

TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

Lenguaje de marcas: aquel que para codificar un documento se apoya en el uso de etiquetas o marcas

Estas marcas proporcionan información acerca de la estructura del texto, su contenido o su presentación

XML:

- metalenguaje de marcado .

- Multiplataforma, sencillo, gratuito y de fácil aprendizaje

- Permite crear etiquetas de acuerdo a las necesidades del usuario, de ahí lo de extensible

- Este lenguaje es un estándar del Consorcio WWW(W3C)

- El sitio oficial del estándar es <http://www.w3.org/XML>



TEMA 8 . LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

Historia de los lenguajes de marcas. El estándar XML

Primer lenguaje de marcas lo registró IBM en los años 70: GML (generalized Markup Language).

A raíz de ahí, la ISO, creó el SGML , lenguaje muy potente, capaz de adaptarse a las múltiples situaciones y problemas.

1989 , Tim Berners Lee, decide crear el HTML , encaminado a web el vertiginoso éxito de la web cogió desprevenidos a los desarrolladores de HTML, generando un crecimiento descontrolado del lenguaje.

En 1996, HTML se convierte en un estándar W3C,

En 1998 el W-C, dada la popularidad de HTML, lanza el lenguaje de Mercado Extensible o XML

XML es un metalenguaje de marcado , es decir que **permite describir a otro lenguaje a través del uso de etiquetas.**



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

XML permite estructurar los datos de manera que estos puedan ser modelados emulando un árbol o jerarquía.

Un árbol XML está conformado por uno o varios nodos

Cada nodo representa una unidad de información y puede contener mas nodos o unidades.

El árbol XML no incluye información sobre cómo presentar o utilizar estos datos. La forma de modelar un árbol de datos es arbitraria y depende de las necesidades de quien lo modela

Otra de las ventajas es que permite crear a los usuarios sus propias etiquetas , adaptándose a las necesidades de los usuarios

Independiza el formateo de los datos , de los datos en si mismos

Por otro lado hay que tener en cuenta que está regido por estrictas reglas que guían la composición de un documento XML, facilitando así su análisis sintáctico y permitiendo la creación de técnicas de extracción y búsqueda de información



TEMA 8 . LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

PARTES DE UN DOCUMENTO XML

Como lenguaje de marcado XML, se basa en el concepto de marca o etiqueta.

La etiqueta sirve para caracterizar a un dato y aporta significado al contenido.

Representada por <>

Etiqueta de cierre con barra inclinada

El conjunto de etiqueta de apertura y de cierre, se denomina **elemento**

Los elemento pueden incorporar **atributos** que son características o propiedades que dan valor adicional acerca de una etiqueta

Cada atributo es un par **nombre=valor**.

Los atributos pueden ser obligatorios u opcionales, pero si se usan es necesario que se les asigne un valor y que este se marque con comillas



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

Reglas sintácticas:

XML es un lenguaje sensitivo, es decir, distingue entre mayúsculas y minúsculas

El nombre de un elemento , atributo o entidad, no puede comenzar por. un número o carácter especial, es decir es obligatorio que el nombre de elemento comienza por una letra . A continuación de ella , se puedan poner, letras, dígitos, guiones, rayas, punto, dos puntos

No pueden utilizarse las palabras reservadas XML, xml....como caracteres iniciales del nombre de un atributo, entidad....

El uso de espacios en blanco y los saltos de línea solo se toman en cuenta cuando aparece en el valor de un atributo o cuando se incida su significado de otra forma se omiten

Pueden existir etiquetas sin contenido, lo cual se denomina etiquetas vacías.



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

ELEMENTOS EN XML

La información que hacen entre la etiqueta de inicio y de cierre es lo que llamamos contenido de la etiqueta.

Este contenido puede ser una cadena de caracteres, un valor numérico entero o decimal, una fecha, un valor “booleano”.....

Las etiquetas siempre se deben cerrar, aún con la manera corta

Los elementos pueden contener a otros elementos (se conocen como elementos anidados), y la base de estructura de árbol



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

SINTAXIS Y SEMÁNTICA DE DOCUMENTOS XML: DOCUMENTOS VÁLIDOS Y BIEN FORMADOS

Es posible validar la sintaxis de un XML para determinar si este se encuentra “bien formado”.

Se dice que un documento XML está bien formado si cumple un conjunto de reglas, muchas de las cuales ya hemos dicho:

- Todo documento XML debe tener un elem raíz que no puede duplicarse y que es el primero que se abre y el último que se cierra
- Ningún elemento puede aparecer sin su correspondiente cierre
- Los elementos deben estar anidados correctamente
- Los valores dados a los atributos siempre deben ir entrecomillados
- Para los atributos que sean booleanos, su valor debe explicitarse de manera redundante



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

NAMESPACE

- Un **espacio de nombres XML** es una recomendación W3C para proporcionar elementos y atributos con nombre único en un archivo XML.
- Un archivo XML puede contener nombres de elementos o atributos procedentes de más de un vocabulario XML. Si a cada uno de estos vocabularios se le da un espacio de nombres, un ámbito semántico propio, referenciado a una URL donde se listen los términos que incluye, se resuelve la ambigüedad existente entre elementos o atributos que se llamen igual, la homonimia.
- Se trata de que los nombres de elementos dentro de cada espacio de nombres deben ser únicos.



