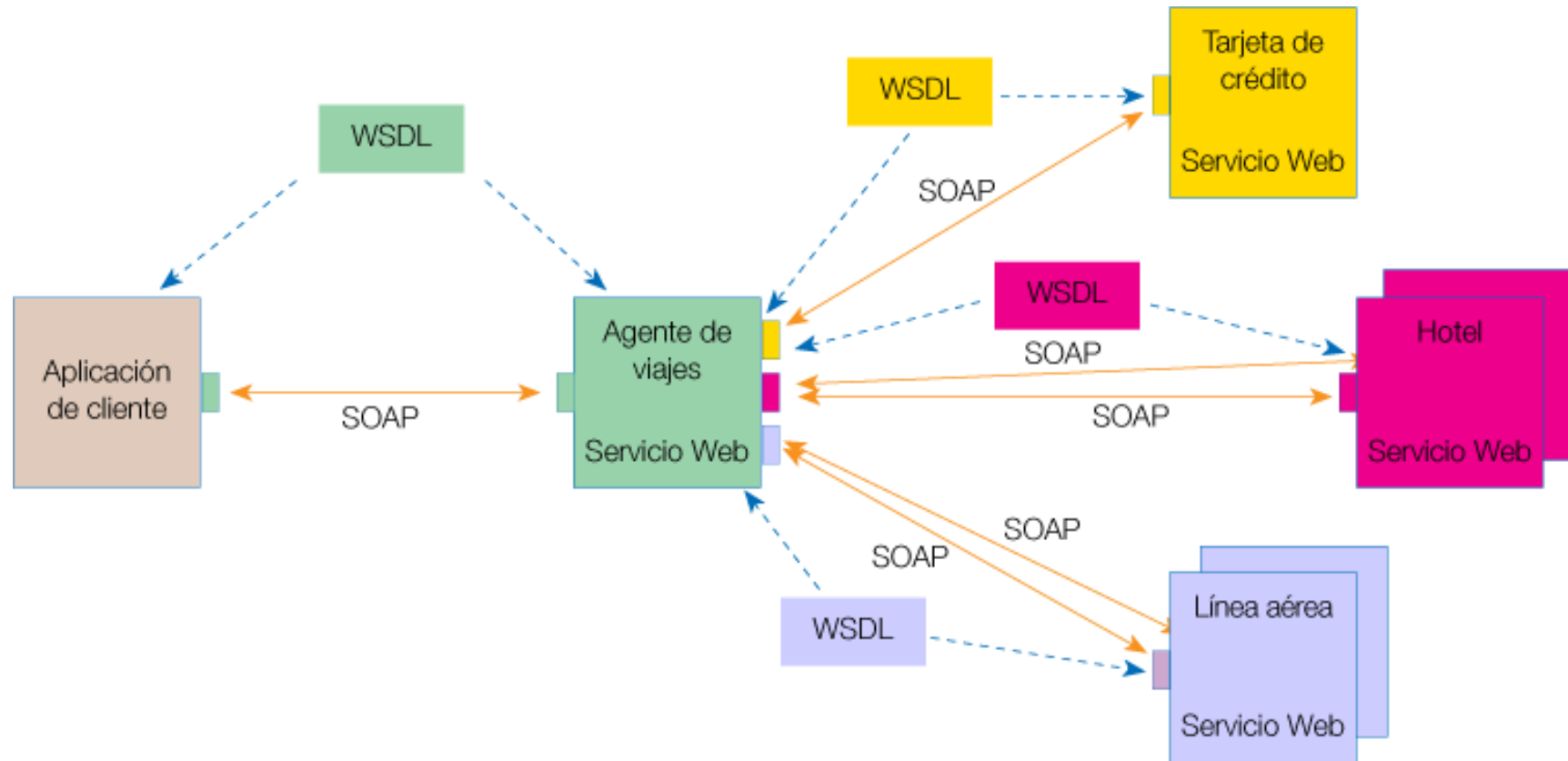
Tipos de Servicios Web

- A nivel técnico, los servicios pueden implementarse de varias formas. En este sentido, podemos distinguir varios tipos conocidos de servicios Web, según se base tecnológica: **XML-RPC, SOAP, RESTful**
- A nivel conceptual, un servicio web es un **componente software** proporcionado a través de un **endpoint** *accesible a través de la red*.
- **Definición de endpoint:** Los servicios web pueden interconectarse a través de la red. En una arquitectura orientada a servicios, cualquier interacción punto a punto implica dos *endpoints*: *uno que proporciona un servicio, y otro de lo consume*. Es decir, que un *endpoint es cada uno de los "elementos", en nuestro caso nos referimos a servicios, que se sitúan en ambos "extremos" de la red que sirve de canal de comunicación entre ellos*. Cuando hablamos de servicios Web, un *endpoint se especifica mediante una URI*.

