

Universidad de las Ciencias Informáticas
Facultad # 10



Título: Tecnología Ajax para el desarrollo Web.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Yenivet Paula Ruiz

Yanaisy González Galbán

Tutor: Ing. Maikel Manuel Fernández Fernández

Julio 2007

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yanaisy González Galbán

Yenivet Paula Ruíz

Ing. Maikel Manuel Fernández Fernández

DATOS DE CONTACTO

Tutor: Ing. Maikel Manuel Fernández.Fernández (maikelm@uci.cu)

Profesor graduado de Ingeniero en Informática en el Instituto Superior Politécnico José Antonio Echeverría. Ha impartido clases de las diferentes Matemáticas. Es profesor de la facultad 10 y actualmente se desempeña como jefe de proyecto dentro del proyecto "Portales PHP". Se especializa en las líneas de investigación de dicho proyecto, Desarrollo Web y Sistemas de Gestión de Contenidos.

Pensamiento

“Después de escalar una montaña muy alta, descubrimos que hay muchas otras montañas por escalar”

Nelson Mandela

Agradecimientos

AGRADECIMIENTOS

A mi mamá por siempre guiarme para seguir adelante....

A mi papito por ser el mejor papá del mundo y siempre estar presente cuando lo necesito.....

A mis abuelos por siempre esperar lo mejor de mí.....

A mi hermana por brindarme todo su cariño....

A Alejandro por apoyarme en todo momento y brindarme todo su amor.....

A toda mi familia por ser la mejor.....

A todos mis vecinos por estar siempre presentes.....

A mi tutor por guiarme en todo momento.....

A todos mis profesores y compañeros de grupo por ayudarme a transitar este gran camino.....

A mis amigas por estar siempre ahí y ser como son.....

A todas las personas que han hecho posible que logre este gran sueño.....

A mi amigo Cesar por ayudarme en todo momento.....

Yanaisy González Galbán

A mi mamá por ser la mejor mamá de este mundo y siempre guiarme por el buen camino con su gran paciencia y amor.....

A mi papá por ser el mejor papá y siempre confiar en mi y por estar presente en todos los momentos de mi vida....

A mi prima Isabel por ayudarme con todo su apoyo y cariño y esperar de mi lo mejor....

A mi hermano Alexeis y su esposa Mara por apoyarme y brindarme su cariño.....

A mis abuelos por quererme tanto....

A todas las personas que han aportado un granito de arena para hacer este gran sueño una realidad.....

A mis profesores por su dedicación y colaborado con mi enseñanza.....

A mis compañeros de grupo por apoyarme durante cinco años.....

A mi compañera de tesis Yanaisy por ser mi mejor Amiga.....

Yenivet Paula Ruíz

Dedicatoria

DEDICATORIA

A todas las personas que han hecho posible que logre este gran sueño con su amor, cariño y dedicación.....

Yanaisy González Galbán

A mi mamá y mi papá.....

Yenivet Paula Ruíz

RESUMEN

El siguiente trabajo trata sobre la utilización de una técnica para el Desarrollo de Aplicaciones Web llamada Ajax, la cual será el hilo conductor de la investigación debido a que con la misma es posible crear aplicaciones interactivas que se ejecuten en el cliente, es decir una manera de crear una aplicación que responda a las acciones del usuario de forma rápida, eficaz y con mayor eficiencia y sobre todo muy cercanas a las aplicaciones de escritorio. Por esta y otras ventajas es que se hace un estudio profundo a la hora de realizar el desarrollo investigativo de este trabajo con el cual se llegará a conclusiones con respecto a la utilidad de esta técnica, además se creará un manual para que los usuario y desarrolladores interesados en esta tecnología puedan basarse en este trabajo investigativo como fuente de bibliografía para ampliar sus conocimientos y así poder elaborar aplicaciones mucho más interactivas y de la misma manera demostrar la importancia de incorporarla al Proyecto Productivo "Portales".

ÍNDICE

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 5

 Introducción..... 5

 1.1 HTML 7

 1.2 CSS..... 9

 1.3 JavaScript 10

 1.4 Lenguajes Dinámicos 13

 1.5 DHTML..... 16

 1.6 XML..... 18

 1.7 JSON..... 19

 1.8 DOM..... 21

 1.9 Ajax 21

 1.10 Casos de estudio y aplicaciones 26

 1.10.1 CMS..... 28

 1.10.2 Frameworks 28

 Conclusiones..... 29

CAPÍTULO 2: INDICACIONES PARA LA UTILIZACIÓN DE AJAX..... 30

 Introducción..... 30

 2.1 Principales usos de Ajax 30

 2.2 Cuando no usar Ajax..... 34

 2.3 Como utilizar Ajax 35

 2.3.1 Objeto XMLHttpRequest 35

 2.3.2 Cómo crear el objeto XMLHttpRequest..... 45

 2.3.3 Cómo realizar una petición con Ajax 46

 2.3.4 Cómo recibir la petición Ajax 49

 2.3.5 Respuesta Ajax..... 51

Índice

2.3.6 Tratamiento de la respuesta Ajax	53
2.3.7 Cómo mostrar los datos al usuario	55
2.3.8 Implementaciones de Ajax.....	57
Conclusiones.....	58
CONCLUSIONES	59
RECOMENDACIONES.....	60
REFERENCIAS BIBLIOGRÁFICAS	61
BIBLIOGRAFÍA.....	62
ANEXOS.....	68

INTRODUCCIÓN

Actualmente, en el mundo de las TIC (Tecnologías de la Información y las Comunicaciones) es sumamente importante la utilización de Aplicaciones Web porque en la medida que los límites de estas se van superando, se va haciendo más potente la posibilidad de mover todas las aplicaciones a la web, ya que la disponibilidad es mejor, el modelo de acceso desde cualquier parte del mundo es genial, y el mantenimiento y la configuración realmente fácil. Una aplicación web no puede ser tomada como una simple página web, sino como un programa complejo de naturaleza concurrente, multiusuario y capaz de ser utilizado en múltiples entornos. Las viejas páginas web con HTML y JavaScript han dado paso a aplicaciones web con conexiones a bases de datos, a servidores de aplicaciones, a sistemas de comunicación instantánea, tanto síncronos como asíncronos, entre otras muchas facilidades que brindan estas aplicaciones.

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, como parte de la aplicación y requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal.

Se puede hacer un sitio completo sin necesidad de recargar toda la página y sólo cargando las partes requeridas. La tecnología usada con este fin es el XMLHttpRequest, bautizada recientemente con el nombre de Ajax, que en realidad no es una tecnología, sino la combinación de muchas tecnologías ya existentes, las cuales florecen por su propio mérito, uniéndose en poderosas nuevas formas. El empleo de esta tecnología se generaliza cada día más, ya que ha aparecido con gran fuerza y a cada instante son más los que la adoptan y desarrollan sus productos basados en esta forma de trabajar. Ajax (Asynchronous JavaScript and XML) puede marcar una época en el desarrollo de la web, ya que al utilizar esta tecnología como filosofía de

Introducción

trabajo se puede cada vez más realizar aplicaciones online , ofrecer servicios globales interactivos basados en la red y crear aplicaciones interactivas mucho más eficientes.

En la Universidad de las Ciencias Informáticas (UCI) existe un proyecto productivo llamado Portales el cual tiene la necesidad de introducir esta tecnología en sus producciones ya que esta técnica brinda las mejores ventajas en cuanto al dinamismo de los sitios web que son elaborados con ella y para esto se precisa de un conocimiento basto de la misma. Por ello surge el siguiente problema: ¿Cómo introducir el uso de la tecnología Ajax en el desarrollo web del proyecto portales?

De acuerdo con lo que anteriormente se expone surge la siguiente hipótesis: Si se confecciona una guía para la utilización de la tecnología Ajax como fuente de consulta para los desarrolladores del proyecto productivo Portales entonces se podrá introducir esta técnica en el proyecto. Por lo que la variable independiente está relacionada con el estudio de la tecnología Ajax y la dependiente no es más que lograr un documento completo y eficiente para que los desarrolladores del proyecto “Portales” puedan relacionarse con esta técnica y además utilizarla.

Parte de la culpa de que las aplicaciones web sean cada vez más parecidas, en calidad, a las aplicaciones de escritorio (las que se activan desde la PC) hay que atribuírsela al desarrollo de Ajax. Este nombre engloba a toda una cantidad de tecnologías que se utilizan para el diseño y desarrollo de programas interactivos.

La principal ventaja de las aplicaciones realizadas con estas tecnologías es que se actualizan de forma dinámica, sirviéndose casi siempre del lenguaje XML, y no mediante el 'refresco' de páginas enteras que se muestran al usuario cada vez que este realiza alguna acción.

Por esta razón, las aplicaciones desarrolladas sobre la base Ajax son más rápidas que las herramientas web anteriores a la aparición de este lenguaje de programación.

Introducción

El desarrollo de Ajax ha permitido que Internet haya ido poblándose en los últimos meses con aplicaciones cada vez más prácticas y funcionales y que han sido desarrolladas por pequeñas compañías y no por gigantes de la talla de Yahoo, Google o Microsoft.

El objeto de estudio de esta investigación corresponde a las Aplicaciones Web y el campo de acción lo conforma la tecnología Ajax para el desarrollo de aplicaciones web.

El objetivo general es crear una base de conocimiento y fuente de consulta que ayude a la introducción de la tecnología Ajax para el desarrollo de aplicaciones web en el proyecto productivo “Portales” de la facultad número diez.

Los objetivos específicos para el desarrollo del presente trabajo son:

- Definir la importancia y necesidad del empleo de Ajax.
- Exponer con claridad las ventajas y desventajas de la tecnología Ajax.
- Definir las tecnologías que involucra el empleo de Ajax y la forma en que se hace.
- Exponer el impacto y tendencias del empleo de Ajax.
- Crear un manual para el uso de la tecnología.

A la hora de elaborar el presente documento se le da cumplimiento a los objetivos propuestos desarrollando las siguientes tareas:

- Estudio de documentos relacionados con la filosofía Ajax y sus componentes, para así poder incorporarla al proyecto productivo.
- Investigación sobre el funcionamiento de la filosofía Ajax.

En el diseño metodológico de la investigación se emplearon diferentes Métodos Teóricos por ejemplo el Analítico-Sintético: porque en la investigación se buscará y analizarán diferentes documentos, opiniones y teorías acerca de la tecnología Ajax para el desarrollo de las aplicaciones web. El Inductivo-Deductivo: porque permite obtener conocimientos generalizados de la tecnología Ajax, partiendo desde el análisis de lo particular a lo general. El Análisis

Introducción

Histórico-Lógico: porque permite constatar de forma teórica cómo ha evolucionado la herramienta Ajax.