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

Un ejemplo sería una instancia XML que contuviera referencias a un cliente y a un producto solicitado por éste. Tanto el elemento que representa el cliente como el que representa el producto pueden tener un elemento hijo llamado "numero_ID". Las referencias al elemento "numero_ID" podrían ser ambiguas, salvo que los elementos, con igual nombre pero significado distinto, se llevaran a espacios de nombres distintos que los diferenciaran.

```
<?xml version="1.0"?>
<cli:cliente xmlns:cli='http://es.wikipedia.org/wiki/Espacio_de_nombres_XML/cliente'
             xmlns:ped='http://es.wikipedia.org/wiki/Espacio_de_nombres_XML/pedido'>
  <cli:numero_ID>1232654</cli:numero_ID>
  <cli:nombre>Fulanito de Tal</cli:nombre>
  <cli:telefono>99999999</cli:telefono>
  <ped:pedido>
    <ped:numero_ID>6523213</ped:numero_ID>
    <ped:articulo>Caja de herramientas</ped:articulo>
    <ped:precio>187,90</ped:precio>
  </ped:pedido>
</cli:cliente>
```

TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

Un espacio de nombres se declara usando el atributo XML reservado xmlns, cuyo valor debe ser un identificador uniforme de recurso.

Por ejemplo:

```
xmlns="http://www.w3.org/1999/xhtmll"
```

La declaración puede incluir también un prefijo corto con el que los elementos y atributos pueden identificarse, por ejemplo:

xmlns:xhtml="http://www.w3.org/1999/xhtmll" Un espacio de nombres XML no necesita que su vocabulario sea definido, aunque es una buena práctica utilizar un DTD o un esquema XML para definir la estructura de datos en la ubicación URI del espacio de nombres



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

La tecnología XML busca dar solución al problema de expresar información **estructurada** de la manera más abstracta y reutilizable posible.

Que la información sea **estructurada** quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de trozos de información. Ejemplos son un tema musical, que se compone de compases, que están formados a su vez por notas. Estas partes se llaman *elementos*, y se las señala mediante etiquetas.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento. Un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma <nombre>, donde *nombre* es el nombre del elemento que se está señalando.



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

A continuación se muestra un ejemplo para entender la estructura de un documento XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail>Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```

TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

XML Schema

Es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML.

Se consigue así una percepción del tipo de documento con un nivel alto de abstracción.



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

El término "XML Schema" es utilizado con varios significados dentro del mismo contexto de descripción de documentos, y es importante tener en cuenta las siguientes consideraciones:

- "XML Schema" (Esquema XML) es el nombre oficial otorgado a la recomendación del W3C, que elaboró el primer lenguaje de esquema separado de XML
- Es habitual referirse a los esquemas como "XML schema" de forma genérica, pero se recomienda utilizar el término "documento esquema" (*schema document*) o "definición de esquema" (*schema definition*), y reservar "XML Schema" para la denominación de este lenguaje específico.
- Aunque genéricamente se utilice "XML schemas", XSDL (*XML Schema Definition Language*) es el nombre técnico de los lenguajes de esquema de XML como:



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

ML Schema es un lenguaje de esquema escrito en XML, basado en la gramática y pensado para proporcionar una mayor potencia expresiva que las DTD (definición del tipo de documento)

Los documentos esquema se concibieron como una alternativa a las DTD, más complejas, intentando superar sus puntos débiles y buscar nuevas capacidades a la hora de definir estructuras para documentos XML. El principal aporte de XML Schema es el gran número de tipos de datos que incorpora. De esta manera, XML Schema aumenta las posibilidades y funcionalidades de aplicaciones de procesamiento de datos, incluyendo tipos de datos complejos como fechas, números y strings.

Tipos de componentes

Los esquemas XML Schema superan muchas de las limitaciones y debilidades de las DTDs. soporta tipos de datos típicos de los lenguajes de programación, como también tipos personalizados simples y complejos.



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

La programación en Schema XML se basa en *namespaces* o espacios de nombres

Cada namespace contiene elementos y atributos que están estrechamente relacionados con el namespace. Así, a la hora de definir un elemento o un atributo de un namespace, siempre se creará una conexión entre los diferentes campos de éste. Además, esta forma de trabajar nos permite relacionar elementos que no están en el mismo namespace.

Después de escribir un Schema XML se puede confirmar la correcta realización mediante la validación de esquemas XML: Validación XML.5623