Infraestructura típica



Ejemplo: Una agencia de viajes



Servicios Web. Escenarios:

- **Simple**
 - Proporcionar alguna parte fundamental de funcionalidad para que sus clientes la utilicen.
- **Integración de Aplicaciones**
 - Manera de integrar un grupo aparentemente dispar de aplicaciones existentes.
- **Soluciones de Flujo de Trabajo (Workflow)**
 - Realizando servicios web para cada componente necesario para realizar una tarea, orquestando a través de una aplicación externa la ejecución del flujo.

Arquitectura básica de Servicios Web

Para que los servicios web funcionen adecuadamente, hay que implementar diferentes “**roles**”:

- **Proveedor del servicio**: La aplicación que implementa el servicio y lo hace accesible desde Internet.
- **Solicitante del servicio**: Cualquier cliente que pueda requerir el funcionamiento del servicio web para su uso. Puede ser un formulario web u otro servicio.
- **Registro y publicación del servicio**: Idealmente, debe haber algún repositorio centralizado de servicios que nos proporcione información sobre los servicios que tenemos disponibles y cómo se utilizan.

Service Oriented Architecture (SOA): Características

- Define utilizar servicios para implementar la lógica de negocio.
- Sistemas altamente escalables.
- Facilita la interacción entre sistemas.
- Servicios poco acoplados pero altamente interoperables.
- Basados en una definición independiente de plataforma.
- Alta reusabilidad.
- Poco sensible a cambios.

Terminología utilizada en SOA

- **Servicio**
 - Funcionalidad publicada.
- **Orquestación**
 - Coordinación de servicios para implementar una lógica determinada.
- **Sin estado**
 - Los servicios no dependen unos de otros y se pueden orquestar de distintas maneras.
- **Proveedor**
- **Consumidor**

Protocolos de Servicio

- Diversos estándares de comunicación:
 - Servicio de transporte :**HTTP**, FTP, SMTP, BEEP
 - Mensajería XML, maneras de codificar los mensajes que se precisan para invocar un servicio web : **XML-RPC**, **SOAP**.
 - Descripción del servicio : **WSDL**. ¿Qué hacen y cómo se usan?
 - Publicación del servicio : **UDDI**, ofrece un directorio de servicios en Internet.

Mensajería XML

- XML es un lenguaje que se ha popularizado mucho pues **permite el intercambio de información de manera fácil independientemente de la plataforma o el lenguaje de programación.**
- Hay multitud de herramientas y librerías casi en cualquier lenguaje para producir XML, por lo que cualquier programador puede integrarlo de manera fácil en sus editores y sus lenguajes.
- Hay dos tipos de contenedores de mensajes XML:
XML-RPC y SOAP

SOAP . Definición

- SOAP = Simple Object Application Protocol
- **SOAP** es todo un **protocolo basado en XML para el intercambio de información entre ordenadores**. Aunque se puede utilizar en diferente ámbitos, donde más éxito está teniendo es en el intercambio de datos vía HTTP (servicios web).
- También es **independiente de la plataforma y del lenguaje de programación**.
- Normalmente utilizaremos SOAP para conectarnos a un servicio e invocar métodos remotos.

El protocolo SOAP

- Simple Object Access Protocol
 - Invocación remota basada en XML
 - Mensaje estructurado en XML con posibilidad de implementar múltiples formatos, seguridad etc.
 - Contiene llamadas y respuestas de procedimientos
 - Trata de sustituir a formatos propietario

El protocolo SOAP

- En concreto SOAP especifica:
 - Un **formato de mensaje**, describiendo como se organiza la información en un documento XML.
 - Un conjunto de **normas**, definiendo como los clientes pueden **invocar** un procedimiento remoto enviando un mensaje SOAP y como los servicios pueden replicar enviando otro mensaje al cliente.
 - Un conjunto de **reglas** que cualquier entidad que procesa un mensaje SOAP debe seguir. Definen **que elementos debería leer y comprender**, y las acciones que deberían realizar si no lo entienden.
 - Una **descripción** de cómo su mensaje SOAP se debería transportar sobre HTTP y SMTP.

