

**INSTITUTO TECNOLÓGICO DE OAXACA**

INSTITUTO TECNOLÓGICO DE OAXACA

DE OAXACA

SUBSECRETARÍA DE EDUCACIÓN SUPERIORDIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA

***ING. EN SISTEMAS COMPUTACIONALES***

**INVESTIGACION**

**DSOS**

**15:00-16:00 HRS**

**PRESENTA***:*

HERNÁNDEZ JIMENEZ ALAN FIDEL

OAXACA DE JUÁREZ, OAX. 04/ FEBERRO / 2019

**Servidor web:**

Un servidor Web es un programa que utiliza el protocolo de transferencia de hiper texto, HTTP (Hypertext Transfer Protocol), para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP de sus computadoras. Las computadoras y los dispositivos dedicados también pueden denominarse servidores Web.

El proceso es un ejemplo del modelo cliente/servidor. Todos los equipos que alojan sitios Web deben tener programas de servidor Web. Los principales servidores Web incluyen Apache (el servidor Web más ampliamente instalado), Internet Information Server (IIS) de Microsoft y nginx (que se pronuncia engine X) de NGNIX. Otros servidores Web incluyen el servidor NetWare de Novell, el servidor Web de Google (GWS) y la familia de servidores Domino de IBM.

Los servidores Web a menudo forman parte de un paquete más amplio de programas relacionados con internet e intranet para servir correo electrónico, descargar solicitudes de archivos de protocolo de transferencia de archivos (FTP) y crear y publicar páginas Web. Las consideraciones al elegir un servidor Web incluyen cuán bien funciona con el sistema operativo y otros servidores, su capacidad para manejar la programación del servidor, las características de seguridad y las herramientas particulares de publicación, motor de búsqueda y creación de sitios que vienen con él.

**SOAP**

(originalmente las siglas de 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.

Básicamente SOAP es un paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más complejos y completos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "pila de protocolos de web service", ofreciendo un framework de mensajería básica en el cual los web services se pueden construir.

Este protocolo está basado en XML y se conforma de tres partes:

* Sobre (envelope): el cual define qué hay en el mensaje y cómo procesarlo.
* Conjunto de reglas de codificación para expresar instancias de tipos de datos.
* La Convención para representar llamadas a procedimientos y respuestas.

El protocolo SOAP tiene tres características principales:

* Extensibilidad (seguridad y WS-routing son extensiones aplicadas en el desarrollo).
* Neutralidad (bajo protocolo de transporte TCP puede ser utilizado sobre cualquier protocolo de aplicación como HTTP, SMTP o JMS).
* Independencia (permite cualquier modelo de programación).

**AJAX**

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor.

JavaScript es un lenguaje de programación (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Ajax es una combinación de cuatro tecnologías ya existentes:

* [XHTML](https://es.wikipedia.org/wiki/XHTML) (o [HTML](https://es.wikipedia.org/wiki/HTML)) y [hojas de estilos en cascada](https://es.wikipedia.org/wiki/Hojas_de_estilos_en_cascada) (CSS) para el diseño que acompaña a la información.
* [Document Object Model](https://es.wikipedia.org/wiki/Document_Object_Model) (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript" \o "ECMAScript) como [JavaScript](https://es.wikipedia.org/wiki/JavaScript) y [JScript](https://es.wikipedia.org/wiki/JScript" \o "JScript), para mostrar e interactuar dinámicamente con la información presentada.
* El objeto [XMLHttpRequest](https://es.wikipedia.org/wiki/XMLHttpRequest" \o "XMLHttpRequest) para intercambiar datos de forma asíncrona con el servidor web. En algunos [frameworks](https://es.wikipedia.org/wiki/Framework" \o "Framework) y en algunas situaciones concretas, se usa un objeto [iframe](https://es.wikipedia.org/wiki/Iframe" \o "Iframe) en lugar del XMLHttpRequest para realizar dichos intercambios. [PHP](https://es.wikipedia.org/wiki/PHP) es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico también utilizado en el método Ajax.
* [XML](https://es.wikipedia.org/wiki/XML) es el [formato](https://es.wikipedia.org/wiki/Formato_de_archivo_inform%C3%A1tico) usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, [JSON](https://es.wikipedia.org/wiki/JSON) y hasta [EBML](https://es.wikipedia.org/wiki/EBML).