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="0.1" xml:lang="es">
</xsd:schema>
```

Un ejemplo de definición con XML Schema sería el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Libro">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Título" type="xsd:string"/>
        <xsd:element name="Autores" type="xsd:string" maxOccurs="10"/>
        <xsd:element name="Editorial" type="xsd:string"/>
      </xsd:sequence>
      <xsd:attribute name="precio" type="xsd:double"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

ANALISIS Y TRANSFORMACION DE XML

XML ha tenido gran aceptación a la hora de almacenar y transportar información. Pero lo importante no solo es tener la información, sino la capacidad de extraer información así como tener la posibilidad de transformar la información a formatos concretos que sean los que el cliente necesite.

- **búsqueda y extracción de información xquery y xpath**

Para la búsqueda y extracción de información existen dos recomendaciones XQuery y Xpath



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

XQuery es un lenguaje de consulta diseñado para colecciones de datos XML.

Características

- XQuery proporciona los medios para extraer y manipular información de documentos XML, o de cualquier fuente de datos que pueda ser representada mediante XML como, por ejemplo, bases de datos relacionales o documentos ofimáticos.
- XQuery utiliza expresiones XPath para acceder a determinadas partes del documento XML. Añade, además, expresiones similares a las usadas en SQL, conocidas como expresiones **FLWOR**.
- Las expresiones FLWOR toman su nombre de los 5 tipos de sentencias de las que pueden estar compuestas: **FOR**, **LET**, **WHERE**, **ORDER BY** y **RETURN**.
- También incluye la posibilidad de construir nuevos documentos XML a partir de los resultados de la consulta.
- Se puede usar una sintaxis similar a XML si la estructura (elementos y atributos) es conocida con antelación, o usar expresiones de construcción dinámica de nodos en caso contrario. Todos estos constructores se definen como expresiones dentro del lenguaje, y se pueden anidar arbitrariamente.
- El lenguaje se basa en el modelo en árbol de la información contenida en el documento XML, que consiste en siete tipos distintos de nodo: elementos, atributos, nodos de texto, comentarios, instrucciones de procesamiento, espacios de nombres y nodos de documentos.

TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

El sistema de tipos usado por el lenguaje considera todos los valores como secuencias, asumiéndose un valor simple como una secuencia de un solo elemento. Los elementos de una secuencia pueden ser valores atómicos o nodos.

Los valores atómicos pueden ser números enteros, cadenas de texto, valores booleanos, etc. La lista completa de los tipos disponibles está basada en las primitivas definidas en XML Schema.

XQuery 1.0 no incluye capacidad de actualizar los documentos XML. Tampoco puede realizar búsquedas textuales. Estas dos capacidades están siendo objeto de desarrollo para su posible incorporación en la siguiente versión del lenguaje.



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

Xpath

- es un lenguaje que permite construir expresiones que recorren y procesan un documento XML.
- La idea es parecida a las expresiones regulares para seleccionar partes de un texto sin atributos (*plain text*).
- XPath permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del XML.
- XPath fue creado para su uso en el estándar XSLT, en el que se usa para seleccionar y examinar la estructura del documento de entrada de la transformación.
- Todo el procesamiento realizado con un fichero XML está basado en la posibilidad de direccionar o acceder a cada una de las partes que lo componen, de modo que podamos tratar cada uno de los elementos de forma diferenciada.
- El tratamiento del fichero XML comienza por la localización del mismo a lo largo del conjunto de documentos existentes en el mundo. Para llevar a cabo esta localización de forma unívoca, se utilizan los URI (Uniform Resource Identifiers), de los cuales los URL (Uniform Resource Locators) son sin duda los más conocidos.



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

Xpath

Una vez localizado el documento XML, la forma de seleccionar información dentro de él es mediante el uso de XPath, que es la abreviatura de lo que se conoce como XML Path Language. Con XPath podremos seleccionar y hacer referencia a texto, elementos, atributos y cualquier otra información contenida dentro de un fichero XML.

XPath en sí es un lenguaje sofisticado y complejo, pero distinto de los lenguajes procedurales que solemos usar (C, C++, Basic, Java...). Además, como casi todo en el mundo de XML, aún está en estado de desarrollo, por lo que no es fácil encontrar herramientas que incorporen todas sus funcionalidades.



TEMA 8 .LENGUAJE DE MARCAS DE USO COMÚN EN EL LADO SERVIDOR

El modelo de datos de XPath