Un ejemplo de mensaje SOAP

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns:getTemperatura xmlns:ns="http://j2ee.ua.es/ns">
      <area>Alicante</area>
    </ns:getTemperatura>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- En él estamos llamando a nuestro método getTemperatura para obtener información meteorológica, proporcionando como parámetro el área de la que queremos obtener la temperatura.

WSDL

- WSDL representa una capa de **descripción del servicio**.
- Consiste en una **gramática XML** que sirve para especificar **la interfaz pública** de un servicio web.
- Esta interfaz pública puede incluir:
 - Información sobre todas las **funciones públicas disponibles**
 - **Tipos** de todos los **parámetros** de los mensajes XML
 - Información vinculante sobre el **protocolo de transporte específico** que se va a utilizar
 - Información sobre **localización del servicio Web**
- Aunque WSDL no está ligado específicamente a una mensajería XML concreta, lo cierto es que incluye extensiones para describir servicios SOAP.

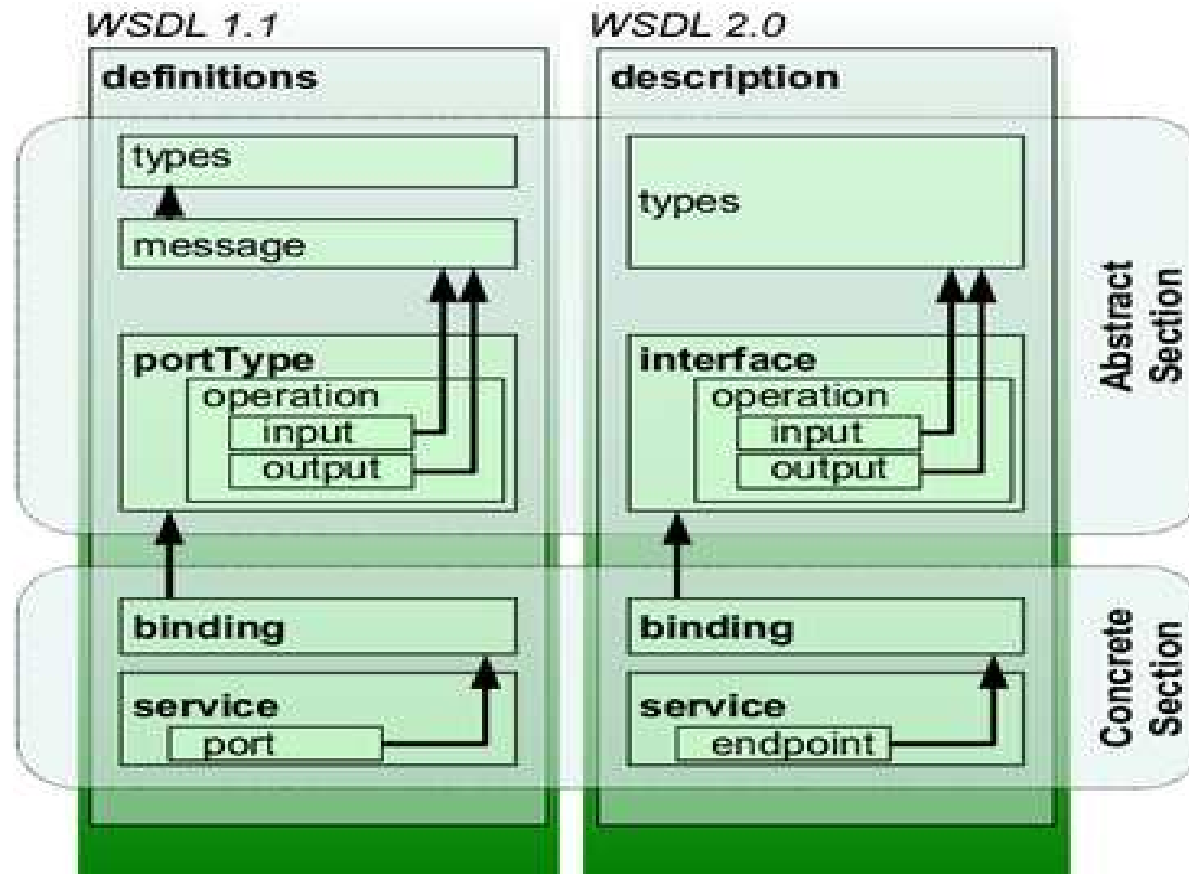
Documentos WSDL

- WSDL describe un servicio utilizando varios elementos (etiquetas xml).
- Dichos elementos podemos clasificarlos como abstractos o concretos.
- La **parte WSDL abstracta** describe las operaciones y mensajes con detalle. En otras palabras, la parte abstracta de un WSDL especifica **QUÉ hace el servicio**:
 - Qué operaciones están disponibles
 - Qué entradas, salidas, y mensajes de error tienen las operaciones
 - Cuáles son las definiciones de los tipos para los mensajes de entrada, salida y error

Documentos WSDL (cont.)