El contenido de esta investigación está distribuido en dos capítulos donde el primero es todo lo referente a la fundamentación teórica, es decir se explica todo lo relacionado a las tecnologías con la que se relaciona Ajax hasta caer en esta que es la posee el sentido de esta investigación. El segundo capítulo trata lo relacionado con las indicaciones para la utilización de Ajax, donde se explica cuándo debe usarse Ajax, cuándo no debe usarse y cómo utilizarlo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se trata de hacer un acercamiento a los aspectos más importantes que se necesitan saber para entender mejor el tema que desarrolla esta investigación, que es el relacionado con las aplicaciones web elaboradas con la tecnología Ajax (Asynchronous JavaScript and XML).

Una aplicación web no es más que una aplicación informática, las cuales son muy populares ya que el cliente solo necesita de un navegador web para acceder e interactuar con dichas aplicaciones. Esto trae consigo la posibilidad de actualizar y mantener las aplicaciones sin necesidad de distribuir o instalar software en los miles de clientes que las utilizan.

Al mismo tiempo generan dinámicamente una serie de páginas en un formato estándar, que son soportadas por navegadores web, que son muy comunes. Se utilizan también lenguajes interpretados del lado del cliente y un ejemplo de este tipo de lenguaje es el JavaScript.

Una de las consideraciones técnicas que deberíamos tener en cuenta a la hora de analizar las aplicaciones web es la ventaja de que deben funcionar igual independientemente del sistema operativo que tenga instalado el cliente al que se le está brindando el servicio.

La Web ha ido evolucionando desde sus inicios hasta la actualidad y continuará creciendo ya que hoy se encuentra en una etapa clave de este desarrollo, marcada por servicios y características particulares que requieren de nuevas técnicas de desarrollo de aplicaciones web y en muchos casos el uso de tecnologías ya existentes. En sus inicios la web era utilizada por los navegantes para obtener información y la unidad de trabajo era la página web, de esta forma, los usuarios navegaban entre múltiples páginas a través de hipervínculos. Generalmente, los sitios consistían en sitios corporativos, sitios de noticias, sitios de información específica. Actualmente la visión de la web es como una plataforma, es decir, como el medio para la distribución de servicios y para la

Capítulo 1 Fundamentación Teórica

construcción de información. Hoy en día tienden en su mayoría al dinamismo, estas generalmente se presentan como una aplicación de tres-capas, una capa de presentación, que es la primera capa, un motor usando alguna tecnología web dinámica que es la capa de medio, y una base de datos que es la última capa, conocida como capa de datos. El navegador web manda peticiones (realizadas mediante la capa de presentación) a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos y se genera una interfaz de usuario mostrando los resultados de la petición. (Anexo 2) (1)

La accesibilidad y los requerimientos mínimos por parte del cliente no son los únicos puntos que han hecho la web tan popular, también se puede mencionar el tamaño de las aplicaciones, la posibilidad de extensión de las mismas, la implementación de servicios de cara al usuario final y la posibilidad de realizar una presentación muy agradable.

Una limitante que se puede mencionar es la imposibilidad en sus inicios de implementar algunos recursos que incluían las aplicaciones de escritorio, la necesidad de recargar cada página al realizar una acción y la ausencia de algunos controles que si se incluían en las aplicaciones escritorio. Con la evolución de los lenguajes propios de la web, como el HTML, XHTML , CSS y la incorporación y desarrollo de otros como JavaScript y XML, además de nuevas tecnologías como DHTML y Ajax, esta limitante va quedando atrás.

El futuro de la web presenta grandes desafíos. La implementación de ellos requerirá de nuevas tecnologías y técnicas de programación. Ajax es una de estas técnicas de programación utilizadas, en el mundo existen miles de técnicas para el desarrollo de Aplicaciones Web, por lo que se analizarán algunas de ellas, es decir las que están relacionadas con la tecnología Ajax porque son en las que se basa para poder llevar a cabo todas sus funcionalidades, y así de esta manera poder formar la información necesaria referente a esta técnica.

1.1 HTML

HTML es un lenguaje muy sencillo llamado HyperText Markup Lenguaje que como su nombre lo indica es un lenguaje de marcas pensado para la creación de hipertextos los cuales no son mas que un texto al cual se le puedan incluir elementos multimedia es decir audio, video y gráficos, también acepta la presencia de hiperenlaces que permiten relacionar otras fuentes de información. Es un lenguaje de marcas ya que en él las instrucciones son trozos de texto resaltados convenientemente, que definirán la estructura lógica del documento. Es decir un documento HTML constará de texto que será el contenido, la información del documento y de instrucciones HTML que resaltarán este contenido y le darán un formato fácil de leer y con la posibilidad de relacionar documentos y fuentes de información mediante hipervínculos.

HTML es, sin duda, el primer paso para todo aquel que se interese por el diseño de páginas web, de ahí la importancia de este lenguaje que es absolutamente imprescindible. Una página web básicamente es una colección de contenedores de información situados en un documento, siguiendo las instrucciones marcadas mediante un lenguaje de etiquetas denominado HTML. Los documentos así escritos son interpretados por un programa llamado explorador o navegador que se encarga de presentar el documento al usuario final. (2)

El principio esencial del lenguaje HTML es el uso de las etiquetas (tags). Indistintamente existen etiquetas de apertura y de cierre, las mismas afectan al contenido que quede entre ellas. Estas se clasifican principalmente en cuatro grandes grupos el primero de Contenido, el segundo de Formularios, el tercero de Estructura y el cuarto de Formato. Las etiquetas de Contenido representan información semántica del contenido que afectan, por ejemplo, una información puede estar dividida en capítulos, párrafos y secciones. Las etiquetas de Formularios, definen elementos de la página que posibilitan al usuario enviar datos hacia el servidor. Las etiquetas de Estructura permiten especificar la distribución espacial de la información en la página, por

ejemplo, mostrarla a manera de columnas, en bloques, que abarque toda el área visible. Las etiquetas de Formato, permiten darle color, tamaño y forma, al contenido que encierren.

Es importante señalar que un documento HTML siempre debe estar entre dos etiquetas <HTML> y </HTML> las cuales permiten un mejor entendimiento y elaboración del documento. También que los documentos de este tipo están divididos en dos zonas esenciales la primera es el encabezamiento comprendida entre las etiquetas <HEAD> y </HEAD> dentro de la cual hay información del documento que no se ve en la pantalla principal del BROWSER, generalmente la información encontrada en esta zona es el título del documento. En la segunda está el cuerpo comprendido entre las etiquetas <BODY> y </BODY>, dentro de esta zona se encuentra todo lo que se quiere que aparezca en la pantalla principal (texto, imágenes). De forma general la estructura de un documento HTML queda de esta forma:

```
<HTML>
  <HEAD>
    <TITLE>.....titulo de la pagina.... </TITLE>
  </HEAD>
  <BODY>
    .....etiquetas que visualizan la página.....
  </BODY>
</HTML>
```

Existen incontables etiquetas que hacen posible la elaboración de un documento HTML algunas de ellas son por ejemplo: <P> y </P> las cuales son utilizadas para separar el texto en diferentes párrafos.<H1> y </H1>, <H2> y </H2> así sucesivamente hasta el número 6 siendo el número el indicativo del tamaño el mayor tamaño es el correspondiente al número 1, estas etiquetas son para las cabeceras de los textos; y para poner algo en negrita; <I> y </I> para poner algo en cursiva. Cuando se realiza algún enlace se utiliza la siguiente estructura Texto de Enlace , donde URL es el destino del enlace, con un color especial

y generalmente subrayado; existen diferentes tipos de enlaces ya sean dentro de la misma página, a otra página del mismo sitio, con una página fuera de nuestro sistema o con una dirección de correo electrónico. Para insertar alguna imagen en el sitio también se utiliza una etiqueta que es `` donde con el atributo SRC (source) se indica que se quiere cargar una imagen llamada imagen.gif (o el nombre que tenga), además es posible dentro de la etiqueta añadir otros comandos como por ejemplo `` donde con el atributo se introduce una descripción ya sea una palabra o una frase breve indicativa de la imagen. Existe además la etiqueta `<TABLE>` que puede parecer un modo sencillo de disponer el texto en columnas o de añadir un titular a una ilustración pero en realidad es una poderosa herramienta de estructura porque con ella se puede dejar de mostrar el borde de una tabla en absoluto, también para ubicar una imagen con precisión en cualquier lugar de la página, dentro de ella también hay otras etiquetas que la conforman por ejemplo `<TR>` y `</TR>` para las columnas y `<TD>` y `</TD>` para las filas.

Los documentos creados con este lenguaje son estáticos es decir la información de la página que se presenta al usuario siempre es de la misma forma, siendo necesario que el diseñador modifique la página si se desea que tenga una apariencia diferente. Además de ser estáticas, las páginas web así escritas permiten muy poca interacción con los usuarios, básicamente los enlaces y los formularios. El panorama mejora mediante las hojas de estilo en cascada (CSS).

1.2 CSS

Las Hojas de Estilo en Cascada o CSS(Cascading Style Sheets) surgen por la limitación del lenguaje HTML a la hora de darle forma a un documento, permitiendo dar solución a estos problemas ya que son mecanismos simples que posibilitan aplicar formato a los documentos escritos en HTML y en lenguajes estructurados como XML, dando la posibilidad de separar el contenido de la presentación, es decir, la información estará incluida en la página HTML pero sin ser este el archivo que defina como será visualizada la información, las indicaciones referentes a

cómo quedará comprendido visualmente el documento estarán especificadas en los archivos de la CSS. Describe cómo se verá un documento en pantalla, siendo esta forma de descripción de estilos la vía que ofrece a desarrolladores el control total de estilos y formato de sus documentos así como de múltiples páginas web al mismo tiempo. Un cambio en el estilo para un elemento en la CSS implica que todas las páginas vinculadas a esa CSS en la que aparezca ese elemento serán afectadas. Funciona a través de reglas, o sea declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por varias de esas reglas aplicadas a un documento HTML o XML. Estas reglas constan de dos partes: un selector y la declaración. La declaración esta compuesta por una propiedad y el valor que se le asigne. El selector es como el enlace entre el documento y el estilo, especificando los elementos que serán afectados por esa declaración. La declaración es la parte de la regla que establece cual será el efecto. Una de las formas que se utilizan para dar estilo a un documento es la hoja de estilo externa, que se vincula al documento a través del elemento link, que debe estar situado en la sección <HEAD >, también utilizando el elemento <STYLE> dentro del documento que se le quiere dar estilo, y que generalmente se encuentra situado en la sección <HEAD>, siendo así los estilos reconocidos antes de que la página se cargue completamente. Otra forma de dar estilo a un documento es utilizando estilos sobre los elementos que lo permiten a través del atributo <STYLE> dentro del <BODY>. Es importante resaltar el cuidado que se debe tener a la hora de usar esta tecnología ya que muchos usuarios puede que no vean los formatos que son aplicados a las páginas con CSS.

1.3 JavaScript

JavaScript es un lenguaje que permite dar dinamismo a las páginas web. Los scripts escritos con JavaScript amplían las páginas web dotándolas de una interactividad con el usuario mucho mayor que los simples hipervínculos y formularios, además permiten a las páginas web cambiar de forma dinámica, es decir se adaptan al entorno de la máquina cliente mientras son recibidas

por el navegante y permite crear efectos especiales en las páginas. El encargado de interpretar las instrucciones JavaScript es el navegador cliente siendo el que las ejecuta para realizar las interactividades.