- Un documento XML es procesado por un analizador (o pares) construyendo un árbol de nodos. Este árbol comienza con un elemento raíz, que se diversifica a lo largo de los elementos que cuelgan de él y acaba en nodos hoja, que contienen solo texto, comentarios, instrucciones de proceso o incluso que están vacíos y solo tienen atributos.
- La forma en que XPath selecciona partes del documento XML se basa precisamente en la representación arbórea que se genera del documento. De hecho, los "operadores" de que consta este lenguaje nos recordarán la terminología que se utiliza a la hora de hablar de árboles en informática: raíz, hijo, ancestro, descendiente, etc.
- Un caso especial de nodo son los nodos atributo.
- Un nodo puede tener tantos atributos como desee, y para cada uno se le creará un nodo atributo.
- No obstante, dichos nodos atributo NO se consideran como hijos suyos, sino más bien como etiquetas añadidas al nodo elemento.



Arquitecturas distribuidas orientadas a servicios

Tema 9

Arquitecturas distribuidas orientadas a servicios

Tanto las arquitecturas Orientadas a Servicio como el Cloud Computing constituye paradigmas punteros y están llamado a ser los protagonistas de los nuevos desarrolladores en el campo de la informática

Ambos proponen una nueva visión de las funciones de los recursos web y buscan crear entornos más dinámicos y más orientados a suplir las necesidades de los usuarios de manera óptima.

En el caso del SOA, su implementación más común son los servicios web.

Arquitecturas distribuidas orientadas a servicios

Sistema distribuido

Un sistema distribuido se define como una colección de computadoras separadas físicamente y conectadas entre sí por una red de comunicaciones;

cada máquina posee sus componentes de hardware y software que el programador percibe como un solo sistema (no necesita saber qué cosas están en qué máquinas).

El programador accede a los componentes de software (objetos) remotos, de la misma manera en que accedería a componentes locales,

Arquitecturas distribuidas orientadas a servicios

Propiedades que debe cumplir el sistema distribuido:

- Transparencia
 - De identificación
 - De la ubicación física de los recursos
 - De replicación
 - De paralelismo
 - De comparación
 - De rendimiento
- Escalabilidad
- Fiabilidad y tolerancia a fallos

Arquitecturas distribuidas orientadas a servicios

- Fiabilidad y tolerancia a fallos
 - Disponibilidad
 - Tolerancia a fallos
 - consistencia

Arquitecturas distribuidas orientadas a servicios

- MODELO CONCEPTUAL DE LAS ARQUITECTURAS ORIENTADAS A SERVICIOS

(**SOA**, siglas del inglés *Service Oriented Architecture*) es un paradigma de arquitectura para diseñar y desarrollar [sistemas distribuidos](#). No hay estándares en relación a la composición exacta de una arquitectura orientada a servicios, aunque muchas fuentes de la industria han publicado sus propios principios.

Permite la creación de [sistemas de información](#) altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios (comúnmente pero no exclusivamente [servicios web](#)), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

Algunos de los principios publicados son los siguientes:

Arquitecturas distribuidas orientadas a servicios

SOA y los Servicios Web[[editar](#)]

Hay que tener cuidado cuando se manejan estos términos y no confundirlos. Web Services (WS) engloba varias tecnologías, incluyendo [XML](#), [SOAP](#), [WSDL](#), [UDDI](#)...los cuales permiten construir soluciones de programación para mensajes específicos y para problemas de integración de aplicaciones.[4](#)

En cambio SOA es una arquitectura de aplicación en la cual todas las funciones están definidas como servicios independientes con interfaces invocables que pueden ser llamados en secuencias bien definidas para formar los procesos de negocio.

Arquitecturas distribuidas orientadas a servicios

En SOA la clave está en la interfaz puesto que define los parámetros requeridos y la naturaleza del resultado. Esto significa que define la naturaleza del servicio y no la tecnología utilizada. Esta función permite realizar dos de los puntos críticos: los servicios son realmente independientes y pueden ser manejados.

WS es el estándar apoyado por la industria ([Microsoft](#), [IBM](#), [BEA](#), [Oracle](#), [Sun](#) y otros), por empresas de distintos rubros, no tecnológicas (Ford, United Airlines, KPMG, Daimler-Chrysler), agrupadas en un comité conocido como *Web Services Interoperability* ([WS-I](#)). Este organismo tiene por principal objetivo asegurar que los grupos de trabajo que definen las especificaciones sobre WS utilizan estándares adecuados, a la vez que monitoriza el avance de sus trabajos; no define ni desarrolla estándares

Arquitecturas distribuidas orientadas a servicios

MANIFIESTO SOA:

Los fundamentos el SOA son:

- La Orientación a Servicio es un paradigma que enmarca lo que usted hace