- La parte **WSDL concreta describe el cómo y dónde** del servicio:
 - Cómo tiene que llamar un cliente al servicio
 - Qué protocolo debería usar
 - Dónde está disponible el servicio

Estructura de un documento WSDL



UDDI

- **Consiste en una especificación técnica para publicar y buscar servicios Web.**
- Define la especificación para construir un directorio distribuido de Servicios Web, donde los datos se almacenan en XML.
- En este registro no sólo se almacena información sobre servicios, sino también sobre las organizaciones que los proporcionan, la categoría en la que se encuentran, y sus instrucciones de uso (en WSDL)
- Accedido por mensajes SOAP para obtener documentos WSDL

XML-RPC

- XML-RPC es un [protocolo de llamada a procedimiento remoto](#) que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.
- Es un [protocolo muy simple ya que solo define unos cuantos tipos de datos y comandos útiles](#), además de una descripción completa de corta extensión.
- La simplicidad del XML-RPC contrasta con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso.
- [A partir de 1998, al considerar Microsoft que era muy simple decidió añadirle funcionalidades, tras las cuales, después de varias etapas de desarrollo, el estándar dejó de ser sencillo y se convirtió en lo que es actualmente conocido como SOAP.](#)
- Una diferencia fundamental es que en los procedimientos en SOAP los parámetros tienen nombre y no interesa su orden, no siendo así en XML-RPC.

Uso popularizado

- Existen implementaciones para varios sistemas operativos, lenguajes de programación, licencias comerciales y de software libre: C/C++, Delphi, Java, Microsoft .NET, Perl, PHP, Python,...

Ejemplo de Conexión

[Python](#) [Ruby](#) [PHP](#) [Java](#)

```
final String url = <insert server URL>,  
    db = <insert database name>,  
    username = "admin",  
    password = <insert password for your admin user (default: admin  
>>;
```

[Python](#) [Ruby](#) [PHP](#) [Java](#)

```
final XmlRpcClient client = new XmlRpcClient();  
  
final XmlRpcClientConfigImpl start_config = new XmlRpcClientConfigImpl(  
);  
start_config.setServerURL(new URL("https://demo.odoo.com/start"));  
final Map<String, String> info = (Map<String, String>)client.execute(  
    start_config, "start", emptyList());  
  
final String url = info.get("host"),  
    db = info.get("database"),  
    username = info.get("user"),  
    password = info.get("password");
```

```
final XmlRpcClientConfigImpl common_config = new XmlRpcClientConfigImpl();
common_config.setServerURL(
    new URL(String.format("%s/xmlrpc/2/common", url)));
client.execute(common_config, "version", emptyList());

{
    "server_version": "8.0",
    "server_version_info": [8, 0, 0, "final", 0],
    "server_serie": "8.0",
    "protocol_version": 1,
}
```

Ejemplo de llamada de un Método

```
Python Ruby PHP Java

final XmlRpcClient models = new XmlRpcClient() {{
    setConfig(new XmlRpcClientConfigImpl() {{
        setServerURL(new URL(String.format("%s/xmlrpc/2/object", url)))
    }});
}};

models.execute("execute_kw", asList(
    db, uid, password,
    "res.partner", "check_access_rights",
    asList("read"),
    new HashMap() {{ put("raise_exception", false); }}
));

true
```

Lectura de un registro

```
final List ids = asList((Object[])models.execute(
    "execute_kw", asList(
        db, uid, password,
        "res.partner", "search",
        asList(asList(
            asList("is_company", "=", true),
            asList("customer", "=", true))),
        new HashMap() {{ put("limit", 1); }}}));
final Map record = (Map)((Object[])models.execute(
    "execute_kw", asList(
        db, uid, password,
        "res.partner", "read",
        asList(ids)
    )
))[0];
// count the number of fields fetched by default
record.size();
```

Implementaciones

- Algunas implementaciones conocidas son:
 - **Apache XML-RPC**, una implementación en Java.
 - **XMLRPC-EPI**, una implementación C.
 - **XML-RPC-C**, una implementación para C y C++.