Después de HTML, JavaScript es el siguiente paso que puede dar un programador web que decida mejorar sus páginas y la resistencia de sus proyectos. Es un lenguaje bastante sencillo y fácil de aprender, pensado para hacer las cosas con rapidez.

Este lenguaje de programación cuenta con acciones típicas de él, siendo los efectos especiales aplicados a las páginas web para crear elementos de la página que tengan movimiento, cambien de color o para crear contenidos dinámicos, además posibilita ejecutar instrucciones que dan respuesta a las acciones del usuario, es decir páginas interactivas con programas como agendas o tablas de cálculo. Permite la programación de pequeños script al igual que grandes programas con funciones y estructuras de datos complejas. (3)

La programación de JavaScript se realiza dentro del documento HTML, donde se mezclan los dos lenguajes de programación con delimitadores que separan las etiquetas HTML de las instrucciones JavaScript. Estos delimitadores son las etiquetas `<SCRIPT>` y `</SCRIPT>` donde todo el código JavaScript se coloca dentro de estas dos etiquetas. Una página puede contener varios `SCRIPT`, la colocación de estos es indiferente, en un principio da igual pero en determinados casos la colocación es importante. Existen dos maneras de ejecutar estos `SCRIPTS` en la página, la primera es la ejecución directa de scripts siendo el más básico. En este se incluyen las instrucciones dentro de la etiqueta `<SCRIPT>` y al navegador leer la página se ejecutan directamente los script, es decir cuando encuentra un script va interpretando las líneas de código y ejecutando una después de la otra. La segunda manera de ejecutar script es la ejecución como respuesta a la acción de un usuario.

Los usuarios realizan acciones conocidas como eventos, algunas de estas acciones realizadas sobre la página en este caso es atrapada por JavaScript y realiza acciones como respuesta, así se realizan programas interactivos, controlando los movimientos del usuario y respondiendo a ellos. Las acciones que se realizan como respuesta a un evento se indican dentro del código

HTML o en atributos HTML que se colocan dentro de la etiqueta que va a responder a las acciones del usuario. (4)

Con JavaScript se crean aplicaciones específicamente orientadas a su funcionamiento en la red. Este lenguaje permite crear páginas HTML dinámicas que procesen la entrada del usuario y además que sean capaces de gestionar datos usando archivos y bases de datos relacionales.

Las aplicaciones cliente y servidor en JavaScript comparten el núcleo del lenguaje el cual se corresponde con ECMA-262, el lenguaje de script estándar de la Oficina de Estándares de la Unión Europea, con algunos adicionales. Es de utilidad resaltar que aunque JavaScript de cliente y de servidor comparten el conjunto de base de funciones y características, en algunos casos se utiliza de forma diferente.

JavaScript para clientes encierra el núcleo del lenguaje y algunos elementos adicionales. El código se coloca directamente en páginas HTML y es interpretado por el cliente web en tiempo de ejecución. Cuando el cliente web solicita páginas donde existen sentencias de JavaScript integradas en el documento HTML, el servidor envía por la red al cliente el documento completo permitiéndole leer secuencialmente desde el principio hasta el final, representando visualmente los códigos HTML y ejecutando las sentencias JavaScript que aparecen durante el avance del proceso de lectura e interpretación. Estas sentencias pueden dar respuestas a eventos de usuarios es decir a la pulsación de un botón del ratón, la entrada de datos de un formulario y la navegación por una página, además permite validar información sin necesidad de realizar transmisión de datos por la red.

El servidor JavaScript también esta integrado en páginas HTML. El código JavaScript para el servidor incluye el núcleo de lenguaje, objetos y funciones primordiales para el funcionamiento correcto en el marco de un servidor. Las sentencias de JavaScript en el servidor permite la conexión a bases de datos relacionales, compartir información entre usuarios de una aplicación, acceder a los ficheros del servidor así como comunicarse con otras aplicaciones a través de LIVECONNECT y JAVA. Las aplicaciones JavaScript del servidor se compilan generando

archivos binarios, además cuenta con servicios especiales en el servidor, el servicio de Gestión de Sesiones y el Servicio de Bases de Datos LiveWire.

Posee una posición ventajosa al ser utilizado para la creación de grandes soluciones para la web. Es sencillo de aprender, es ideal para agregar funciones rápidas a una página web, además de rápido y potente. Para ejecutar el código solo hay que crearlo y cargarlo. Muchas páginas web incorporan elementos que lo usan y por lo general los exploradores web trabajan con él siendo uno de los lenguajes más utilizados en la web. Posibilita la reducción de carga del servidor, se hace cargo de la mayoría de las funcionalidades del cliente de las cuales se encargaba el servidor.

1.4 Lenguajes Dinámicos

Los lenguajes dinámicos son lenguajes de programación que en tiempo de ejecución pueden extender un programa agregando código adicional, también extender sus objetos y sus definiciones. Además proveen herramientas directas para dichos comportamientos.

Dentro de los diferentes lenguajes dinámicos que son muy conocidos está el ASP (Active Server Pages, páginas de servidor activas) es una tecnología para hacer scripts del lado del servidor. Los scripts ASP se ejecutan antes de ser enviados, a diferencia de las páginas HTML simples donde el cliente solicita una página de un servidor, el cual se la envía y es mostrada en el navegador, con ASP el servidor lo procesa a través de una DLL llamada ASP.DLL la que interpreta los mandatos ASP. Se utiliza para que la relación entre cliente servidor sea más dinámica. El código ASP primero es ejecutado en el servidor y los resultados son enviados en HTML al usuario que los pidió, además sirve para interactuar con el navegante. El código ASP se escribe dentro de los TAGS `<% %>` para insertar sus instrucciones dentro de código HTML, siendo los que empiezan y terminan el código. Las páginas ASP se guardan con la extensión `.asp`. Son ejecutados en servidores con Windows usando el IIS (Internet Information Server) o el

PWS (Personal Web Server) ya que sin un servidor que lo soporte, el ASP no funciona y para comenzar la programación se debe instalar un servidor. Existen diversidades de lenguajes que pueden ser utilizados como es JavaScripts aunque ASP se inicio con el VBScript. Es una tecnología dinámica dependiente del servidor siendo una ventaja ya que brinda seguridad a los programadores sobre su código al este encontrarse únicamente en los archivos del servidor que es ejecutado cuando un usuario solicita un archivo.asp a través de la web donde estos no tienen acceso más que a las páginas resultantes en formato HTML a través de su navegador, siendo el que realiza todo el trabajo para generar las páginas que se envían al explorador. Los archivo .asp son archivos de texto que no necesitan de ningún editor específico para ser creados, solo debe cumplir que genere código ASCII. Están conformados por texto, código HTML, código ASP o la mezcla de estos, aunque al no tener código ASP se comporta como un archivo .html. Permite el acceso a base de datos, envío de correo electrónico siendo unas de sus funciones principales.

Para correr ASP en servidores que no sean IIS, se ha desarrollado en caso de un servidor Linux el chilisoft, otra opción es InstantAsp.

Por otro lado Java que es capaz de ejecutar programas de una manera diferente desde navegadores con soporte java, reproduce sonidos directamente desde él, además puede enseñar al navegador a manejar nuevos formatos de ficheros. Elimina los inconvenientes de la interfaz CGI. Su código fuente brinda seguridad al sistema ya que no contiene ningún segmento de código malicioso. Es un lenguaje simple lo que no quiere decir que no sea un lenguaje potente, es dinámico, trabaja sus datos como objetos y con interfaces a esos objetos, soporta la encapsulación, herencia y polimorfismo características propias del paradigma de orientación a objeto. Tiene amplias capacidades de interconexión TCP/IP, además cuentan con librerías y herramientas para que los programas sean distribuidos, para acceder e interactuar con protocolos como http y ftp lo que posibilita el acceso a la información a través de la red así como a los ficheros locales. Tanto en tiempo de ejecución como en tiempo de compilación java realiza verificaciones en busca de problemas, la compilación ayuda a detectar errores en el ciclo de desarrollo, aunque al obligar a declarar explícitamente los métodos reduce las posibilidades de

error. Es un lenguaje seguro, independientemente de su portabilidad básica, implementa otros estándares de portabilidad para facilitar el desarrollo.

Además PHP (Hypertext Pre-processor) que es un lenguaje de programación interpretado y multiplataforma muy usado para la creación de aplicaciones para servidores y contenido dinámico para sitios Web. Permite a programadores de experiencia crear aplicaciones complejas sin tener que esforzarse al máximo siendo el nivel de aprendizaje suave y le permite mezclarse con aplicaciones de contenido dinámico sin tener que aprender nuevas funciones por lo fácil que resulta su uso. Es posible mediante el diseño PHP crear aplicaciones con interfaz gráfica para el usuario utilizando la extensión PHP-GTK. Es interpretado y ejecutado en el servidor web donde se encuentran los script y el cliente solo recibe el resultado de la ejecución. Cuando el usuario solicita una página web el servidor ejecuta el intérprete de PHP el cual de manera dinámica procesa los script generando el contenido el cual puede sufrir modificaciones y devuelve el resultado al servidor el cual se encarga de regresárselo al cliente. Se utiliza para generar archivos PDF, Flash, imágenes en diferentes formatos, entre otros. Permite la conexión a diferentes servidores de bases de dato como MySQL, Postgres, Oracle, ODBC, DB2, MicrosoftSQLServer, permitiendo la creación de aplicaciones web muy fuertes y resistentes. Tiene la posibilidad de ejecución en casi todos los sistemas operativos tales como UNIX (como Linux), Windows y MacOSX y puede interactuar con los más populares servidores de web ya que existe en versión CGI, módulo para Apache, e ISAPI. En los sistemas operativos en los que es soportado se puede desarrollar aplicaciones de escritorio creando aplicaciones gráficas independientes del navegador mediante la combinación de PHP y GTK. Permite leer y manipular datos desde distintas fuentes incluyendo datos que los usuarios ingresen desde formularios HTML. Debido a sus múltiples módulos o extensiones presenta la capacidad de expandirse. Es de fácil acceso por ser libre, permite la programación orientada a objeto, crear formularios para la web, contiene una biblioteca de funciones sumamente amplia, no requiere definición de tipos de variables y su código se encuentra dentro del HTML e interactúa con el mismo, permitiendo diseñar la página web en el editor de HTML añadiendo el código dinámico dentro de las etiquetas

<?php >. El éxito de algunos pequeños desarrollos que se han convertido en grandes éxitos utilizados por muchas empresas (caso de Wordpress) así lo demuestra.

