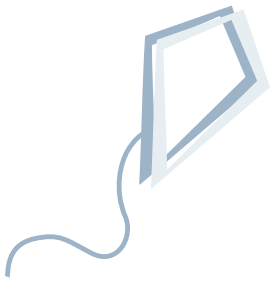
GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN

AJAX

PROGRAMACIÓN



AULA MENTOR

educacion.es

NIPO: 820-09-228-2

Autor:
Javier Robles Cascallar

Diseño gráfico de portada:
Lorena Gordoa López

MÓDULO A

UNIDADES DIDÁCTICAS:

1. 1. Introducción al concepto AJAX y a las tecnologías implicadas
2. 2. Introducción a las hojas de estilo en cascada. (CSS)
3. 3. El Modelo de Objeto de Documento (DOM)

Introducción al concepto AJAX y a las tecnologías implicadas

UNIDAD DIDÁCTICA 1

Índice de la unidad:

OBJETIVOS

En esta unidad aprenderás:

- Significado de AJAX y que tecnologías lo componen
- Como funciona una aplicación AJAX en términos generales
- Cuáles son los puntos fuertes del uso de AJAX y cuáles son sus debilidades
- Como el protocolo HTTP realiza las peticiones al servidor
- Nociones básicas de lo que es y supone el lenguaje XML

1. Introducción

Mas o menos sobre el año 2005 se empezó a popularizar entre la comunidad de usuarios de Internet el término de Web 2.0 o Web social como un grupo cada vez más numeroso de sitios web que estaban ofreciendo al usuario unas grandes posibilidades de interacción y facilitaban su participación colaborativa. El usuario dejaba de ser un mero receptor de información para pasar a interactuar y generar contenidos mediante paginas web que tratan se sustituir a las tradicionales aplicaciones de escritorio.

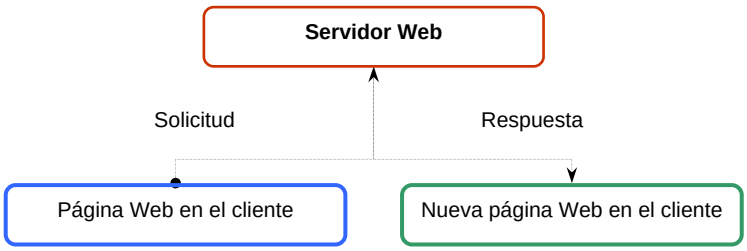
Parece claro, por lo tanto, que algo está cambiando en el mundo de los contenidos en Internet y desde que surgió el término Web 2.0 están apareciendo portales web que permiten un grado de interactividad con el usuario que hasta hace poco no habíamos visto. Gran parte de esta espectacular mejora reside en la aplicación de las tecnologías AJAX.

Ajax como concepto aparece a principios del año 2005 y es un acrónimo de: Asynchronous Javascript And XML. Realmente AJAX no es un nuevo lenguaje de programación ni un nuevo entorno de desarrollo sino un compendio de tecnologías que ya estaban en el mercado y lo que hace AJAX es reunirlas y ponerlas a trabajar juntas. Las tecnologías son:

- HTML (o mejor XHTML) junto con CSS para la parte de diseño.
- Javascript y, dentro de Javascript, el objeto DOM (Document Object Model) para poder modificar dinámicamente los documentos.
- El objeto XMLHttpRequest que es un API que puede invocarse desde JavaScript. Sirve para establecer un canal de comunicación http entre el navegador del cliente y el servidor web y permite transferir información en los dos sentidos. Este objeto se puede considerar como el motor de AJAX (AJAX Engine).
- XML como formato común para realizar intercambio de información entre el cliente y el servidor, aunque cualquier formato puede funcionar incluyendo texto plano o HTML preformateado. (No es necesario conocer XML para empezar a usar AJAX).