Como el [DHTML](https://es.wikipedia.org/wiki/DHTML), [LAMP](https://es.wikipedia.org/wiki/LAMP) o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

**ANGULAR**

Angular (comúnmente llamado "Angular 2+" o "Angular 2"), es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o ser recuperados de recursos JSON estáticos o dinámicos.

Angular es la evolución de AngularJS aunque incompatible con su predecesor.

Angular (comúnmente llamado "Angular 2+" o "Angular 2"), es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o ser recuperados de recursos JSON estáticos o dinámicos.

Angular es la evolución de AngularJS aunque incompatible con su predecesor.

Principales características de Angular:

Velocidad y rendimiento

* Generación de código: Angular convierte tus plantillas en código altamente optimizado para las máquinas virtuales de JavaScript de hoy en día, ofreciéndote todas las ventajas del código escrito a mano con la productividad de un framework.
* Universal: Ejecuta la primera vista de tu aplicación en node.js, .NET, PHP, y otros servidores para renderizado de forma casi instantánea obteniendo solo HTML y CSS. También abre posibilidades para la optimización del SEO del sitio.
* División del código: Las aplicaciones de Angular se cargan rápidamente gracias al nuevo enrutador de componentes. Éste ofrece una división automática de códigos para que los usuarios sólo carguen el código necesario para procesar la vista que solicitan.

Productividad

* Plantillas: Permite crear rápidamente vistas de interfaz de usuario con una sintaxis de plantilla simple y potente.
* Angular CLI: Las herramientas de línea de comandos permiten empezar a desarrollar rápidamente, añadir componentes y realizar test, así como pre visualizar de forma instantánea la aplicación.
* IDEs: Obtén sugerencias de código inteligente, detección de errores y otros comentarios en la mayoría de los editores populares e IDEs.

Historia completa del desarrollo

* Testing: Utiliza Karma para realizar pruebas unitarias, y Pro tractor para realizar pruebas end-to-end de forma rápida y estable.
* Animación: Permite crear animaciones complejas y de alto rendimiento con muy poco código a través de la intuitiva API de Angular.
* Accesibilidad: Posee características para crear aplicaciones accesibles con los componentes disponibles para ARIA.

**JQUERY**

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados.​ jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Las empresas Microsoft y Nokia anunciaron que incluirán la biblioteca en sus plataformas.​ Microsoft la añadirá en su IDE Visual Studio4​ y la usará junto con los frameworks ASP.NET AJAX y ASP.NET MVC, mientras que Nokia los integrará con su plataforma Web Run-Time.

Características

* Selección de elementos DOM.
* Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
* Eventos.
* Manipulación de la hoja de estilos CSS.
* Efectos y animaciones.
* Animaciones personalizadas.
* AJAX.
* Soporta extensiones.
* Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
* Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.5​

**JAVASCRIPT**

**j**avaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, ​ basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas​, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de javascript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio del 2015.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Desde el lanzamiento en junio de 1997 del estándar ECMAScript 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó​). En junio de 2015 se cerró y publicó la versión ECMAScript 6.

Características

Las siguientes características son comunes a todas las implementaciones que se ajustan al estándar ECMAScript, a menos que especifique explícitamente en caso contrario.

Imperativo y estructurado

* JavaScript es compatible con gran parte de la estructura de programación de C (por ejemplo, sentencias if, bucles for, sentencias switch, etc.). Con una salvedad, en parte: en C, el ámbito de las variables alcanza al bloque en el cual fueron definidas; sin embargo JavaScript no es compatible con esto, puesto que el ámbito de las variables es el de la función en la cual fueron declaradas. Esto cambia con la versión de ECMAScript 2015, ya que añade compatibilidad con block scoping por medio de la palabra clave let. Como en C, JavaScript hace distinción entre expresiones y sentencias. Una diferencia sintáctica con respecto a C es la inserción automática de punto y coma, es decir, en JavaScript los puntos y coma que finalizan una sentencia pueden ser omitidos.