1.5 DHTML

Cada día que pasa es una nueva era en el mundo de la informática y es por eso que poco a poco las viejas tecnologías se van haciendo pequeñas para lograr lo que se quiere, por ejemplo HTML se iba quedando pequeño a la hora de programar aplicaciones web ya que no era suficiente este lenguaje para crear documentos interactivos, es decir una vez que se carga la página en el navegador esta no puede ser modificada y es entonces cuando surge el DHTML o el HTML Dinámico que no es más que un conjunto de técnicas que posibilitan la creación de Sitios Web interactivos utilizando una combinación entre el lenguaje HTML estático, JavaScript y CSS (Hojas de estilo en cascada), ofreciendo a los creadores la posibilidad de modificar, cambiar la apariencia o mostrar el contenido de una página dinámicamente.

Es un término utilizado para la descripción de aquellas páginas que son el resultado de la ejecución en la parte cliente de algún tipo de lenguaje, precisamente en el momento de cargar la página. DHTML es utilizado de la siguiente forma, es ejecutado en la parte cliente un código (como por ejemplo JavaScript) antes de cargar la página, este código tiene cualquier tipo de efecto o interactividad en la misma, es decir de esta manera se logra una modificación en el aspecto visual de la misma ya sea video, sonido o control y respuesta de las acciones de los diferentes usuarios y todo esto antes que esté completamente cargada en el navegador. Por otro lado existen una serie de lenguajes (ASP, PHP, Java) que se basan en el servidor para ejecutar sus scripts y al igual que en la programación del cliente se basa en el navegador; cuando es solicitada una página por parte del cliente el servidor ejecuta los scripts y genera una página resultado la cual contiene únicamente código HTML, todo esto tiene una gran ventaja y es que el

código de las páginas con los scripts nunca llega al cliente lo que implica que los visitantes nunca puedan acceder al corazón de la aplicación.

Engloba un conjunto de técnicas con dos objetivos claros el primero es proporcionar un control absoluto al diseñador de páginas HTML y el segundo es que rompa con el carácter estático de este tipo de documentos. El DHTML está compuesto por varios componentes, uno de ellos es las Hojas de Estilo las cuales permiten un mayor control sobre el aspecto de nuestros documentos, con ellas es posible especificar muchos atributos tales como colores, márgenes, alineación de elementos, tipos y tamaños de letras en fin permiten especificar los atributos estilísticos tipográficos de una página; estas pueden crearse empleando dos tipos de sintaxis, CSS (Cascade Style Sheets) y JavaScript. Otro componente es el Posicionamiento de Contenidos ya que es posible mediante la especificación de la posición de los bloques decidir donde se mostrará cada elemento en vez de dejar esta tarea al arbitrio del navegador, es decir ahora se puede decidir la posición exacta de los elementos; también utilizando JavaScript se puede cambiar el aspecto de la página dinámicamente, se puede lograr que los elementos aparezcan o se desvanezcan, cambiar su color y su posición. El tercer componente son Las Fuentes Descargables las cuales no son más que otra forma de asumir un control más directo sobre la presentación de una página, con la utilización de este componente se garantiza que las páginas siempre sean mostradas con la fuente que se desee, ya no es necesario emplear las fuentes genéricas para conseguir que sus páginas tengan aproximadamente el mismo aspecto en todas las plataformas, para proteger los derechos de autor de los diseñadores de fuentes, estas están protegidas de forma que es imposible que el usuario las copie y las pueda usar de nuevo.

En resumen el DHTML proporciona una técnica para realizar un sitio web interactivo, funcional y atractivo. Convierte las etiquetas de HTML en objetos programables lo que permite manipularlos más fácilmente. Todos los cambios de propiedades se producen sin que tenga que volverse a cargar la página desde el servidor, y este no requiere una configuración especial. Mejora la presentación de un sitio web, con facilidad y flexibilidad pues permite separar el contenido de las

propiedades de cada elemento. Tiene ciertos problemas de estandarización sin embargo, pueden ser corregidos con JavaScript.

1.6 XML

XML significa Extensible Markup Lenguaje, o lenguaje de anotación extensible. Es una tecnología muy sencilla que se vincula con otras tecnologías que la complementan y hacen que sea mucho más grande y con posibilidades mayores. Es un lenguaje para información auto-descrita. Tiene un papel fundamental en la actualidad ya que tiende a la globalización y a la compatibilidad entre los sistemas, siendo la tecnología que permite compartir la información de forma segura. Los documentos XML se usan dentro de entornos de aplicaciones. Este entorno permite publicar documentos XML que sufren transformaciones antes de ser enviados al cliente para adaptarlos a las condiciones del mismo.

Es un metalenguaje que permite definir lenguajes de presentación diferenciándolo del HTML que se centra en la representación de la información. Permite crear etiquetas según el contenido del documento siendo mucho más fácil de entender la representación, no posee etiquetas prefijadas lo que constituye la particularidad más importante del XML. Es un formato basado en texto, diseñado para almacenar y transmitir datos. Un documento XML está integrado por elementos XML, cada uno tiene una etiqueta de inicio y una etiqueta de fin y los datos comprendidos entre las etiquetas. A diferencia de HTML, XML no presenta un conjunto limitado de etiquetas. Las etiquetas indican lo que significa cada elemento de datos. Dentro de sus principales características está la extensibilidad, la representación estructural de datos, la separación de datos de la presentación y el proceso, separando la interfase de usuario y los datos estructurados permitiendo procesar y desplegar datos de cualquier forma al aplicar diferentes hojas de estilo y aplicaciones. Al ser independiente del HTML, se puede insertar código XML en documentos HTML.

XML presenta algunas limitaciones, este es útil solamente cuando las personas están de acuerdo en la forma en que se describen los datos.

XML tiene numerosas ventajas ya que permite la Comunicación de Datos es decir cualquier aplicación puede utilizar datos bajo formato XML y ser recibido por otra aplicación de forma que pueda trabajar con ella siempre y cuando esta información sea transferida por XML. Además permite la Migración de Datos, se puede mover datos de una base de datos a otra siempre y cuando se esté trabajando en formato XML. Permite un Código más Ligerero debido a que los documentos XML pueden ser estructurados para identificar cada pieza importante de información.

Es muy importante la utilización de esta tecnología ya que permite separar la información o el contenido de su presentación o formato, está diseñado para ser utilizado en cualquier lenguaje, permite extraer la información con poderosas técnicas, además las estrictas reglas a la hora de hacer un documento hace que sea mucho más fácil de analizar sintácticamente. Una de las aplicaciones fundamentales del XML es el formato de noticias RSS usado para describir el contenido de páginas web, si el contenido de la web cambia, el RSS se alimenta de dichos cambios y los da a conocer a los usuarios interesados. Además es utilizado para publicar los titulares de noticias, información, artículos, novedades sin tener que navegar por cada sitio buscando sus actualizaciones, simplemente RSS permite acceder a aquellas noticias y novedades con un reader (un lector) que automáticamente chequea todos los sitios indicados por el usuario y muestra los nuevos contenidos que son añadidos.

1.7 JSON

JSON es parte del estándar de la ECMA desde 1999. Se popularizó gracias a Douglas Crockford alrededor del 2002. Es el acrónimo para “JavaScript Object Notation”(Notación de Objetos de JavaScript), es un formato ligero para el intercambio de datos, es decir un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente

conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Por todas estas propiedades es que es un lenguaje ideal para el intercambio de datos. Es un subconjunto de la notación literal de objetos de Javascript pero no requiere el uso de XML.

Es una alternativa a XML en Ajax ya que como formato de intercambio de datos presenta la ventaja de que es más sencillo escribir un analizador semántico de JSON. Se emplea generalmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia por ejemplo en sitios donde la cantidad de usuarios sea grande.

Con respecto a XML JSON puede ser muy eficiente si se utiliza de forma efectiva, además es muy fácil entenderlo y trabajar con el, es rápido, más compacto y muchos programadores prefieren trabajar con JSON aunque los entornos en el servidor requieren que se incorpore una función u objeto analizador de JSON. Con el no son necesarias las etiquetas de cierre y esto reduce el tamaño con respecto al XML es decir tiene menor cantidad de código para representar los mismos datos, por eso Crockford dice que JSON es la alternativa sin grasas al XML, también que tiene una estructura que la mayoría de los lenguajes de programación pueden entender y utilizar. Es importante decir que JSON es mucho menos legible que XML porque en este último es muy fácil entender qué datos están siendo representados con un simple vistazo, de ahí la importancia de utilizar JSON en el servidor cuando hay que mandarle los datos al usuario.

Aunque el JavaScript puede entender e interpretar JSON puro desde el 2006 hay librerías JSON para el empleo de los lenguajes del lado del servidor más populares estas son por ejemplo C#.Net: Json.Net imita la funcionalidad de XML en .Net y es libre, también están ColdFusion: CFJSON, Perl, PHP: JSON-PHP, Python: json-py.

1.8 DOM

El modelo de Objeto del Documento (DOM) es una interfaz de programación de aplicaciones que puede utilizarse en numerosas variedades de entornos y aplicaciones, una representación en forma de objetos de la información contenida en un documento estructurado (XHTML, XML), permitiendo definir cómo acceder a un documento y manipularlo, así como la modificación de datos utilizando diversas tecnologías. Puede ser utilizado en cualquier lenguaje de programación. Permite definir la estructura lógica de los documentos, navegar por su estructura y añadir o eliminar elementos o contenido así como construir documentos. Es un modelo lógico que puede implementarse de cualquier manera.

1.9 Ajax

Ajax es una nueva tecnología que es nombrada por primera vez por Jesse James Garret, es el acrónimo para Asynchronous JavaScript + XML, que en realidad no es una tecnología sino la combinación de muchas tecnologías como son: HTML, XHTML y CSS para la presentación de la información, el DOM (Document Object Model) y JavaScript que permiten el intercambio directo con la información y XMLHttpRequest que es el protocolo sobre el que está apoyado, y muy importante, es verdaderamente el corazón de Ajax. Utiliza JavaScript para manejar el objeto XMLHttpRequest, el HTML que distribuye en la ventana del navegador los elementos de la aplicación y la información recibida por el servidor, CSS que define el aspecto de cada elemento y dato de la aplicación y XML que no es más que el formato de los datos transmitidos del servidor al cliente (navegador) y que posteriormente serán mostrados. (Anexo 1)

Cuando se interactúa con una aplicación Ajax lo primero que ocurre es que el usuario provoca un evento, luego se crea y configura un objeto XMLHttpRequest, este objeto realiza una llamada al servidor, después el servidor retorna un documento XML (XHTML, JSON) que contiene el resultado, posteriormente el objeto XMLHttpRequest procesa el resultado y finalmente se actualiza

el DOM (Modelo de Objetos del Documento) de la página asociado con el resultado devuelto. (Anexo 2) (5)

Con esta técnica se abre todo un mundo de posibilidades a la hora de construir aplicaciones web altamente interactivas. Según Jesse James Garret en su libro "Ajax: Un nuevo acercamiento a las Aplicaciones Web", define que Ajax no es más que "Cargar y reindexar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-reindexando la página y mostrando u ocultando porciones de la misma." (6)

Es una técnica que tiene numerosas características que hacen que cada día sea más potente, por ejemplo que es Basado en estándares abiertos y esto es debido a que de todas las tecnologías que lo conforman XML HTTP Request Object es el único que no es estándar, pero si es soportado por los navegadores que son más utilizados en Internet por ejemplo el Internet Explorer, Mozilla y Opera. También tiene Gran usabilidad: ya que permite a las páginas hacer una petición muy pequeña al servidor y recibirla sin necesidad de recargar la página completa. Es válido en cualquier plataforma y navegador: ya que es muy fácil programar aplicaciones Ajax en los navegadores que tienen el número uno en el mercado de Internet, los cuales son Internet Explorer, Mozilla y Firefox; pero también se pueden construir aplicaciones web basadas en Ajax que funcionen en otros navegadores. Es Independiente del tipo de tecnología de servidor que se utilice: porque como mismo Ajax funciona con cualquier navegador, es compatible con cualquier tipo de servidor estándar y lenguaje de programación web, ejemplo: PHP, ASP. ASP.Net, Perl, y gracias a que es completamente compatible el desarrollo de estas tecnologías hace que Ajax sea cada día más popular y mucho mejor. Además Mejora la estética de la web: ya que con Ajax se puede combinar toda la imaginación del desarrollador con la usabilidad de una aplicación web de forma tal que si una aplicación no estuviera dentro de un navegador, pudiera pasar por una aplicación normal.