- La Arquitectura Orientada a Servicios (SOA) es el tipo de arquitectura que resulta de aplicar la orientación a servicios
- Ayuda a las organizaciones a entregar consistentemente valor a los negocios, de manera sostenida, son mayor agilidad y con efectividad en los costos, alineada con las neceases cambiantes de los negocios

Se prioriza:

El balo redel negocio está por encima de la estrategia técnica

Arquitecturas distribuidas orientadas a servicios

Se prioriza:

- El valor del negocio está por encima de la estrategia técnica
- Las metas estratégicas están por encima de los beneficios específicos de los proyectos ,está por encima de la integración personalizada.
- Los servicios compartidos están por encima de las implementaciones de propósito específico
- La flexibilidad está por encima de la optimización
- El refinamiento evolutivo está por encima de la búsqueda de la perfección inicial

Arquitecturas distribuidas orientadas a servicios

CARACTERÍSTICAS DE SOA:

- **OBJETIVO:** que exista una interacción real entre los procesos de negocio, es decir las organizaciones que utilizan recursos TIC ,y las tecnologías. Para ello se crea el concepto de servicio.
- El servicio se trata de un recurso de software que es ofertado para que pueda ser consumido por parte de los usuarios.
- Un servicio no necesariamente debe ser tangible
- Teniendo en cuenta que la aplicación SOA es multidisciplinar, puede analiza sus características desde varios puntos de vista:
 - Desde el punto de vista de negocio
 - Desde el punto de vista de la tecnología

Arquitecturas distribuidas orientadas a servicios

- Desde el punto de vista de negocio SOA ayuda a:
 - Mejorar la flexibilidad y agilidad de los sistemas
 - Proporcionar una visión integrada de los distintos
 - Mejorar la cobertura de las necesidades de negocio
 - Reducir el impacto de la evolución de la tecnología en las aplicaciones de negocio
- Desde el punto de vista de la tecnología
 - SOA favorece la reutilización y la reducción del tipo
 - Mejora la productividad de los procesos
 - Favorece el proceso de construcción de software
 - Favorece la forma en la que el usuario interactúa con las aplicaciones

Arquitecturas distribuidas orientadas a servicios

CARACTERÍSTICAS DE LOS SERVICIOS

Características de SOA	
Sin SOA	Con SOA
Orientado a Función	Orientado a Procesos
Construido para Durar	Construido para Cambiar
Ciclo de Desarrollos Largos	Ciclos de Desarrollo Incrementales
Aplicaciones Aisladas	Aplicaciones Orquestadas
Fuerte Acoplamiento	Bajo Acoplamiento
Orientada a Objetos	Orientado a Mensajes

Arquitecturas distribuidas orientadas a servicios

CARACTERÍSTICAS DE LOS SERVICIOS

Los servicios deben ser reusables. Para facilitar esta reutilización los servicios deben proporcionar un contrato forma en el que se estipula cual es el nombre del servicio, su forma de acceso, las funcionalidades que ofrece, los datos de entrada de cada una de las funcionalidades y los datos de salida

Deben ser bajamente acoplados. Es decir independientes los unos de los otros. Hay que evitar las dependencias cruzadas de servicios . Este bajo acoplamiento favorece a que sean reutilizables

Los servicios deben encapsular su lógica subyacente

Los servicios deben permitir la composición

Arquitecturas distribuidas orientadas a servicios

Los servicios deben ser autónomos. Un servicio es dueño de su entorno de ejecución y por lo tanto tiene control sobre los recursos subyacentes del mismo

Los servicios no deben tener estado. Un servicio no debe guardar ningún tipo de información

Los servicios deben poder ser descubiertos de alguna forma para que pueda ser utilizado , consiguiendo así evitar la creación accidental de servicios que proporcionen la mismas funcionalidades .

En caso de los servicios web, el descubrimiento se logra publicando los interfaces de los servicios en registros UDDI

Arquitecturas distribuidas orientadas a servicios

CICLO DE VIDA DE LOS SERVICIOS:

El desarrollo de servicios conlleva un ciclo de vida. Sin embargo este ciclo de vida es diferente si quien lo mira es el proveedor o el consumidor del servicio.

El proveedor del servicio es quien crea y oferta el servicio, por tanto su visión del ciclo de vida se centra primero en identificar una necesidad que puede ser solventada por el servicio.

A partir de ahí diseña, desarrolla e implementa el servicio para que este pueda ser ofertado a partir de un contrato formal.

El ciclo de vida desde el punto de vista del consumidor: Comienza cuando detecta la necesidad de servicio. A partir de esta necesidad busca cual puede ser el servicio que mejor satisfaga dichas necesidades. Una vez elegido el servicio, lo contrata y a partir de ahí lo disfruta durante el tiempo que se haya establecido que se haga la prestación del servicio.

Arquitecturas distribuidas orientadas a servicios

IMPLEMENTACION DE ARQUITECTURAS ORIENTADAS A SERVICIOS MEDIANTE TECNOLOGIAS WEB. SERVICIOS WEB:

Un servicio web es un modulo software auto-contenido (independiente) y auto-descrito disponible a través de un arde (como internet), que completa tareas, resuelve problemas o realiza transacciones de acuerdo a las peticiones de un usuario o aplicaciones

ROLES Y OPERACIÓN DE UN SERVICIO EN SOA



Este esquema de trabajo genera un conjunto de operaciones que se denominan las operaciones básicas de SOA. Estas operaciones son : publicar, encontrar y vincular e interactuar

Arquitecturas distribuidas orientadas a servicios

LENGUAJES Y TECNOLOGIAS DE SERVICIOS WEB

una de las grandes ventajas de los servicios web es que se basan en estándares. Dichos estándares se dividen en estándares básicos y estándares principales

Los estándares básicos son aquellos que no han sido creados exclusivamente para el desarrollo de servicio web pero que son los pilares para el buen funcionamiento de estos. Estos protocolos son HTTP como protocolo de transporte y XML como estándar de intercambio de mensajes

Los estándares principales son aquellos creados y pensados exclusivamente en el desarrollo de servicios web.

Arquitecturas distribuidas orientadas a servicios

Los tres estándares Core son:

- SOAP protocolo de mensajes basado en XML y utilizado para intercambiar información entre servicios web. Implementa un modelo de comunicación pregunta/respuesta
- WSDL forma de especificar los datos, operaciones, contratos y funcionalidades ofrecidas por un servicio web. Su gramática está basada en XML y describe los servicios web como colecciones de endpoints (puntos remotos) capaces de intercambiar mensajes
- UDDI directorio público que permite la publicación online de servicios a través de al descripciones de los servicios (WSDL) y su eventual descubrimiento.

Arquitecturas distribuidas orientadas a servicios

CALIDAD DE SERVICIO

Dentro del área de servicios web , la calidad de servicio se definió como la habilidad para responder a las invocaciones esperadas y a la realizadas a nivel exigido y acordado por el proveedor y sus consumidores.

La calidad del servicio se puede referir a muchos atributos y de acuerdo al tipo de servicio ofrecido y al contexto en que este es ofrecido , algunos atributos pueden ser más relevantes que otros.

Algunos de los aspectos más importantes dentro de la calidad del servicio son:

- Disponibilidad
- Accesibilidad: tasa de éxito de invocación a un servicio
- Adecuación a estándares grado de cumplimiento de un estándar, especificado en el Acuerdo de Nivel de Servicios
- Integridad grado en el que un servicio web realiza sus tareas de acuerdo a su descripción WSDL y a los acuerdos especificados en el contrato SLA

Arquitecturas distribuidas orientadas a servicios

Rendimiento: evalúa la relación entre el servicio ofertado y los recursos utilizados para ofertarlo. En este caso podemos evaluar dos variables o números de peticiones servidas en un tiempo determinado y la latencia o tiempo que pasa desde que se envía una petición y se recibe la respuesta

Fiabilidad: Habilidad de un servicio para funcionar correctamente y consistentemente ofreciendo la misma calidad de servicio independientemente de fallos de sistema o red

Escalabilidad: habilidad para servir peticiones a pesar de las variaciones en el volumen de peticiones

Seguridad: aspectos tales como autenticación, autorización, integridad de mensaje y confidencialidad

Arquitecturas distribuidas orientadas a servicios

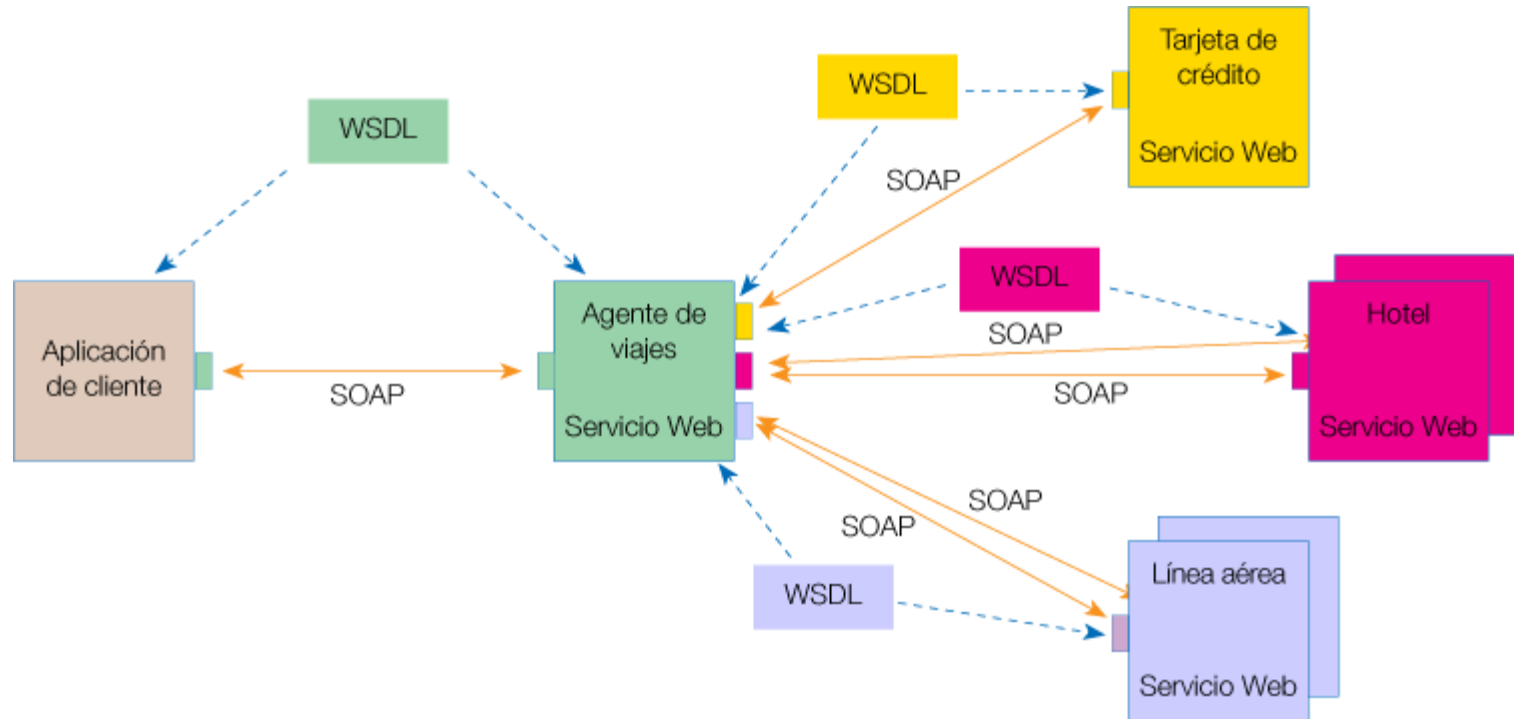
ESPECIFICACIONES DE SERVICIOS WEB DE USO COMUN: SOAP, REST, ETC

Con el fin de nuevas propuestas para mejora de servicios web, surgen las discusiones entre SOAP y REST . Cada uno de estos estándares proporciona una forma de intercambiar información y establecer cual es mejor

- SOAP
Protocolo simple de acceso a Objetos provee un mecanismo simple y ligero para intercambio de información estructurada y tapeada en un ambiente descentralizado y distribuido, usando XML.
- SOAP indica como se deben codificar los mensaje que circularan entre el emisor y el receptor. La especificación SOAP define dos modelos de mensajes
 - Mensaje petición: Se envía desde la aplicación del consumidor a la aplicación del proveedor , solicitando la ejecución de un método a través de una serie de parámetros
 - Mensaje respuesta: Se envía dedo la aplicación de proveedor al consumidor y contiene datos en XML con los resultados de la ejecución del método solicitado

Arquitecturas distribuidas orientadas a servicios

Imagen de intercambio de mensajes usando SOAP



Arquitecturas distribuidas orientadas a servicios

LENGUAJES DE DEFINICION E SERVICIOS: EL ETANDAR WSDL

Una de las características principales de un servicio web es que este debe poseer una descripción del servicio. Por tanto, un lenguaje de definición de un servicio que sirve para construir dicha descripción debe servir para resolver dudas del tipo ¿Qué puede hacer el servicio? ¿Dónde reside? ¿Cómo invocarlo? Permitiendo describir los distintos métodos o funciones que están disponibles en un servicio web