Dinámico

Tipado dinámico

* Como en la mayoría de lenguajes de scripting, el tipo está asociado al valor, no a la variable. Por ejemplo, una variable x en un momento dado puede estar ligada a un número y más adelante, religada a una cadena. JavaScript es compatible con varias formas de comprobar el tipo de un objeto, incluyendo duck typing.​ Una forma de saberlo es por medio de la palabra clave typeof.

Objetual

* JavaScript está formado casi en su totalidad por objetos. Los objetos en JavaScript son arrays asociativos, mejorados con la inclusión de prototipos (ver más adelante). Los nombres de las propiedades de los objetos son claves de tipo cadena: obj.x = 10 y obj['x'] = 10 son equivalentes, siendo azúcar sintáctico la notación con punto. Las propiedades y sus valores pueden ser creados, cambiados o eliminados en tiempo de ejecución. La mayoría de propiedades de un objeto (y aquellas que son incluidas por la cadena de la herencia prototípica) pueden ser enumeradas a por medio de la instrucción de bucle for... in. JavaScript tiene un pequeño número de objetos predefinidos como son Function y Date.

Evaluación en tiempo de ejecución

* JavaScript incluye la función eval que permite evaluar expresiones expresadas como cadenas en tiempo de ejecución. Por ello se recomienda que eval sea utilizado con precaución y que se opte por utilizar la función JSON.parse() en la medida de lo posible, pues resulta mucho más segura.

Funciones de primera clase

* A las funciones se les suele llamar ciudadanos de primera clase; son objetos en sí mismos. Como tal, poseen propiedades y métodos, como. call() y .bind().Una función anidada es una función definida dentro de otra. Esta es creada cada vez que la función externa es invocada. Además, cada función creada forma una clausura; es el resultado de evaluar un ámbito conteniendo en una o más variables dependientes de otro ámbito externo, incluyendo constantes, variables locales y argumentos de la función externa llamante. El resultado de la evaluación de dicha clausura forma parte del estado interno de cada objeto función, incluso después de que la función exterior concluya su evaluación.

Prototipos

* JavaScript usa prototipos en vez de clases para el uso de herencia. ​ Es posible llegar a emular muchas de las características que proporcionan las clases en lenguajes orientados a objetos tradicionales por medio de prototipos en JavaScript.

Funciones como constructores de objetos

* Las funciones también se comportan como constructores. Prefijar una llamada a la función con la palabra clave new crear una nueva instancia de un prototipo, que heredan propiedades y métodos del constructor (incluidas las propiedades del prototipo de Object).​ ECMAScript 5 ofrece el método Object.create, permitiendo la creación explícita de una instancia sin tener que heredar automáticamente del prototipo de Object (en entornos antiguos puede aparecer el prototipo del objeto creado como null).​ La propiedad prototype del constructor determina el objeto usado para el prototipo interno de los nuevos objetos creados. Se pueden añadir nuevos métodos modificando el prototipo del objeto usado como constructor. Constructores predefinidos en JavaScript, como Array u Object, también tienen prototipos que pueden ser modificados. Aunque esto sea posible se considera una mala práctica modificar el prototipo de Object ya que la mayoría de los objetos en Javascript heredan los métodos y propiedades del objeto prototype, objetos los cuales pueden esperar que estos no hayan sido modificados.

**BOOTSTRAP**

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

Bootstrap es el segundo proyecto más destacado en GitHub​ y es usado por la NASA y la MSNBC entre otras organizaciones.

Características

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores. Por ejemplo, las propiedades introducidas en CSS3 para las esquinas redondeadas, gradientes y sombras son usadas por Bootstrap a pesar de la falta de soporte de navegadores antiguos. Esto extiende la funcionalidad de la herramienta, pero no es requerida para su uso.

Desde la versión 2.0 también soporta diseños web adaptables. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles).

Bootstrap es de código abierto y está disponible en GitHub. Los desarrolladores están motivados a participar en el proyecto y a hacer sus propias contribuciones a la plataforma.