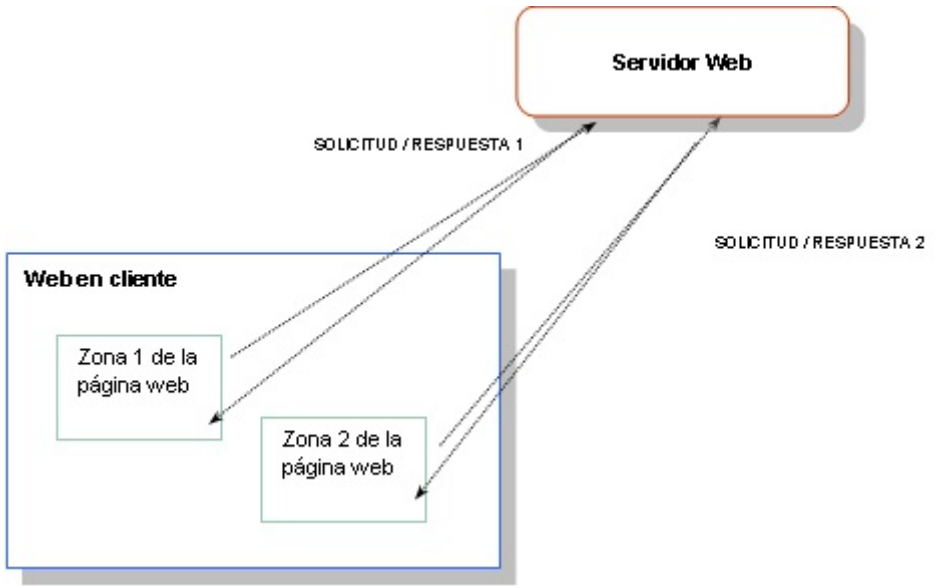
2. Funcionamiento de AJAX desde el punto de vista del usuario y del desarrollador

Primero vamos a repasar como funcionan las aplicaciones web tradicionales. Un cliente accede a un aplicación web y carga una página en su navegador, si ese cliente requiere nueva información al servidor necesariamente tiene que hacer un refresh completo de la página pulsando bien un botón de formulario o un link de la propia página.



Método tradicional de solicitud respuesta en una aplicación Web

La gran novedad es que mediante AJAX ya no es necesario refrescar por completo la página cada vez que se solicita información al servidor. Desde una misma página se pueden hacer llamadas asíncronas al servidor (mediante el objeto XMLHttpRequest) y el servidor responde actualizando con nueva información distintas zonas de la página web pero no la página web completa.



Si observamos el dibujo superior vemos que hay dos zonas de la pagina que van a acceder al servidor web de forma independiente una de otra y van a poder modificar su contenido con información que les suministre el servidor al haberse producido determinado evento sobre la pagina. Por ejemplo la Zona 1 puede actualizar su contenido cuando se seleccione, por ejemplo, un elemento de una lista de valores y la Zona 2 puede hacerlo a su vez cuando se active un determinado checkbox. Teniendo ambas zonas la posibilidad de ser completamente independientes en lo que respecta a su comunicación con el servidor web.

Si se dispone de un buen ancho de banda y la respuesta del servidor es ágil se consigue el efecto de trabajar con una aplicación de escritorio en lugar de estar interactuando con un servidor web.

AJAX combina además muy bien con los lenguajes del lado del servidor para el desarrollo web PHP,ASP,JSP,etc..

3. Ejemplos AJAX en la web

La popularidad de AJAX sin duda se debe a los ingenieros de Google (aunque paradójicamente el objeto XMLHttpRequest fue desarrollado por la gente de Microsoft para solucionar un problema que tenían con Outlook[1]). Google ha sido el primero en introducir AJAX en sus herramientas web (Google Maps, Google Suggest, etc). Muchas otras compañías relacionadas con la Web 2.0 se han apresurado a incorporar AJAX en sus sitios web (como es el caso de flickr, Yahoo mail, etc…)

Veamos un ejemplo gráfico de cómo trabaja AJAX en una aplicación web real como es Google Suggest[2], un sistema de autosugerencias que consiste en que, mientras el usuario va tecleando su entrada de búsqueda, el servidor trata de anticiparse al usuario y le muestra en tiempo real una lista con los términos mas probables que se ajustan a la búsqueda que esta tratando de teclear, el usuario puede seleccionar el termino buscado de la lista que se le presenta en lugar de seguir tecleando. Esta posibilidad de autosugerencias es una de las aplicaciones características de las tecnologías AJAX y de hecho, en este curso, se dedica un capitulo completo a su profundización.

El usuario comienza a teclear en la caja de texto para describir lo que esta buscando (vamos a suponer que es un americano que busca información sobre el aeropuerto de Almería) y a medida que va tecleando Google va sugiriendo posibles términos de búsqueda que otros usuarios han realizado (acompañado con el número de veces que ese termino de búsqueda a sido empleado).