Es una plataforma tecnológica y además es un estilo de arquitectura ya que es un grupo de tecnologías que se usan conjuntamente de forma concreta. Es de gran importancia y necesidad

la utilización de Ajax ya que esta técnica para el desarrollo de aplicaciones web es muy diferente a las técnicas tradicionales, es decir a las de toda la vida, porque elimina la siempre seguida forma de trabajar de una aplicación, es decir evita el “arrancar-frenar- arrancar-frenar”, y lo logra con un motor Ajax que va entre el usuario y el servidor, el cual está escrito en JavaScript y generalmente sacado en un frame oculto. Todo parece indicar que se le está agregando otra capa a la aplicación, la que la haría menos reactiva, pero en realidad es todo lo contrario la hace mucho mejor ya que el usuario nunca mirará una página en blanco o esperará mirando el ícono de un reloj de arena a que el servidor haga lo que tiene que hacer para devolver la petición. Es importante la aclaración de que cualquier respuesta a una petición del usuario en la cual no sea necesaria una llamada al servidor ya sea porque es una simple validación de datos o para editar datos de memoria o algo de navegación será manejada por el propio motor Ajax. Sin embargo si la petición necesita de una llamada al servidor para dar la respuesta porque necesita enviar datos para procesar, cargar código adicional o recuperar datos adicionales entonces hace las peticiones de forma asíncrona utilizando XML sin tener la necesidad de frenar la interacción del usuario con la aplicación.

A la hora de analizar Ajax como técnica para desarrollar aplicaciones web hay que referirse a las ventajas que hacen que cada día esta técnica sea más popular y más usada a nivel mundial, la primera ventaja es que utiliza tecnologías ya existentes. Permite la recuperación asíncrona de datos, es decir el usuario no tiene que esperar después de una petición. Permite acercar metafóricamente las aplicaciones de escritorio a la web, es muy importante mencionar que no requiere plugins esto es relacionado con la portabilidad, y se reduce el tamaño de la información intercambiada. Además es soportada por la mayoría de los navegadores modernos. En cuanto a la interactividad el usuario no tiene que esperar hasta que lleguen los datos del servidor. En cuanto a la velocidad es mucho mayor y esto es debido a que no hay que retornar toda la página nuevamente.

Es muy importante mencionar las principales dificultades que trae consigo la utilización de esta técnica ya que pueden aumentar las llamadas al servidor. El uso de Ajax en aplicaciones web

puede acabar con la tradicional utilidad del botón "atrás" del navegador. También existe el peligro de la incompatibilidad en navegadores. Se rompe el flujo de navegación tradicional (botón "volver" y "actualizar"). Si se guarda en favoritos no necesariamente al visitar nuevamente el sitio se ubique donde se encontraba al grabarla. La percepción de cambio es menor, por tanto se debe indicar al usuario que ha habido un cambio o que se va a producir ("cargando..."), esto es en cuanto a la usabilidad ya que un usuario acostumbrado a un modelo tradicional de interacción web entiende que cuando la página se refresca quiere decir que el sistema ha recibido su petición, sin embargo con Ajax es conveniente replantear el modelo de respuesta de la pantalla para que el usuario perciba que el sistema le está atendiendo, un ejemplo de aplicación Ajax que muestra lo planteado anteriormente es el botón del cargamento de GMail que está en la derecha superior de la página. Es necesario tener la precaución de mantener el funcionamiento tradicional de los formularios, es decir no se debe enviar los datos al servidor hasta que el usuario lo solicite. Con Ajax no se puede modificar el navegador, no permite el acceso a ficheros locales, no permite ejecutar ficheros musicales ni el acceso al hardware. Requiere programadores que conozcan todas las tecnologías que intervienen en Ajax. En cuanto a la accesibilidad existe el riesgo de que cierta cantidad de usuarios no puedan usar aplicaciones Ajax, esto es debido a que está basado en JavaScript, es decir que tengan dispositivos que no soporten JavaScript o que lo tengan desactivado y la solución sería acudir al modelo tradicional de refresco de páginas, es decir diseñar servicios que puedan ser usados sin necesidad de utilizar JavaScript. Es muy importante decir también que con Ajax hay que ser muy cuidadosos al diseñar flujos de pantallas ya que se puede caer en el error de ocultar información que puede ser idónea como documento ya sea para consultar, buscar o navegar. Finalmente dependiendo de la carga del servidor es posible experimentar tiempos tardíos de respuesta que desconciertan al visitante. (7)

Para programar y que tus web tengan funcionalidades Ajax no necesitas instalar nada en el servidor. Cuando se diseña un proceso web con Ajax no tiene que ser más rápido que el diseñado de la forma tradicional, lo que si puede pasar es que el usuario percibe que todo funciona más rápido debido a que le estás permitiendo interactuar con la aplicación de una forma

más parecida a la forma con la que interactúa con las aplicaciones de escritorio. El uso de Ajax aumenta la complejidad del código de programación de una web y es necesario valorar muy bien en qué momentos se debe utilizar Ajax y en cuáles optar por otras soluciones. Lo que verdaderamente se consigue con la tecnología Ajax es cargar una sola vez la página y dar la opción al usuario de interactuar con ella de una forma diferente ya que anteriormente requeriría varios refrescos ya sea consultando, guardando información o editándola.

Es importante saber que los principales desafíos a la hora de crear aplicaciones Ajax no son técnicos ya que las principales tecnologías que la conforman son estables y bien conocidas. Sin embargo el verdadero desafío es para los desarrolladores que deben olvidar las limitaciones de la web y comenzar a imaginarse todo con un nivel más amplio de posibilidades.

Existen numerosas interrogantes en cuanto a que tecnología sería más conveniente utilizar para desarrollar una aplicación web, estas técnicas son Ajax o Flash. Flash a sido hasta hace poco una herramienta muy buena para diseñadores y animadores. Prácticamente todos los ordenadores del mundo lo tienen instalado, lo que no implica que el usuario de un ordenador tenga instalada la última versión del mismo. Presenta calidad visual, sobre todo con el avance en el manejo de los textos, animación, multimedia, contenido interactivo, video, audio, así como lenguaje de programación de alto nivel orientado a objetos. Una de las desventajas que presenta es que depende de un plugin propietario, se tiende a abusar de las animaciones, es poco accesible. Ajax es basada en XHTML, es decir más estandarizable, accesible, por su integración con CSS es sencillo separar los datos de la presentación, es más ligera ya que una página con HTML es más ligera que en flash, aunque el JavaScript necesario para su funcionamiento no es tan ligero, por lo que al final se depende de mecanismos de caché en servidor. También presenta lenguajes interpretados en el navegador, sencillo, orientado a objeto. La desventaja es que el usuario puede deshabilitar el JavaScript en el navegador, que depende de un objeto propiedad de Microsoft para las comunicaciones asíncronas. No es tan ligero y por mucho que se intente el resultado final no se puede comparar al de una web en Flash. Finalmente no debe ser el uso de una u otra tecnología lo que defina si una solución es buena o

no, sino el buen uso que se haga de ella y la forma en que se utiliza, tanto Flash como Ajax se pueden utilizar para hacer cosas buenas y malas.

Como se había planteado anteriormente Ajax comienza a marcar una época en el desarrollo de la web sobre todo cuando se habla de aplicaciones de escritorio, pero por otro lado existe otra tecnología que tiene los mismos objetivos es decir pretende ofrecer funcionalidades de las aplicaciones de escritorio sobre la web esta técnica es E-DES el cual se basa en JavaScript, en ficheros de definición (ficheros DF), DOM y CSS. Es importante ver que E-DES no es solo una técnica que tiene fundamentos similares a Ajax sino que además es un entorno de programación. Esta tecnología está basada en LAMP lo que hace que sea más atractivo en el mundo del Software Libre. También tiene partes débiles como por ejemplo su compatibilidad ya que este desde sus inicios se desarrolló con el DOM de Microsoft ya que era el navegador que en aquellos tiempos se utilizaba en la gran mayoría de las computadoras, además de adicionar que desde sus inicios está pensada para desarrollar Intranet Corporativas y no web abiertas a todo el mundo.

1.10 Casos de estudio y aplicaciones

La tecnología para desarrollar aplicaciones Ajax es una expectativa en todo el mundo, por lo existen en la actualidad muchas aplicaciones Ajax. Por la riqueza de la funcionalidad combinada con simplicidad, interactividad, gratuidad entre otras características es que se pueden seleccionar las mejores aplicaciones Ajax como son por ejemplo Kiko que es un calendario es decir una agenda online, Backbase's RSS Reader que no es más que un lector RSS, Backpack que es una aplicación de gestión "Project Management", incluye e-mail y avisos por e-mail, Writely que es un procesador de textos online, TimeTracker que es una herramienta de planificación del tiempo personal, Protopage que sirve como portal personal de información, Periodic Table of the Elements que no es más que la tabla periódica de los elementos, Voo2do que es un buen administrador de tareas y proyectos, Colr es un sitio para jugar con los colores, LfVr para navegar

en Flickr de forma rápida y fácil, TaskTHIS es un gestor de tareas muy interesante, TreeHouseChat que es un Chat, Meebo que como todos saben es un mensajero instantáneo que conecta a las principales redes que son ICQ, Yahoo, MSN y AIM, Plazo que es un calendario que tiene muy buen diseño y utiliza mucho Ajax.

Es importante mencionar también las bien conocidas Google maps, Google suggest, Google groups y Gmail que son desarrolladas por grandes empresas como Google.