**WSDL**

WSDL, las siglas de Web Services Description Language, es un formato del Extensible Markup Language (XML) que se utiliza para describir servicios web (WS). La versión 1.0 fue la primera recomendación por parte del W3C y la versión 1.1 no alcanzó nunca tal estatus. La versión 2.0 se convirtió en la recomendación actual por parte de dicha entidad.

WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

Así, WSDL se usa a menudo en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

El WSDL nos permite tener una descripción de un servicio web. Especifica la interfaz abstracta a través de la cual un cliente puede acceder al servicio y los detalles de cómo se debe utilizar.

La estructura del WSDL tiene los siguientes elementos:

Tipos de datos

<types>: Esta sección define los tipos de datos usados en los mensajes. Se utilizan los tipos definidos en la especificación de esquemas XML.

Mensajes

<message>: Aquí definimos los elementos de mensaje. Cada mensaje puede consistir en una serie de partes lógicas. Las partes pueden ser de cualquiera de los tipos definidos en la sección anterior.

Tipos de puerto

<portType>: Con este apartado definimos las operaciones permitidas y los mensajes intercambiados en el Servicio.

Bindings

<binding>: Especificamos los protocolos de comunicación usados.

Servicios

<service>: Conjunto de puertos y dirección de los mismos. Esta parte final hace referencia a lo aportado por las secciones anteriores.

Con estos elementos no sabemos qué hace un servicio pero sí disponemos de la información necesaria para interactuar con él (funciones, mensajes de entrada/salida, protocolos...)

EJEMPLOS

<definitions name = "HelloService"

targetNamespace = "http://www.examples.com/wsdl/HelloService.wsdl"

xmlns = "http://schemas.xmlsoap.org/wsdl/"

xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:tns = "http://www.examples.com/wsdl/HelloService.wsdl"

xmlns:xsd = "http://www.w3.org/2001/XMLSchema">

<message name = "SayHelloRequest">

<part name = "firstName" type = "xsd:string"/>

</message>

<message name = "SayHelloResponse">

<part name = "greeting" type = "xsd:string"/>

</message>

<portType name = "Hello\_PortType">

<operation name = "sayHello">

<input message = "tns:SayHelloRequest"/>

<output message = "tns:SayHelloResponse"/>

</operation>

</portType>

<binding name = "Hello\_Binding" type = "tns:Hello\_PortType">

<soap:binding style = "rpc"

transport = "http://schemas.xmlsoap.org/soap/http"/>

<operation name = "sayHello">

<soap:operation soapAction = "sayHello"/>

<input>

<soap:body

encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/"

namespace = "urn:examples:helloservice"

use = "encoded"/>

</input>

<output>

<soap:body

encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/"

namespace = "urn:examples:helloservice"

use = "encoded"/>

</output>

</operation>

</binding>

<service name = "Hello\_Service">

<documentation>WSDL File for HelloService</documentation>

<port binding = "tns:Hello\_Binding" name = "Hello\_Port">

<soap:address

location = "http://www.examples.com/SayHello/" />

</port>

</service>

</definitions>

 En este ejemplo concreto se implementa un servicio que muestra a partir del nombre de un valor bursátil su valor actual en el mercado.

**<definitions** name="StockQuote"

targetNamespace="http://example.com/stockquote.wsdl"

xmlns:tns="http://example.com/stockquote.wsdl"

xmlns:xsd1="http://example.com/stockquote.xsd"

xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns="http://schemas.xmlsoap.org/wsdl/"**>**