Vemos en la figura que se muestra a continuación la salida generada por la aplicación cuando el usuario lleva tres caracteres tecleados:

[1] Outlook es el cliente de correo electrónico de Microsoft

[2] Actualmente Google Suggest esta completamente integrado en el buscador de Google

Para que esto funcione cada vez que el usuario escribe un carácter nuevo dispara un evento que lanza un objeto XMLHttpRequest al servidor web. El servidor web recibe el objeto con el contenido de texto que tiene en la caja, hasta este momento "alm", con esa información realiza una consulta en sus bases de datos de los términos de búsqueda mas usados que contengan la cadena de texto "alm" al inicio. Una vez obtenidos los diez términos de búsqueda más relevantes los devuelve al navegador del cliente y los visualiza en la zona de la página situada debajo de la caja de texto de búsqueda. Observa en la imagen que viene a continuación lo que pasa cuando el cliente sigue tecleando el siguiente carácter:

P

Vemos como, con la nueva cadena de la caja de texto “alme” enviada al servidor, este vuelve a generar su consulta en la base de datos y retorna al cliente los nuevos términos localizados (entre ellos el término “almeria airport” que estaba buscando) El usuario no necesita seguir escribiendo puesto que ya puede seleccionar su término de búsqueda con un clic de ratón.

Evidentemente para que esto funcione es necesario un muy buen canal de comunicación entre cliente y servidor web. Hay que tener en cuenta que para cada carácter tecleado se dispara un evento que genera una consulta en una base de datos remota y la percepción que debe tener el usuario es que todo se hace en modo local como si los datos estuvieran en su ordenador. AJAX solo funciona si el ancho de banda es realmente ancho y las respuestas del servidor son ágiles. Este tema de la mejora en las comunicaciones y la popularización de las conexiones a Internet mediante adsl y cable es lo que ha popularizado la Web 2.0 permitiendo que múltiples sitios web puedan incorporar AJAX en sus páginas.

4. Ventajas e inconvenientes AJAX

Como siempre que aparece una nueva tecnología que sorprende a los usuarios y gusta a los desarrolladores hay que tratar de evitar que el uso no derive en abuso. Algo parecido pasó con Flash hace unos años donde la sorpresa inicial derivó en un claro abuso y sobrecarga de efectos que hacían en ocasiones muy pesados algunos sitios Web.

Entre otras, cabe destacar las siguientes ventajas del uso de AJAX:

- AJAX supone, en general, menos carga en el servidor, aunque no es el caso del ejemplo anterior donde se produce una consulta por cada carácter tecleado, pero si en general, por que no es necesario refrescar la página entera, sino sólo la parte de la página que se va a actualizar, lo que supone menor transferencia de datos.
- Las aplicaciones Web aumentan en interactividad y hacen más atractivo de su manejo para los usuarios finales.
- Todas las tecnologías que componen AJAX son de código abierto

No todos son ventajas en el uso de AJAX, también puede generar bastantes problemas. Por citar algunos:

- AJAX no esta soportado de la misma forma por todos los navegadores.
- Demasiado código AJAX ralentiza el navegador.
- AJAX conlleva un código javascript bastante complejo. Se necesita desarrolladores más preparados.

5. Introducción a HTTP Request

En este capitulo de introducción y antes de entrar mas en detalle en las distintas tecnologías que componen AJAX es interesante dedicar unos párrafos a las peticiones y respuestas HTTP que es la base de las llamadas AJAX en modo asíncrono entre cliente y servidor.

HTTP (HyperText Transfer Protocol) o Protocolo de Transferencia de Hipertexto es un conjunto de reglas que rigen la transferencia de datos en una comunicación Web.

En su base su finalidad es la transferencia de Hipertexto (texto con componentes y enlaces a otros textos) pero en la actualidad es utilizado tanto para la transferencia de Hipertexto cómo para transferencia de Ficheros (carga y descarga de ficheros), datos (XML), trafico de red ...

Los datos de una transacción HTTP son enviados en una comunicación TCP al puerto 80 (por defecto) tipo cliente-servidor en la que el cliente (navegador web) envía una petición al servidor HTTP y el servidor responde cerrando o no la conexión.

El formato de la petición y la respuesta HTTP es el siguiente:

HTTP Request

- Línea apertura

  GET /cursoajax/ejem1.html/ 1.1

  método ruta versión

- Línea de cabecera

  Host: localhost
  User-Agent: Mozilla/5.0

- Línea en blanco
- Cuerpo del mensaje

HTTP Response

- Línea de status

  HTTP / 1.X 200 OK

  versión código razón estado

- Línea de cabecera

  Date: wed, 07 Nov 2007 18:42 :43 GMT
  Server: Apache/2.0.55 (win32) PHP/5.1.2

En la tabla siguiente se listan los códigos de estado generados por el servidor como respuesta a la petición http