Google Groups es un servicio que almacena todos los contenidos publicados en los 'newsgroups' de Usenet desde 1981, los 'newsgroups' de Usenet eran grupos de discusión en Internet surgidos en 1979 en los cuales los usuarios debatían sobre muchos asuntos, divididos en centenares de grupos. En fin permite realizar búsquedas dentro de toda esta información almacenada. Tiene una característica muy interesante y es que crea un ambiente seguro en la Web que no es totalmente público como las páginas tradicionales, ni totalmente privado como el correo electrónico, esta característica permite que la transferencia de datos sea restringida aunque también sencilla. Google lanzó una nueva plataforma para el trabajo de grupo en línea Groups 3 que permite no sólo intercambiar mensajes sino también crear páginas web personalizadas y diseñadas por el usuario. También provee un espacio para el trabajo en colaboración como es la inclusión de funcionalidades destacadas para la creación de portales virtuales y la apariencia personalizada que permite seleccionar fotos, colores y estilos para expresar el contenido.

Google Suggest es un nuevo recurso de Google que va mostrando, según se va escribiendo en el recuadro de las palabras de búsqueda, posibles términos junto al número de resultados que devolvería, es decir al introducir una palabra en el buscador y pulsar la barra espaciadora te irá proponiendo combinaciones de palabras lógicas asociadas a la que está puesta, y así sucesivamente seguirá sugiriéndote otra a continuación, en base a las anteriores, hasta que se quiera terminar.

Google Maps es un servidor de aplicaciones de mapas en Web, ofrece imágenes de mapas desplazables y fotos satelitales del mundo entero así como la ruta entre diferentes ubicaciones. También tiene la capacidad de hacer acercamientos o alejamientos para mostrar el mapa. Los

usuarios pueden ingresar una dirección, una intersección o un área en general para buscar en el mapa, los resultados de la búsqueda pueden ser restringidos a una zona.

GoogleMail es un servicio gratuito de correo electrónico, ofrecido por Google, con más de 2.600 megabytes de almacenamiento. El programa permite realizar búsquedas de la forma más fácil y rápida. Gmail no contiene pop-ups ni anuncios de banner irrelevantes, sólo pequeños anuncios de texto. Integra la función de mensajería instantánea en el correo electrónico. Es una nueva perspectiva de comunicación. Es importante señalar que se basa en una filosofía que persigue la simplicidad y eficacia en las comunicaciones.

1.10.1 CMS

Los sistemas de gestión de contenidos CMS son de las herramientas que más se emplean hoy en el desarrollo de aplicaciones web, porque permiten la creación y administración de los contenidos desde la propia aplicación web, además brindan la posibilidad de definir flujos de trabajos, dan permisos y lo fundamental separan el diseño del contenido. Existe un número importante de CMS que hacen posible todas estas funcionalidades. Gran cantidad de estas aplicaciones, al menos las más importantes, como lo pueden ser Joomla, Drupal, etc, ya incluyen aspectos donde incluyen la tecnología Ajax.

Ejemplo, la nueva versión de Drupal, Drupal 5.x utiliza Ajax en una serie de módulos que se han ido desarrollando por su comunidad, como es por ejemplo Ajax Form Builder el cual es una aplicación web basada en esta tecnología para generar formularios. También existen otros módulos tales como el Spajax el cual permite a otros módulos animar textos, el AjaxSubmit que se utiliza para crear formas con solo agregar a la definición de la misma el siguiente código: `$form['#ajaxsubmit'] = TRUE`, y el módulo ActiMenu que permite la creación de menús, entre otros, los cuales requieren la instalación previa del módulo JTools que vincula el JavaScript y la tecnología Ajax.

1.10.2 Frameworks

Un framework es una estructura de soporte, en la cual cualquier otro proyecto puede ser desarrollado y organizado. Puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar los diferentes componentes de un determinado proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, generando una estructura y una metodología de trabajo que extiende o utiliza las aplicaciones del dominio.

Existen numerosos frameworks disponibles que facilitan la implementación de soluciones Ajax, los más populares son: Dojo, Prototype, jQuery, script.aculo.us, MochiKit, moo.fx ; con respecto a esto Ajaxian que es un sitio dedicado a Ajax hace una publicación con el resultado de una encuesta que ha realizado entre sus lectores sobre sus preferencias a la hora de utilizar un determinado entorno, dicha encuesta arrojó como resultado que Prototype es el frameworks más utilizado es decir el más popular y todo esto debido a que es basado en JavaScript, con él se logra un desarrollo sencillo, potente y dinámico de páginas web ya que este simplifica el trabajo cuando se pretende desarrollar páginas altamente interactivas, facilita la programación OO (orientada a objetos), extiende objetos básicos de JavaScript para proporcionar nuevas funcionalidades como por ejemplo String, Array. Luego le sigue Script.aculo.us y seguidamente Dojo. (Anexo 3) (8)

Conclusiones

Este capítulo ha brindado la posibilidad de argumentar la verdadera importancia o mejor dicho las verdaderas ventajas de la utilización de la técnica Ajax, ya que se han explicado las técnicas que la conforman las cuales son muy ventajosas y hacen en su conjunto que Ajax sea cada día más abarcadora en el mundo de la web.

CAPÍTULO 2: INDICACIONES PARA LA UTILIZACIÓN DE AJAX

Introducción

En este capítulo se argumentará acerca de cuándo debe usarse Ajax es decir los principales usos que tiene esta tecnología también cuándo no debe usarse y todo esto por sus características, ventajas y desventajas. Además se explicará paso a paso cómo elaborar una aplicación web utilizando esta técnica que brinda la gran posibilidad de acercar las aplicaciones web a verdaderas aplicaciones de escritorio.

2.1 Principales usos de Ajax

Comenzar planteando que Ajax está ubicada entre las tecnologías más importantes a la hora de desarrollar una aplicación web ya que aporta una gran madurez en este sentido. Es importante tener en cuenta que un uso restringido de esta técnica puede repercutir a la hora de conseguir el desarrollo de aplicaciones mucho mejores. Es una de las opciones más eficaces para mejorar la navegación web ya que abre posibilidades de interacción entre el navegador y un servidor web porque cuando el usuario efectúa una petición determinada Ajax ya tiene listos esos datos y los muestra directamente sin tener que volver a hacer una petición al servidor y muy importante se pueden enviar peticiones al servidor web para obtener únicamente la información necesaria.

Gracias a Ajax con la mezcla de tecnologías que la conforman se dinamiza el uso de Internet y se impulsa la llegada de la web 2.0 que es un nuevo sistema en el que la mayoría de herramientas que ahora son utilizadas como son los procesador de textos, las hojas de cálculo, entre otros estarán en línea y se utilizarán a partir del navegador y no del sistema operativo.

Con la tecnología Ajax existe la posibilidad de crear un paginador, el cual se utiliza cuando la cantidad de elementos que se quiere cargar es muy grande, paginar las listas de elementos en páginas es un proceso sencillo y muy útil para la navegación a través de las páginas ya sea para

la página siguiente o para los elementos anteriores, con Ajax se pueden programar paginadores con controles mucho mejores por ejemplo el scroll del ratón. También es usado para diferentes tareas como por ejemplo actualizar y eliminar registros, expandir formularios web y devolver peticiones de búsqueda o editar árboles de categorías, todo esto sin necesidad de recargar la página.

Se pueden enumerar las diferentes utilidades de la tecnología Ajax como es por ejemplo el Contenido Dinámico es decir la posibilidad de modificar contenido dinámicamente de forma rápida y eficiente ha abierto muchas puertas como Flickr que es un sistema de gestión de fotografías digitales que brinda la posibilidad de hacer búsquedas de imágenes ya sea por etiquetas o por fechas ya que brinda muchísimas facilidades como son por ejemplo que clasifica las fotos por categorías, también permite escribir perfiles del autor y además enviar imágenes por correo electrónico con comentario. Otro ejemplo es el Contenido Meable ya que la posibilidad de personalizar tu entorno y que automáticamente se guarde, ha hecho realidad sitios como Netvibes que no es más que un sitio pensado para que sea la página de inicio del usuario en la que se centralice el correo y otros servicios importantes, es decir en un rato y con un par de clic resulta muy útil y es así realmente la configuración de una página de inicio personalizada con los servicios que el usuario desee o con la información necesaria, también los Suggests porque sugerir a los usuarios lo que están buscando es algo que ayuda muchísimo al visitante.

Existen cierta cantidad de situaciones en las cuales debería usarse la tecnología Ajax como por ejemplo en los Formularios Interactivos que son los que permiten recoger datos de los visitantes y esto es a través de diferentes elementos como por ejemplo los radio-botones que son los que permiten seleccionar solo una opción, los cuadros marcables que son los que brindan la posibilidad de marcar varias opciones a la vez o las listas desplegables. También están las Encuestas que no son más que la recopilación de datos obtenidos mediante consultas, es decir, encuestas realizadas con el fin de conocer estados de opinión o hechos específicos que le interese a alguien determinado. También las Ayudas Textuales y Auto Llenado de Campos, además Filtros y Manipulación de Datos.

Capítulo 2 Indicaciones para la utilización de Ajax

Existen disímiles formas de utilizar Ajax dentro de ellas hay diez que son las más importantes por ejemplo en la Interacción Manejada por Formularios: esta forma de editar bookmark(marcadores) es muy sencilla y con Ajax aún más ya que simplemente es dar clic en el vínculo de editar para instantáneamente comenzar a cambiar las etiquetas, luego se pincha en el botón de enviar para de esta forma enviar los cambios asincrónicamente y comenzar a ver qué cambió sin recargar la página completa, dentro de estas utilidades de interacción manejadas por formularios existen dos más la primera es el Menú de Selección: las cuales con Ajax son mucho más fácil ya que a medida que se van seleccionando opciones va mostrando si lo que se solicita existe o no por ejemplo se quiere comprar un pullover azul de marca Polo en este caso de la forma tradicional cuando se termina de escoger si este pedido no existe en el almacén devuelve una página donde dice que no existe y se tiene que comenzar de nuevo, todo lo contrario utilizando Ajax, el otro ejemplo es para el autosalvado lo cual permite crear funciones con JavaScript donde los documentos por ejemplo los escritos en Word se autosalven cada cierto tiempo. (9)

Otra es la Navegación Profunda de Árboles Jerárquicos: esto generalmente es una pesadilla. Las topologías simples casi siempre funcionan pero si no hay otra opción se debe utilizar JavaScript para gestionar la interfaz de usuario y Ajax para disminuir la carga del servidor por tener que mostrar datos jerárquicamente muy profundos.(Idem 9)

También la Rápida Comunicación usuario-usuario: donde usando Ajax se puede prescindir en las aplicaciones de comunicación instantánea de estar refrescando una y otra vez la página para ver los nuevos mensajes , incluso GoogleMail que mejora lo que hacía Yahoo y Hotmail podría explotar más la utilización de Ajax en este sentido. (Idem 9)

Las Votaciones, cuadro si/no: estas aplicaciones con Ajax se vuelven más interactivas. Si registrar un voto toma 40 segundos las personas pasarán de ello, sin embargo si tomara un segundo un porcentaje mayor de persona si participarían en las votaciones. (Idem 9)

Filtrado y Manipulación de Datos: Aplicar un filtro, organizar por fecha y nombre, cambiar el estado de los filtros(es decir poner o quitar filtros), etc. Cualquier manipulación de datos interactiva debería ser realizada en JavaScript en lugar de una serie de solicitudes al servidor.