**<types>**

**<schema** targetNamespace="http://example.com/stockquote.xsd"

xmlns="http://www.w3.org/2000/10/XMLSchema"**>**

**<element** name="TradePriceRequest"**>**

**<complexType>**

**<all>**

**<element** name="tickerSymbol" type="string"**/>**

**</all>**

**</complexType>**

**</element>**

**<element** name="TradePrice"**>**

**<complexType>**

**<all>**

**<element** name="price" type="float"**/>**

**</all>**

**</complexType>**

**</element>**

**</schema>**

**</types>**

**<message** name="GetLastTradePriceInput"**>**

**<part** name="body" element="xsd1:TradePriceRequest"**/>**

**</message>**

**<message** name="GetLastTradePriceOutput"**>**

**<part** name="body" element="xsd1:TradePrice"**/>**

**</message>**

**<portType** name="StockQuotePortType"**>**

**<operation** name="GetLastTradePrice"**>**

**<input** message="tns:GetLastTradePriceInput"**/>**

**<output** message="tns:GetLastTradePriceOutput"**/>**

**</operation>**

**</portType>**

**<binding** name="StockQuoteSoapBinding" type="tns:StockQuotePortType"**>**

**<soap:binding** style="document" transport="http://schemas.xmlsoap.org/soap/http"**/>**

**<operation** name="GetLastTradePrice"**>**

**<soap:operation** soapAction="http://example.com/GetLastTradePrice"**/>**

**<input>**

**<soap:body** use="literal"**/>**

**</input>**

**<output>**

**<soap:body** use="literal"**/>**

**</output>**

**</operation>**

**</binding>**

**<service** name="StockQuoteService"**>**

**<documentation>**My first service**</documentation>**

**<port** name="StockQuotePort" binding="tns:StockQuoteSoapBinding"**>**

**<soap:address** location="http://example.com/stockquote"**/>**

**</port>**

**</service>**

**</definitions>**

**JSON**

JSON (acrónimo de JavaScript Object Notation, «notación de objeto de JavaScript») es un formato de texto ligero para el [intercambio de datos](https://es.wikipedia.org/w/index.php?title=Intercambio_de_datos&action=edit&redlink=1). JSON es un subconjunto de la notación literal de [objetos](https://es.wikipedia.org/wiki/Objeto_(programaci%C3%B3n)) de [JavaScript](https://es.wikipedia.org/wiki/JavaScript) aunque hoy, debido a su amplia adopción como alternativa a [XML](https://es.wikipedia.org/wiki/XML), se considera un formato de lenguaje independiente.

Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un [analizador sintáctico](https://es.wikipedia.org/wiki/Analizador_sint%C3%A1ctico) (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función [eval](https://es.wikipedia.org/wiki/Eval)(), lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores [AJAX](https://es.wikipedia.org/wiki/AJAX), debido a la ubicuidad de JavaScript en casi cualquier [navegador web](https://es.wikipedia.org/wiki/Navegador_web).

En la práctica, los argumentos a favor de la facilidad de desarrollo de analizadores o de sus rendimientos son poco relevantes, debido a las cuestiones de seguridad que plantea el uso de eval() y el auge del procesamiento nativo de XML incorporado en los navegadores modernos. Por esa razón, JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia (de aquí su uso por Yahoo, Google, etc, que atienden a millones de usuarios) cuando la fuente de datos es explícitamente de fiar y donde no es importante el no disponer de procesamiento [XSLT](https://es.wikipedia.org/wiki/XSLT) para manipular los datos en el cliente.

Sintaxis

Los tipos de datos disponibles con JSON son:

* Números: Se permiten números negativos y opcionalmente pueden contener parte fraccional separada por puntos. Ejemplo: 123.456
* Cadenas: Representan secuencias de cero o más caracteres. Se ponen entre doble comilla y se permiten cadenas de escape. Ejemplo: "Hola"
* [Booleanos](https://es.wikipedia.org/wiki/Tipo_de_dato_l%C3%B3gico): Representan valores booleanos y pueden tener dos valores: true y false
* null: Representan el valor nulo.
* [Array](https://es.wikipedia.org/wiki/Vector_(inform%C3%A1tica)): Representa una lista ordenada de cero o más valores los cuales pueden ser de cualquier tipo. Los valores se separan por comas y el vector se mete entre corchetes. Ejemplo ["juan","pedro","jacinto"]
* Objetos: Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena y entre ellas. El valor puede ser de cualquier tipo. Ejemplo: {"departamento":8,"nombredepto":"Ventas","director": "juan rodriguez","empleados":[{"nombre":"Pedro","apellido":"Fernandez"},{"nombre":"Jacinto","apellido":"Benavente"} ]}