Códigos de Status

100	Continua
101	Cambio de protocolo
200	OK
201	Creado
202	Aceptado
203	Información no oficial
204	Sin Contenido
205	Contenido para reset
206	Contenido parcial
300	Múltiples posibilidades
301	Mudado permanentemente
302	Encontrado
303	Vea otros
304	No modificado
305	Utilice un proxy
307	Redirección temporal
400	Solicitud incorrecta
401	No autorizado
402	Pago requerido
403	Prohibido
404	No encontrado
405	Método no permitido
406	No aceptable
407	Proxy requerido
408	Tiempo de espera agotado
409	Conflicto
410	No mapas disponible
411	Requiere longitud

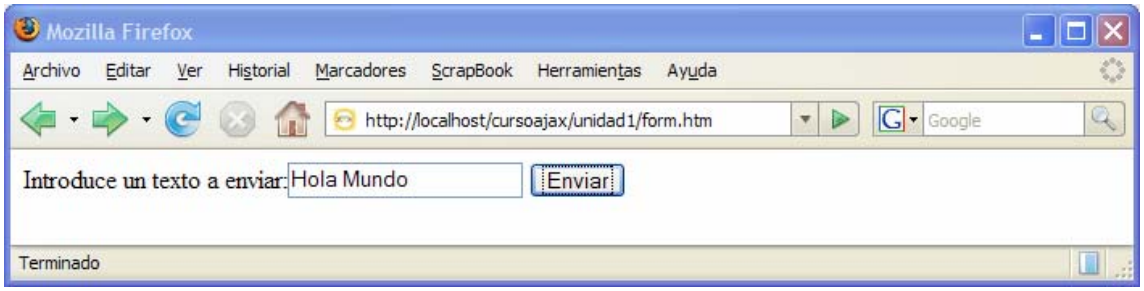
Códigos de Status

412	Falló precondición
413	Entidad de solicitud demasiado larga
414	URI de solicitud demasiado largo
415	Tipo de medio no soportado
416	Rango solicitado no disponible
417	Falló expectativa
500	Error interno
501	No implementado
502	Pasarela incorrecta
503	Servicio no disponible
504	Tiempo de espera de la pasarela agotado
505	Versión de HTTP no soportada

Es importante conocer las diferencias entre los dos métodos básicos de envío de variables a través de HTTP que son los métodos GET [3] y POST.

Cuando se envían valores de formulario vía GET dichos valores se codifican y se adjuntan a la url mientras que si envían vía POST se colocan en la zona opcional del cuerpo del mensaje de la solicitud http. Vamos a verlo con el siguiente ejemplo de un sencillo formulario con una caja de texto y un botón.

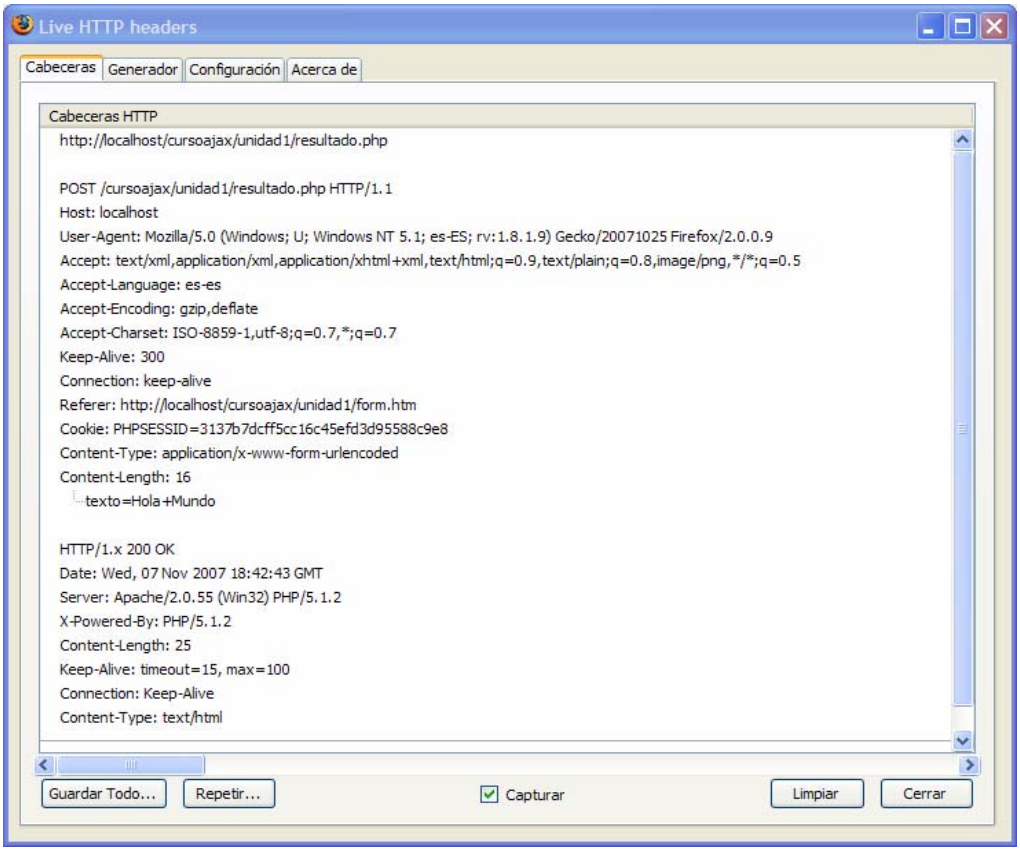
En primer lugar enviamos el formulario usando el método POST