Para suplir esta necesidad se ha creado el estándar WSDL. El lenguaje de Descripción de Servicios web (WSDL) busca liberar a los usuarios y desarrolladores de la necesidad de recordar o entender todos los detalles de acceso a los servicios. Solo es necesario una descripción WSDL para poder acceder a estos.

Arquitecturas distribuidas orientadas a servicios

Para realizar la descripción de un servicio WSDL separa la descripción abstracta, métodos y o funciones que posee el servicio de la descripción concreta, es decir, el protocolo concreto de conexión utilizado y el formato del mensaje.

Para lograr esta independencia entre la descripción abstracta y concreta, WSDL, describe de manera independiente tanto los métodos y o funciones del servicio como los tipo de datos usados en los mensajes.

Además, distingue claramente los mensajes de los puertos, donde los mensajes poseen la sintaxis y semántica de los mensajes que espera un servicio web y los puertos poseen las dirección de red en las que se invoca al servicio web

Arquitecturas distribuidas orientadas a servicios

Existen tres componentes principales para definir la interfaz abstracta de un servicio:

- El vocabulario: para definir el vocabulario que será utilizado por el servicio se definen elementos tipos , que son los tipos de datos utilizados en los mensajes. Por ejemplo estos mensajes pueden estar definidos en un XML.
- Los mensajes: se definen a través de elementos de tipo mensaje. Esta es una definición abstracta y estricta de los datos que son transmitidos. Por tanto, los mensajes de petición que se esperan como los mensajes de respuesta que emite el servicio web
- La interacción: explica exactamente lo que hace el servicio. Los elementos utilizados para esto son
 - Operación en donde se describe de forma abstracta una acción admitida por el servicio
 - Porttype que representa un conjunto de operaciones abstractas admitidas por uno o más puntos finales

Arquitecturas distribuidas orientadas a servicios

DIRECTORIOS DE SERVICIOS

Una vez creado el servicio web, el siguiente paso consiste en definir cómo se dará a conocer el servicio web para que los clientes interesados puedan descubrirlo fácilmente y utilizarlo en sus aplicaciones

Debido a la gran aceptación de la tecnología SOA, el número de proveedores de servicios web aumenta y con ello aumenta el número de servicios web disponibles para los usuarios.

Por tanto, se necesita un sistema de referencia que facilite a los consumidores las tareas de búsqueda y localización de estos servicios. Esta es la idea de un directorio de servicios.

Aunque existen otras implementaciones de directorios como WSIL y ebWML, aquí nos centraremos en UDDI que es el estándar de directorio de servicios web por W3C

Arquitecturas distribuidas orientadas a servicios

ESTANDARES SOBRE DIRECTORIOS DE SERVICIOS : UDDI

Es un registro publico diseñado para almacenar de forma estructurada información sobre proveedores y los servicios web que estos ofrecen

El objetivo de UDDI es proporcionar toda la información que pueda ser valiosa para un consumidor a la hora de buscar y elegir un servicio.

Por ejemplo para unos usuario el precio puede ser lo más relevante a la hora de elegir un servicio mientras que para otros lo más importante puede ser el proveedor que oferte el servicio. . Es por ello que Addis posee información sobre los proveedores de servicios, donde estos pueden ser empresas o particulares, sobre los servicios que estos proveedores exponen, sobre los enlace o localizaciones (URI) en los que pueden encontrar dichos servicios, así como de las interfaces soportadas por los mismos

Arquitecturas distribuidas orientadas a servicios

A nivel técnico un directorio UDDI esta basado en XML y codifica tres tipos de información acerca de un servicio web.

Paginas blancas: almacenan información de contacto sobre el proveedor como por ejemplo dirección email de contacto, paginas web y otros identificadores conocidos

Paginas amarillas: provee información sobre todos los servicios web que ofrece un determinado proveedor categoriza dando los servicios. La categorización puede ser de acuerdo al tipo de industria al que está enfocado el sitio web .

Por ejemplo se pueden listar todos los servicios web de un proveedor que puedan ser aplicados al sector de la salud o todos los que tengan que ver con el sector de la educación

Paginas verdes contienen la información técnica sobre los servicios que aportan las propias empresas, así como las especificaciones de los mismos

Arquitecturas distribuidas orientadas a servicios

Cada tipo de información esta asociada a una sección y cada sección tendrá un tipo de entidad (a nivel técnico) que permite almacenar dicha información.

Sección blanca : su elemento o estructura básica es de tipo `bussinessEntity`. Este elemento describe una familia de servicio web ofrecidos por el proveedor descrita en el `bussineessEntitiy`