Capítulo 2 Indicaciones para la utilización de Ajax

Encontrar y manipular cantidades de datos es bastante duro si hay que esperar 30 segundos entre cada cambio de vista. Con Ajax este proceso se puede acelerar. (Idem 9)

La Entrada de texto común, Consejos y Autocompletamiento: JavaScript es realmente conveniente a la hora de autocompletar textos y como se plantea anteriormente Ajax utiliza JavaScript en su concepto. (Idem 9)

Las Consultas que toman mucho tiempo y llamadas remoto: Si una consulta o llamada a un webservice (servicio web) remoto va a tomar mucho tiempo y no puede ser evitado, Ajax gestiona muy bien el tiempo que el usuario debe esperar por el retorno de la llamada. Por ejemplo los usuarios no tienen que esperar que el webservice de Google responda para realizar otros proyectos. (Idem 9)

Las Operaciones computacionalmente caras: Desafortunadamente JavaScript tiene una tendencia a ser muy lento, los cálculos matemáticos complejos no es el fuerte de JavaScript. Una llamada al protocolo XMLHttpRequest puede ser útil para enviar los cálculos estos que son complejos a servidores remotos. (Idem 9)

Además las Salvas del Servidor: Algunas veces un proceso de usuario solicita pequeñas cantidades una y otra vez para ser enviados, pero la carga de las páginas completas puede restringir el ancho de banda y recursos de los servidores. Ajax puede ser usado para cargar páginas más eficientemente. (Idem 9)

Finalmente las Vistas y Movimientos interactivos sobre los datos: donde al mover y mostrar grandes cantidades de datos se hace impracticable cargar todos los datos. Cargar solo lo que está frente al usuario da la impresión de que todos los datos son accesibles mientras que ayuda a reducir los tiempos de carga. Un ejemplo de esto es el sistema de tejas (tiles) de desplazamiento de Google Map que brinda el efecto de mover en un mapa, recogiendo los tiles de atrás y colocándolos delante del usuario, llenándolos con nuevos datos, solicitados vía Ajax. (Idem 9)

2.2 Cuando no usar Ajax

Las aplicaciones Ajax combinadas con una red de pobre conexión o poco ancho de banda, pueden hacer realmente difícil manejar una interfaz ya que cada vez que se hace algo hay que esperar que el servidor devuelva una respuesta, pero es bueno señalar que la 'A' de Ajax es de asincrónico, es decir la interacción puede ser diseñada para que el usuario continúe trabajando en la página mientras se procesa la petición.

La seguridad de las aplicaciones Ajax está sujeta a las mismas reglas de cualquier web, excepto que cada vez que se comunicarse con el servidor se puede enviar información personal de una manera potencialmente insegura, todo el tráfico debe revisarse para cerciorarse de que la seguridad no está comprometida. Ajax introduce una manera de hacer usos mucho más interesantes del JavaScript, lo que significa más funcionamiento del código y más funcionamiento del código significa más trabajo para el browser, es decir se necesita tener una CPU de gran alcance para mantener las funcionalidades, es importante decir que el problema de la CPU ha sido realmente un límite en funcionalidad del JavaScript.

Los usos de Ajax que cargan cantidades grandes de texto sin una recarga pueden causar un problema grande para los motores de búsqueda, esto va al problema de la URL, es decir si los usuarios pueden entrar a través de los motores de búsqueda, el texto del uso necesita ser algo estático de modo que las arañas puedan leerlo.

Ajax provoca alteraciones graves en la navegación, es decir los botones "Atrás", "Adelante", "Actualizar" o la opción de "Añadir a Favoritos" no funcionan como se espera de ellos; con un arduo trabajo de programación se pueden mitigar estas alteraciones, tal vez GMail es la aplicación que más cerca está de conseguirlo.

Uno de los problemas mas significativos de la gran mayoría de las aplicaciones Ajax es la baja compatibilidad entre navegadores, puesto que la capa JavaScript es de una gran complejidad y a menudo por falta de experiencia en el lenguaje se opta por programar solo para Internet Explorer.

No debe usarse Ajax en Formularios simples, Búsquedas, Navegación Básica o en el Reemplazamiento de Textos, ya que para todas estas funcionalidades tan pequeñas se emplearía un código JavaScript bastante grande, el cual se ejecuta directamente en el cliente y en vez de ganar en eficiencia, se puede caer en un proceso bastante lento sin necesidad de ello.

En resumen la tecnología Ajax es súper potente, pero se debe usar con mucho juicio y solo si es realmente necesaria. Además, es conveniente no usar Ajax a la ligera sino apoyado por un framework que le permita abstraerse de la tecnología subyacente.

2.3 Como utilizar Ajax

Primero que todo comenzar planteando que el corazón de Ajax es el objeto XMLHttpRequest el cual permite realizar una conexión al servidor y al enviarle una petición y recibir la respuesta que se procesa en el código Javascript, se está hablando del verdadero motor de Ajax.

El objeto XMLHttpRequest es un API que permite la transferencia de datos en formato XML desde los script del navegador que son por ejemplo JavaScript, VBScript a los del servidor o inversamente, los del servidor son PHP, ASP, Perl entre otros y es lo primero que hay que hacer para realizar una aplicación web. Este objeto trae incluidos atributos y métodos por ejemplo:

2.3.1 Objeto XMLHttpRequest

- El atributo readyState es el encargado de devolver el estado actual del objeto XMLHttpRequest, es decir cada vez que se cambia el valor de readyState se lanza la función indicada en onreadystatechange. Es utilizado de la siguiente forma:

```
iEstado = oXMLHttpRequest.readyState;
```

donde iEstado no es más que el estado en el cual se encuentra el objeto, los cuales pueden ser: 0 cuando se encuentra sin inicializar y siempre será de la siguiente manera: 1 para indicar que se

encuentra abierto es decir acaba de llamar la función open, 2 para indicar que está en el estado de enviado, 3 cuando es el estado de recibiendo, 4 cuando está a punto de recibirlo.

La propiedad readyState es utilizada en todas las comunicaciones asíncronas para comprobar que es posible acceder a los atributos responseXML y responseText los cuales solo son accesibles en los estados 3 y 4.

Este es un ejemplo donde una vez creado el objeto XMLHttpRequest (oXMLHttpRequest) alertará del texto recibido de modo asíncrono:

```
<script language="JavaScript">  
function fFuncion () //Se crea la función.  
{  
if (oXMLHttpRequest.readyState==4) //Si el estado es "Apunto".  
{  
alert (oXMLHttpRequest.responseText); //Se alerta el texto.  
}  
}  
oXMLHttpRequest.onreadystatechange = fFuncion; // Se Define la función  
                                // handler del evento.  
</script>
```

- El atributo onreadystatechange es el encargado de asignar la función que se ejecutará cada vez que readyState cambie de valor. Es utilizado de la siguiente forma:

```
oXMLHttpRequest.onreadystatechange = fFuncion ;
```

donde oXMLHttpRequest es el objeto XMLHttpRequest y fFuncion es la función que se va a ejecutar.

El onreadystatechange es utilizado para definir una función la cual lee los datos recibidos del servidor, en este caso en su interior se comprobaría que readyState tenga valor 4 y entonces se leerá el valor de responseXML, responseText.

Ejemplo en el cual una vez creado el objeto XMLHttpRequest alertará el estado del readyState cada vez que se cambie:

```
<script language="JavaScript">  
function fFuncion () // Se crea la función.  
{  
    alert (oXMLHttpRequest.readyState); // Se alerta el estado de la petición.  
}  
oXMLHttpRequest.onreadystatechange = fFuncion; // Se define la función  
                                     // handler del evento.  
</script>
```

- El atributo responseText es el encargado de devolver el texto del documento descargado del servidor en una petición con XMLHttpRequest y es utilizado así:

```
sDocumento = oXMLHttpRequest.responseText;
```

donde sDocumento es la cadena de caracteres con el texto del documento.

Esta propiedad responseText es utilizada para tratar los datos recibidos desde el servidor que no tienen formato XML, se puede acceder a los datos siempre y cuando el estado de la conexión devuelto con readyStatechange sea igual a 3 es decir recibiendo o 4 a punto. Siempre que sea posible se debe utilizar responseXML en vez de responseText para dar los datos y no texto plano.

Un ejemplo de como utilizarlo ya se ha visto en el ejemplo del atributo readyState donde alerta del texto recibido en forma asíncrona una vez creado el objeto.

- El atributo responseXML se encarga de devolver una referencia al cuerpo del documento descargado del servidor en una petición con XMLHttpRequest en formato XML. A la hora de utilizarlo se pone así:

```
oDocumento = oXMLHttpRequest.responseXML;
```

donde oDocumento no es más que la referencia al cuerpo del objeto recibido.

La propiedad responseXML es utilizada para tratar los datos en formato XML desde el servidor, es posible acceder a los datos siempre que el estado de la conexión devuelto con readyStatechange sea igual a 4. Devuelve nulo siempre que la respuesta XML del servidor no tenga el encabezado text/xml, application/xml o acabe en +xml y finalmente se utilizarán las propiedades del Modelo de Objetos de Documento (DOM) para tratar los datos XML recibidos.

Este es un ejemplo donde después de haber creado el objeto XMLHttpRequest y realizado una petición de un xml con marcas <ítem> alertará el número de item's:

```
<script language="JavaScript">  
    function fFuncion () // Se crea la función.  
    {  
        if ( oXMLHttpRequest.readyState == 4) //Si el estado es a punto.  
        {  
            oDocumento = oXMLHttpRequest.responseXML.documentElement; //Se  
                //accede al  
                //documento  
                //XML.  
            alert ( oDocumento.getElementsByTagName('item').length ); // Se alerta el  
                // número de  
        }  
    }
```

```
                                //item's.
                                }
                                }
                                oXMLHttpRequest.onreadystatechange = fFuncion; // Se define la función
                                //handler del
                                //evento.
</script>
```

- El atributo statusText devuelve el código del estado HTTP de la transmisión devuelto por el servidor web. Es utilizado de la forma siguiente:

```
iEstado = oXMLHttpRequest.status;
```

donde *iEstado* es un entero con el código HTTP de estado.

Esta propiedad es utilizada para comprobar que no hay problemas en la comunicación con el servidor, se puede acceder a los datos siempre que el estado de la conexión devuelto con readyStatechange sea 3 o 4. El código de estado para una transmisión que esté correcta es el 200, es necesario comprobar este dato antes de acceder a los datos con responseText y responseXML. (Anexo 2)

Este es otro ejemplo pero en este caso se muestra un error cuando la conexión no ha sido satisfactoria todo esto una vez creado el objeto XMLHttpRequest:

```
<script language="JavaScript">
    if ( oXMLHttpRequest.status != 200 ) // Se compara el estado.
    {
        alert ('Ha ocurrido un error con la conexión' ); //Se muestra un mensaje de
```

error.

```
    }  
</script>
```

- El atributo statusText es el encargado de devolver el estado HTTP de la transmisión devuelto por el servidor web. En cuanto a la hora de utilizarlo se pone de la siguiente forma:

```
sEstado = oXMLHttpRequest.statusText;
```

donde *sEstado* es la cadena con el texto del estado HTTP.