[3] El método GET es el método por defecto cuando no se indica de modo explicito

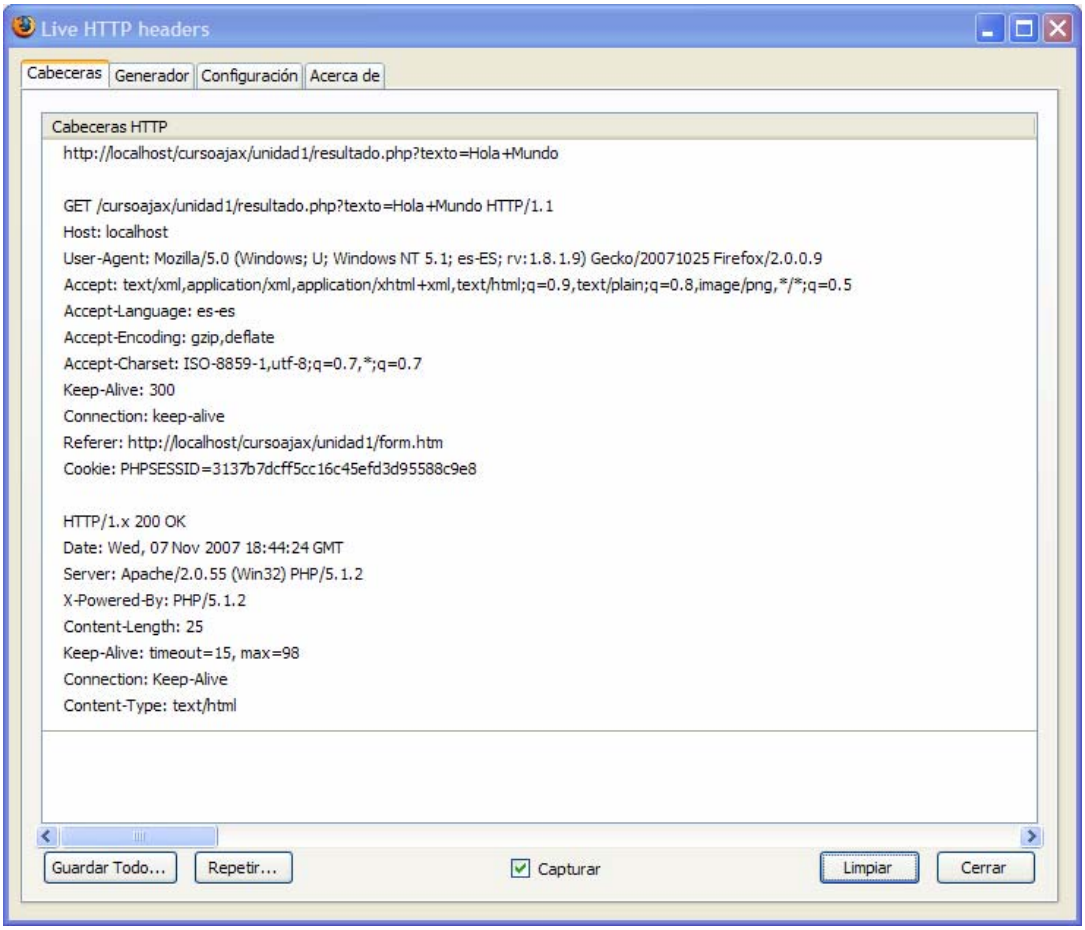
El resultado es el siguiente y, si observamos la url, vemos que no hay variables adjuntas lo cual quiere decir que se han transmitido vía POST.