EJEMPLO

{

"glossary": {

"title": "example glossary",

"GlossDiv": {

"title": "S",

"GlossList": {

"GlossEntry": {

"ID": "SGML",

"SortAs": "SGML",

"GlossTerm": "Standard Generalized Markup Language",

"Acronym": "SGML",

"Abbrev": "ISO 8879:1986",

"GlossDef": {

"para": "A meta-markup language, used to create markup languages such as DocBook.",

"GlossSeeAlso": ["GML", "XML"]

},

"GlossSee": "markup"

}

}

}

}

}

EJEMPLO 2

{"widget": {

"debug": "on",

"window": {

"title": "Sample Konfabulator Widget",  
 "name": "main\_window",  
 "width": 500,  
 "height": 500

},  
 "image": {

"src": "Images/Sun.png",

"name": "sun1",  
 "hOffset": 250,  
 "vOffset": 250,  
 "alignment": "center"

},  
 "text": {

"data": "Click Here",

"size": 36,

"style": "bold",  
 "name": "text1",  
 "hOffset": 250,  
 "vOffset": 100,  
 "alignment": "center",

"onMouseUp": "sun1.opacity = (sun1.opacity / 100) \* 90;"

}

}}

**XML**

XML, siglas en inglés de eXtensible Markup Language, traducido como "Lenguaje de Marcado Extensible" o "Lenguaje de Marcas Extensible", es un meta-lenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

XML no ha nacido únicamente para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande, con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Ventajas del XML

* Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
* El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.
* Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.
* Transformamos datos en información, pues se les añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.

EJEMPLO book.xml

<?xml version="1.0"?>

<Catalog>

<Book id="bk101">

<Author>Garghentini, Davide</Author>

<Title>XML Developer's Guide</Title>

<Genre>Computer</Genre>

<Price>44.95</Price>

<PublishDate>2000-10-01</PublishDate>

<Description>An in-depth look at creating applications

with XML.</Description>

</Book>

<Book id="bk102">

<Author>Garcia, Debra</Author>

<Title>Midnight Rain</Title>

<Genre>Fantasy</Genre>

<Price>5.95</Price>

<PublishDate>2000-12-16</PublishDate>

<Description>A former architect battles corporate zombies,

an evil sorceress, and her own childhood to become queen

of the world.</Description>

</Book>

</Catalog>

Breakfast.xml

breakfast\_menu>   
   <food>   
       <name>Belgian Waffles</name>   
       <price>$5.95</price>   
       <description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>   
       <calories>650</calories> </food>   
   <food>   
       <name>Strawberry Belgian Waffles</name>   
       <price>$7.95</price>   
       <description>Light Belgian waffles covered with strawberries and whipped cream</description>   
       <calories>900</calories> </food>   
   <food>   
       <name>Berry-Berry Belgian Waffles</name>   
       <price>$8.95</price>   
       <description>Light Belgian waffles covered with an assortment of fresh berries and whipped cream</description>   
       <calories>900</calories> </food>   
   <food>   
       <name>French Toast</name>   
       <price>$4.50</price>   
       <description>Thick slices made from our homemade sourdough bread</description>   
       <calories>600</calories> </food>   
   <food>   
       <name>Homestyle Breakfast</name>   
       <price>$6.95</price>   
       <description>Two eggs, bacon or sausage, toast, and our ever-popular hash browns</description>   
       <calories>950</calories> </food>   
</breakfast\_menu>