Sección amarilla: se utiliza el elemento o estructura tipo `bussinessService`. Este elemento describe una familia a de servicios web ofrecido por el proveedor descrito en el `bussinessEntity`

Sección verde: Se usan los elementos o estructuras tipo `bindingTemplate` las cuales se encargan de guardar la información técnica de acceso a un servicio Web concreto, como por ejemplo, la dirección donde se puede acceder al servicio Un `binding template` siempre posee un elemento que es el que , de manera concreta, establece cual es la interfaz del servicio, el protocolo usado por el servicio web y la categoría al que este pertenece .

Una vez definido el UDDI, es importante poder realizar búsquedas así como el UDDI conteste al usuario . El estándar establece que SOAP es el protocolo de mensajería utilizado por el usuario para comunicarse con el UDDI de esta forma tanto la petición como la respuesta deben ser dadas utilizando este protocolo. Esto significa que el UDDI debe tener la capacidad de responder usando SOAP

Programación de servicios web en entornos distribuidos

Las arquitecturas orientadas a servicios han revolucionado la forma en la que se construyen las nuevas soluciones de software.

Esta nueva aproximación se basa en crear aplicaciones altamente escalables con un bajo acoplamiento en donde se ofrece a los consumidores un servicio que debe aportarles valor.

La principal implementación de SOA son los servicios web, los cuales se basan en un conjunto de estándares básicos y principales tales como HTTP, XML, SOAP, WSDL y UDDI.

Los frameworks de programación actuales permiten a los proveedores la creación de servicios web así como su montaje en distintos contenedores que permitan que dichos servicios puedan ser ofertados a los consumidores.

Programación de servicios web en entornos distribuidos

TIPOS DE SERVICIOS A IMPLEMENTAR DENTRO DE UNA ARQUITECTURA SOA

Las arquitecturas Orientadas a Servicios , no solo están enfocadas a nuevos desarrollos de modelos de negocio.

Por ellos, existen distintos tipo de servicios que pueden ser implementados dentro de un modelo SOA:

- Servicios derivados de la adaptación de sistemas legados. Estos servicios buscan proveer interfaces para las funcionalidades e información almacenada en el sistema legado puede ser utilizado en la nueva arquitectura.
- Servicios ofrecidos por aplicaciones software . Estos servicios son nuevos desarrollos diseñados e implementados siguiendo la orientación a servicios y cuyos consumidores no son clientes externos sino clientes internos de la organización
- Servicios ofrecidos por un proveedor externo a través de un software . EN este caso, el proveedor y el consumidor de los servicios son entidades totalmente independientes

Programación de servicios web en entornos distribuidos

El software como servicio es una de las nuevas tendencias en desarrollo que se soportan en la nube.

A diferencia del modelo de licencias habitual del software que se instala en las empresas el acceso a las aplicaciones SAAS Se suele, basar en un modelo de suscripción Donde los cliente pagan una tarifa por adelantado para utilizarlas. Este tipo de software no es necesario adquirirlo, instalarlo, configurarlo y mantenerlo. Solo es necesario suscribirse y utilizarlo.

Una vez se finaliza el periodo de suscripción el servicio deja de estar disponible

Programación de servicios web en entornos distribuidos

ESCENARIOS DE IMPLEMENTACION SOA

Existen dos roles involucrados en un interacción orientada a servicios: el consumidor y el proveedor del servicio. Mientras que el proveedor de servicios es aquel que desarrolla y se encarga del mantenimiento del servicio, el consumidor es aquel que se encarga de utilizar el servicio durante el periodo de suscripción convenido por el proveedor

Se pueden tener tres tipos de escenario de implementación de SOA:

1. Escenario A: consumidor del servicio: A partir del WSDL, el consumidor podrá establecer cual es el punto de acceso al servicio, las funcionalidades ofrecidas y los mensajes que espera el servicio web para que este pueda ser ejecutado y devuelva una respuesta
2. Escenario B: proveedor del servicio: Este escenario implica el desarrollo total del servicio. Dicho desarrollo engloba tres componentes:
 1. La infraestructura SOA
 2. La interfaz del servicio y
 3. la aplicación empresarial
3. Escenario C: implantación total

Programación de servicios web en entornos distribuidos

1. Escenario B: proveedor del servicio: Este escenario implica el desarrollo total del servicio. Dicho desarrollo engloba tres componentes:
 1. La infraestructura SOA
 2. La interfaz del servicio y
 3. la aplicación empresarial
2. Escenario C: implantación total. En este caso, quien desarrolla el servicio debe cumplir los dos roles, proveedor del servicio y consumidor del mismo.
Este escenario se puede presentar si los servicios con desarrollados por un departamento de informática para clientes internos