Si usamos el plugin de Firefox "Live http Headers", que permite ver en detalle las cabeceras HTTP que entran en juego, nos muestra la siguiente información como resultado de la petición y respuesta del formulario anterior



Se observa analizando la solicitud como el método empleado ha sido POST y como el par "variable=contenido" `texto=Hola Mundo` se encuentra en el cuerpo del mensaje de la petición http.

Si cambiamos el método POST por GET y analizamos de nuevo las cabeceras que aparecen a continuación vemos como la url ha cambiado adjuntando la información `texto=Hola+Mundo`



Por lo tanto aunque el método GET es más cómodo de usar sobre todo para el programador por que se ve fácilmente si la información que se le esta haciendo llegar al script del servidor es correcta tiene un par de limitaciones que no tiene el método POST:

- La información que puede ir adjunta en una url está limitada. Esa limitación no la tiene el método POST.
- El método GET puede tener problemas con la respuesta de la caché. Si se envían dos peticiones GET idénticas (con la misma URL) puede darse el caso que la respuesta a esa petición la devuelva la caché intermedia del servidor WEB en lugar de ejecutarse el script por completo en el servidor. Hay dos maneras de evitar esto una es generando un aleatorio

para adjuntar a la url y que siempre sea distinta o usar el método de envio POST que evita la caché intermedia.

6. Introducción a XML

Para finalizar este primer tema de introducción al curso vamos a ver unas nociones básicas de lo que es y supone el lenguaje XML. Ya hemos visto que la X del acrónimo AJAX se refiere a este metalenguaje y por lo tanto es imprescindible que antes de entrar en las otras tecnologías que envuelven el concepto de AJAX el lector tenga claro que es XML, para que se utiliza y cuales son sus ventajas e inconvenientes.

¿Qué es XML?

XML es un acrónimo cuyo significado en inglés es EXtensible Markup Language (Lenguaje de formato ampliable) lo cual significa que utiliza etiquetas. Es decir simplemente describe información y la distribuye en un formato independiente de la plataforma porque no usa un lenguaje específico. Las etiquetas de XML no están predefinidas, lo cual significa que cada uno escribe sus propias etiquetas. La ventaja de esto es que XML no precisa ninguna explicación adicional.

XML no es un sustituto de HTML y su objetivo final es distinto. XML fue diseñado para describir, almacenar e intercambiar datos, mientras que HTML fue diseñado para presentar datos en un formato legible para las personas. HTML utiliza un conjunto predefinido de elementos (llamados etiquetas y atributos) para definir aspectos visuales de un documento, como el diseño de la página o el formato del texto, y para incluir vínculos a documentos o imágenes. En HTML uno está limitado a usar el conjunto de etiquetas de HTML. Por tanto, el tipo de información que puede mostrar es limitado. Por ejemplo, mostrar una fórmula matemática con HTML puede ser muy complicado. XML resuelve este tipo de problemas mediante la extensibilidad: puede "inventar" sus propias etiquetas y su propia estructura del documento. Se pueden añadir o eliminar elementos sin que esto afecte a la estructura global del documento.

El siguiente texto es un ejemplo de un pequeño documento en código XML:

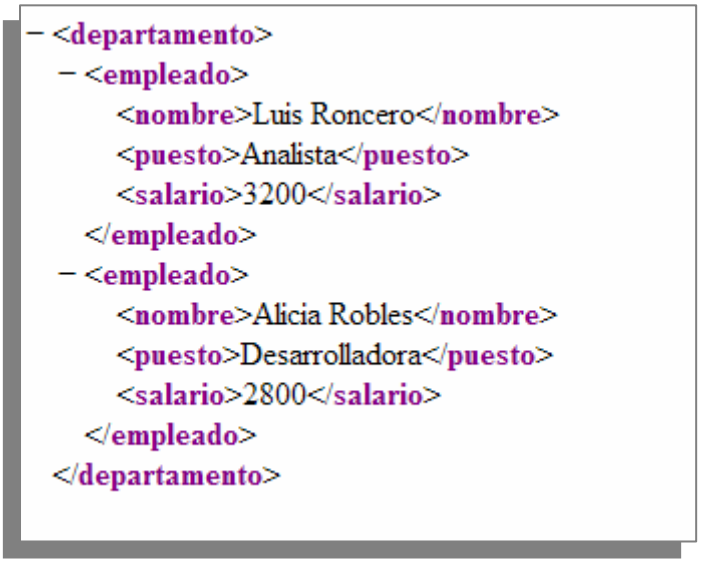
```
<?xml version="1.0" encoding="iso-8859-1"?>
<departamento>
  <empleado>
          <nombre>Luis Roncero</nombre >
  <puesto>Analista</puesto>
          <salario>3200</salario>
  </empleado >
  <empleado>
          <nombre>Alicia Robles</nombre>
          <puesto>Desarrolladora</puesto>
  <salario>2800</salario>
  </empleado>
</departamento>
```