TABLA COMPARATIVA DE SERVICIOS WEB Y ARQUITECTURA WEB

|  |  |
| --- | --- |
| Servicios Web | Arquitectura Web |
| Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.  Esta tecnología permite que aplicaciones escritas en diferentes lenguajes se integren y se ejecuten en plataformas diferentes. | La arquitectura Web es un conglomerado de acciones tendentes al desarrollo de páginas web y su optimización en torno al posicionamiento. |
| Utilizan estándares de internet. La única forma para que los servicios Web sean utilizados por la cantidad de sistemas heterogéneos existentes en Internet es el empleo del protocolo de transferencia de datos HTTP utilizado por todos los navegadores Web y XML. | El principal objetivo de la arquitectura Web es resolver las necesidades específicas del negocio:   * Venta de productos. * Servicios online. * Satisfacción de las necesidades de los potenciales clientes. |
| Basados en tecnologías de paso de mensajes. La interacción entre el cliente y el proveedor del servicio es empaquetada en unidades autodesciptivas denominadas mensajes. Dicha interacción se describe en función de los mensajes intercambiados. | Disciplina y arte encargada del estudio, análisis, organización, disposición y estructuración de la información en espacios de información, en este caso específicamente, Páginas Web. |
| Combinan lo mejor de la tecnología de componentes y de la tecnología Web. Los servicios Web presentan una funcionalidad de caja negra que puede ser reutilizada sin preocuparse de cómo es implementada y ello proporciona interfaces bien definidas. | La arquitectura web es el concepto que rodea la creación de una página web. Siguiendo con el símil del edificio o una casa, el arquitecto o diseñador |
| La interacción se basa en el envío de solicitudes y respuestas entre un cliente y un servidor, que incluyen datos. | Disciplina encargada de estructurar, organizar y etiquetar el contenido como también establecer los puntos de acceso, sistemas de búsqueda y recuperación de información de cualquier aplicación soportada en la Web, con el fin de que el usuario cumpla con los objetivos y su experiencia sea óptima. |
| Las rutinas de los Servicios Web se actualizan de forma transparente para el programador y para el encargado de mantenimiento de la aplicación. | El arquitecto web no solo desarrolla sus páginas, si no que establece los principios para que otros desarrollen de una forma correctas. |
| Servicio Web se puede implementar un programa que tenga funciones imposibles de contemplar bajo el uso de rutinas de librerías, como por ejemplo, incorporar un buscador de páginas Web. | Permite que un grupo de personas, equipo u organización sigan unas pautas para poder realizar productos de la forma más óptima mediante el uso de buenas prácticas en la web. |
| Podemos entender un servicio web como un tráfico de mensajes entre dos máquinas. | La arquitectura web es una arquitectura cliente-servidor formada por los siguientes elementos:  Un navegador: Hace de cliente y realiza peticiones solicitando recursos a los servidores web.  Un servidor web: Recibe peticiones de clientes y responde a esas peticiones enviando un recurso o notificando un error.  El protocolo basadoen TCP/IP que se utiliza para que el navegador realice las peticiones al servidor web y este responda |
| Los servicios web funcionan con los siguientes componentes: **SOAP** (Simple Object Access Protocol), **WSDL** (Web Services Description Language), **UDDI** (Universal Description, Discovery and Integration), | La arquitectura de un Sitio Web tiene tres componentes principales:   * Un servidor Web * Una conexión de red * Uno o más clientes |
| Permiten que compañías ubicadas en diferentes lugares geográficos combinen fácilmente servicios y software para proveer servicios integrados. | Puede ser descrita como el arte de planear, diseñar y construir espacios habitables, y engloba, por tanto, no sólo la capacidad de diseñar los espacios, sino también la ciencia de construir los volúmenes necesarios. |
| Aportan interoperabilidad entre las aplicaciones de software sin tener en cuenta sus propiedades o las plataformas sobre las que se instalen. | Tiene tres conceptos fundamentales:   * Planear * Diseñar * Construir |
| Son fáciles de entender (su contenido y funcionamiento) debido a que fomentan los estándares y protocolos basados en texto. |  |
| Pueden aprovecharse de los sistemas de seguridad firewall (se apoyan en HTTP) sin necesidad de cambiar las reglas de filtrado. |  |
| Facilitan la integración con afiliados de negocio, al poder compartir servicios internos con un alto grado de integración. |  |
| Disminuyen el tiempo de desarrollo de las aplicaciones. |  |
| No están ligados a ningún Sistema Operativo o Lenguaje de Programación. |  |
| No necesitan usar browsers (navegadores) ni el lenguaje de especificación HTML. |  |
| Permite el cambio de la lógica de presentación de manera sencilla, debido a que su arquitectura ofrece la alternativa de separar por completo la lógica de presentación y el almacenamiento de los datos. |  |