Si nos fijamos en la primera línea del documento:

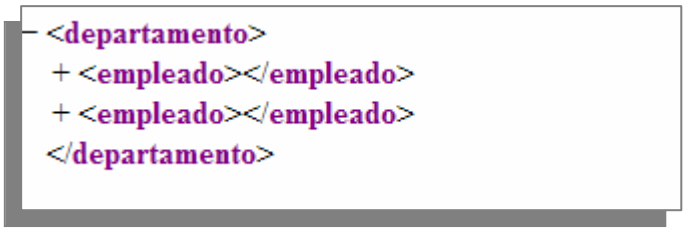
```
<?xml version="1.0" encoding="iso-8859-1"?>
```

Esta línea es la declaración XML y debe incluirse al principio de cada documento XML. Contiene la versión de XML y el juego de caracteres utilizado en el documento.

Si editamos el texto anterior y guardamos el documento con el nombre por ejemplo de `prueba.xml` podemos visualizarlo en cualquier navegador. Se pude observar que el navegador también colorea el código y muestra el documento XML como un árbol que puede contraerse (si el navegador admite XML).



Si se hace clic en el signo menos (-) al lado de cada etiqueta podrá contraer el elemento. Para expandir un elemento basta hacer clic en el signo más (+) situado junto al elemento.



¿En qué se diferencia XML de un sistema de administración de bases de datos (DBMS) típico?

En primer lugar, XML es compatible con múltiples plataformas. A diferencia de las bases de datos, que se basan en un lenguaje específico, XML incluye el significado en sus propias etiquetas. XML no precisa ninguna explicación adicional, lo cual significa que contiene tanto la estructura como la semántica de los datos, mientras que las bases de datos sólo pueden definir la estructura de los datos.

Además, XML puede representar datos mediante árboles jerárquicos, como ya hemos visto en el ejemplo anterior, el elemento <empleado> es un elemento hijo del elemento <departamento> y, como se ve en la figura anterior, se incluye como tal en el árbol XML.

Por supuesto, XML también tiene ciertos inconvenientes comparado con las bases de datos, lo cual es normal, puesto que éstas fueron diseñadas con otras finalidades. El inconveniente más obvio de XML es que carece de funciones específicas de bases de datos, como desencadenadores, acceso multiusuario, almacenamiento eficiente, índices, seguridad, transacciones, comprobaciones de integridad de datos o consultas en múltiples documentos. Esto es normal, teniendo en cuenta que un DBMS está diseñado para manipular, almacenar y recuperar datos de forma rápida y segura, mientras que XML está diseñado para intercambiar datos entre plataformas.

Como consecuencia de lo anterior, la búsqueda en documentos XML también es más lenta, debido a la falta de índices y de funciones de optimización de búsquedas, las cuales sí están disponibles en las bases de datos.

Además, XML es más detallado, ya que necesita un par de etiquetas y/o atributos para cada elemento de datos.

A pesar de que la división en bases de datos centradas en los datos y bases de datos centradas en el documento está un poco obsoleta, esta división resulta útil para entender la filosofía de XML y los conceptos más importantes de la tecnología XML.

En la siguiente sección se muestran algunas de las ventajas e inconvenientes de XML.

Ventajas de XML

- *Extensibilidad*: Cuando se trabaja con XML, con frecuencia se olvida que la X significa "ampliable". Ser ampliable significa que puede definir cualquier conjunto de etiquetas adicionales sin que se bloquee la aplicación. Por ejemplo, puede añadir hijos a este elemento:

```
<empleado>
        <nombre>Luis Roncero</nombre >
        <puesto>Analista</puesto>
        <salario>3200</salario>
</empleado>
```

Para que tenga este nuevo aspecto:

```
<empleado>
        <nombre>Luis Roncero</nombre >
        <puesto>Analista
                <cometido>Generar diagramas</cometido>
                <cometido>Analizar
requerimientos</cometido>
        </puesto>
        <salario>3200</salario>
        <fecha_alta>22/12/2007</fecha_alta>
 </empleado>
```

- *Lenguaje único independientemente de la plataforma*. .La portabilidad de XML es consecuencia de que es el propio desarrollador el que define las etiquetas y los atributos. No se necesitan bibliotecas ni servidores de aplicaciones especiales para leer un documento XML. Los documentos XML son archivos de texto normal, por lo que no requieren un software propietario para interpretarlos. Cualquier editor permite abrir y manipular código XML. Por tanto, XML es la elección correcta cuando se necesita intercambiar información a través de varias plataformas (incompatibles) de hardware o de software, o de varias aplicaciones. La portabilidad de XML también resulta útil en aplicaciones de comercio electrónico entre empresas (B2B), donde las empresas necesitan intercambiar una gran cantidad de información financiera de forma independiente de la plataforma.
- *El contenido es independiente de la presentación*. Con XML, se puede reducir el riesgo de incluir contenido redundante. Sus clientes se concentrarán en usar HTML y CSS para definir el diseño y la presentación, lo cual no se verá afectado por los cambios en la información subyacente, que se almacena por separado en un archivo XML.

Contextos más adecuados para el uso de XML.

- En aplicaciones que manejan gran cantidad de datos y, a la vez, necesitan ser flexibles y ampliables. Por ejemplo, sitios Web, listas de ofertas de empleo o aplicaciones financieras.
- En aplicaciones donde la presencia de contenidos redundantes sea un peligro, como en sistemas de administración de contenido, bibliotecas de documentos o sistemas de seguimiento de sitios Web.
- En aplicaciones donde es necesario intercambiar gran cantidad de datos a través de distintas plataformas, como las aplicaciones B2B, clientes de correo electrónico compatibles con servidores de noticias o dispositivos móviles.
- En aquellas situaciones donde la información debe estar disponible para un gran número de clientes. Por ejemplo, titulares de noticias, comunicados de prensa de empresas, avisos y anuncios importantes, marcadores, listas de reproducción, calendarios de eventos o listas de correo.

Contextos menos adecuados para el uso de XML

- No es aconsejable usar XML en sitios Web personales pequeños, o en sitios Web de presentación de empresas, ya que estos sitios contienen poca información o tienen pocas páginas. En este caso, es más recomendable utilizar HTML estático o diseños CSS/Flash..
- XML no debería emplearse en sitios Web con enormes cantidades de datos, en los cuales la velocidad de recuperación de datos y la seguridad resultan cruciales. En este caso, las bases de datos tradicionales proporcionan una solución mucho más eficiente. Algunos ejemplos de estas aplicaciones son los sitios Web corporativos, almacenes de datos, etc.
- XML no resulta aconsejable como sustituto de HTML ni de bases de datos. Si un sitio Web ya se basa en una base de datos, no hay ningún motivo para cambiarse a XML. No obstante, puede añadirse un agregador RSS para ofrecer noticias o anuncios de su empresa y permitir a los clientes que importen esta información directamente en sus sitios Web.

Sintaxis XML

Como esta introducción no pretende ser una descripción completa de la tecnología XML simplemente vamos a finalizarla con la enumeración de las reglas de sintaxis más importantes para la creación de los documentos XML:

- Todos los documentos XML deben tener un elemento raíz.
- Todos los elementos XML deben tener una etiqueta de cierre.
- Las etiquetas distinguen entre mayúsculas y minúsculas.

- Todos los elementos XML deben estar anidados correctamente.
- Los atributos deben estar incluidos en la etiqueta de apertura y deben ser escritos entre comillas.

Por último insistir en que todos los documentos XML deben empezar con la declaración XML. Las aplicaciones que llaman al documento XML utilizan la declaración XML con el fin de leer e interpretar correctamente la información. La declaración XML no es un elemento y no se trata como parte de un documento XML.

para recordar

- **AJAX** agrupa un compendio de tecnologías que trabajan conjuntamente. Las tecnologías son:
  - HTML (o mejor XHTML) junto con CSS para la parte de diseño
  - Javascript para poder modificar dinámicamente los documentos
  - El objeto XMLHttpRequest para establecer un canal de comunicación asincrona http entre el navegador del cliente y el servidor web y permite transferir información en los dos sentidos
  - XML como formato común para realizar intercambio de información entre el cliente y el servidor

- Los dos métodos básicos de envío de variables a través de **HTTP** que son los métodos **GET** y **POST**. El metodo GET es mas comodo de usar por el programador pero tiene algunas limitaciones no tiene el metodo POST

- **XML** describe información y la distribuye en un formato independiente de la plataforma porque no usa un lenguaje específico. XML esta diseñado para describir, almacenar e intercambiar datos.