Esta propiedad no es de uso común en su lugar generalmente se utiliza `status`, se puede acceder a los datos siempre que el estado devuelto con readyStatechange sea 3 o 4,

En este ejemplo se muestra el texto de estado una vez creado el objeto XMLHttpRequest:

```
<script language="JavaScript">  
{  
alert ( oXMLHttpRequest.statusText); // Se compara el estado.  
}  
</script>
```

- Entre los métodos del XMLHttpRequest está el abort que es el que detiene todas las conexiones asíncronas abiertas por el objeto XMLHttpRequest y es reinicializado poniendo en cero su estado, es decir el readyState. Es utilizado así:

```
oXMLHttpRequest.abort ();
```

Generalmente es utilizado antes de realizar una nueva petición al servidor a través de un objeto que está realizando o recibiendo otra petición anterior.

Este es un ejemplo donde se realiza una petición liberando antes el objeto oXML previamente creado:

```
<script>  
{  
    oXML.abort (); // Se libera el objeto.  
    oXML.open('GET', 'archivo.txt'); // Se envían los datos.  
    oXML.onreadystatechange = leerDatos; // Se prepara la recepción.  
    oXML.send(""); // Se realiza la petición.  
}  
</script>
```

- El método getAllResponseHeaders es el encargado de devolver en una sola cadena de caracteres los encabezados HTTP que se han recibido del servidor en una conexión usando el objeto XMLHttpRequest. A la hora de utilizarlo debe ponerse así:

```
sHeaders = XMLHttpRequest.getAllResponseHeaders ();
```

donde sHeaders no es más que la cadena de caracteres con los encabezados y se podrá acceder a los encabezados siempre que el estado devuelto con readyStateChange sea 3 o 4.

Ejemplo que muestra en un alert todos los encabezados recibidos en una conexión:

```
<script language="JavaScript">  
  
    oXMLHttp = new XMLHttpRequest(); // Se crea el objeto.  
    oXMLHttp.open('get', 'index.htm', false); // Se pide la página en modo síncrono.  
    oXMLHttp.send(); //Se envían los datos.  
    alert(oXMLHttp.getAllResponseHeaders()); // Se alertan los encabezados HTTP.
```

</script>

- El método getResponseHeader se encarga de devolver en una sola cadena de caracteres uno de los encabezados HTTP que se han recibido del servidor en una conexión usando el objeto XMLHttpRequest. Es utilizado de la siguiente manera:

```
sHeaders = XMLHttpRequest.getAllResponseHeaders (sHeaderName);
```

donde sHeaders es la cadena de caracteres con los encabezados y sHeaderName es la cadena de caracteres con el nombre del encabezado y bueno la cadena sHeaders contendrá todos los encabezados con nombre igual a sHeaderName enviados por el servidor.

Siempre se puede acceder a los encabezados cuando el estado de la conexión devuelto con el readyStateChange sea igual a 3 o 4.

Ejemplo en el cual se alertan todos los encabezados 'X-Powered-By' recibidos en una conexión.

script language="JavaScript">

```
oXMLHttp = new XMLHttpRequest(); //Se crea el objeto.
```

```
oXMLHttp.open ('get', 'index.htm', false); //Se pide la página en modo síncrono.
```

```
oXMLHttp.send (); //Se envían os datos.
```

```
alert(oXMLHttp.getResponseHeader('X-Powered-By')); //Se alertan los encabezados HTTP.
```

</script>

- El método open es el encargado de preparar la conexión HTTP a través del objeto XMLHttpRequest con un método y una URL determinados, además este método es el encargado de iniciar todos los atributos del objeto. A la hora de utilizarlo debe ponerse así:

oXMLHttpRequest.open (sMetodo, sURL [, bSincronia [, sUsuario [, sPwd]]]);

donde sMétodo es un String con el método de conexión ya sea GET o POST, sURL es la URL para la petición HTTP, bSincronia es una variable booleano que es opcional cuando está en verdadero es para usar modo asíncrono y cuando está en falso es para usar modo síncrono, sUsuario es opcional y no es más que la cadena de caracteres con el nombre de usuario para la autenticación, sPwd que también es opcional es la cadena de caracteres con la contraseña del usuario (sUsuario) para la autenticación.

Es sumamente importante señalar que no se permiten las llamadas a dominios protocolos o puertos diferentes al de la página que llama la función. Generalmente se utiliza el método GET para permitir que el navegador guarde los datos en caché y POST para obligar a descargarlos de nuevo. Los parámetros sUsuario y sPwd solo serán usados para enviar si se recibe una respuesta 401 (Access Denied), mientras que por URL se usarán siempre. Además por definición debe usarse asíncrono para que pueda ser llamado Ajax. Y muy importante debe utilizarse send después de open.

En este ejemplo se va a realizar una conexión a una página llamada index.htm:

```
<script language="JavaScript">
```

```
oXMLHttp = new XMLHttpRequest(); // Se crea el objeto.
```

```
oXMLHttp.open('get', 'index.htm', true); // Se pide la página en modo asíncrono.
```

```
oXMLHttp.send(); //Se envían los datos.
```

```
</script>
```

- El método send es el encargado de enviar la petición con los datos pasados por parámetro como cuerpo de la petición a través del objeto XMLHttpRequest. Debe utilizarse de la siguiente manera:

```
oXMLHttpRequest.send (mData);
```

donde mData es el cuerpo de la petición HTTP y puede ser una referencia al DOM de un documento o una cadena de caracteres.

Es muy importante decir que debe pasarse siempre el parámetro mData aunque no sea requerido en muchos navegadores.

- El método setRequestHeader es el encargado de añadir un encabezado HTTP a la petición HTTP a través del objeto XMLHttpRequest. Cuando se va a utilizar este método debe ponerse así:

```
oXMLHttpRequest.setRequestHeader (sNombre, sValor);
```

donde sNombre no es más que el nombre del encabezado HTTP y sValor es el valor del encabezado HTTP.

Es importante señalar que el parámetro sNombre no podrá contener espacios, puntos o saltos de línea y el parámetro sValor no podrá contener saltos de línea. Además que solo podrá utilizarse setRequestHeader cuando el valor de readyState sea igual a uno.

En el siguiente ejemplo se le añadirá el encabezado User a la petición XMLHttpRequest:

```
<script language="JavaScript">
```

```
oXMLHttp = new XMLHttpRequest(); //Se crea el objeto.
```

```
oXMLHttp.setRequestHeader('User', 'AJAX'); //Se añade el encabezado.
```

```
oXMLHttp.open('get', 'index.htm', true); // Se pide la página en modo asíncrono.
```

```
oXMLHttp.send(); // Se envían los datos.
```

```
</script>
```

2.3.2 Cómo crear el objeto XMLHttpRequest

Después de ver todo lo relacionado con el objeto XMLHttpRequest se prosigue a la creación del objeto XMLHttpRequest. Comenzar diciendo que este objeto está disponible para la mayoría de los navegadores excepto para las versiones 5 y 6 de Internet Explorer por lo que cuando se crea el objeto de comunicación con el servidor se debe tener en cuenta el navegador con el que trabaja el usuario.

Es importante tener en cuenta que algunos usuarios acceden con un navegador sin JavaScript o con una versión pobre del mismo, en caso de que el objeto no pueda ser creado de ninguna de estas dos formas debe indicársele al usuario o mejor aún debe hacerse de forma tradicional es decir sin Ajax.

Este es un ejemplo donde se crea una función para realizar una conexión, esta función usará variables locales, además de que es recomendable incluir todas las funciones que serán usadas en un fichero .js externo e incluirlo en el documento HTML.

```
<script>
function AJAXCrearObjeto()
{
    var obj;
    if(window.XMLHttpRequest) // Si no es Internet Explorer.
    {
        obj = new XMLHttpRequest();
    }
    else // Es Internet Explorer o no tiene el objeto.
    {
        try
```

```
{
  obj = new ActiveXObject("Microsoft.XMLHTTP");
}
catch(e)
{
  alert('El navegador que ha sido utilizado no está soportado');
}
}
return obj;
}
</script>
```

Para llamar a la función AJAXCrearObjeto, es decir para obtener el objeto que se va a utilizar más adelante, se pone así:

```
<script>
oXML = AJAXCrearObjeto();
</script>
```

2.3.3 Cómo realizar una petición con Ajax

Es importante comenzar diciendo que el primer paso para establecer la comunicación con el servidor usando Ajax, es hacer la petición luego el servidor preparará y devolverá que más adelante se verá cómo se recibe, y cómo se incorpora a la página.

Existen dos tipos de conexión con el servidor como se había dicho anteriormente en la referencia al método open, es decir la petición asíncrona y síncrona, pero como Ajax por definición la que utiliza es asíncrona, será la que se utilizará en esta explicación.

Para realizar la petición al servidor se utilizarán los métodos open, onreadystatechange y send los cuales han sido vistos anteriormente, estos métodos sirven para preparar la petición, seleccionar la función de recepción e iniciar la petición respectivamente.

Como se había dicho antes al método open hay que pasarle el método de petición GET y la URL que se enviará al servidor y mediante la cual, el servidor, creará la respuesta que posteriormente será leída.

En este primer ejemplo se va a pedir un documento de texto:

```
<script>  
oXML = AJAXCrearObjeto (); // Se crea el objeto.  
oXML.open( `GET `, `ejemplo.txt`); //Se prepara la petición.  
oXML.onreadystatechange = leerDatos; //Se prepara la recepción.  
oXML.send( ` `); //Se realiza la petición.  
</script>
```

Para que esto funcione hay que haber declarado anteriormente la función leerDatos para tratar los datos recibidos del servidor y mostrarlos al usuario. Esta función trabaja de la forma siguiente lo primero que se hará será comprobar el estado de la petición con el método readyState que puede devolver cualquiera de los siguientes valores: 0 si los datos de la petición no se han definido, 1 (abierto) es decir la recepción de los datos está en curso, 2 (cargado) si la recepción de datos ha finalizado pero los datos no están disponibles, 3 si el objeto aún no está listo para otra petición pero ha recibido ya los datos y 4 (Completado) es decir el objeto está listo para otra petición y luego finalmente estando en el estado de cargado se puede leer el texto recibido usando el método responseText.

Luego para pasar parámetros por GET se usa una URL con parámetros en la función open independientemente de usar GET o POST. Un ejemplo de lo planteado anteriormente es:

```
<script>  
parámetro = 'Datos pasados por GET'; //Se crea la variable parámetro.  
oXML = AJAXCrearObjeto (); // Se crea el objeto.  
oXML.open( `GET`, ` pagina.php?parametro=` + escape(parametro)) ; //Se  
                                                    //prepara la  
                                                    //petición  
                                                    //con  
                                                    //parámetros  
  
oXML.onreadystatechange = leerDatos ; // Se prepara la recepción.  
oXML.send ( ` `) ; //Se realiza la petición.  
</script>
```

Este es otro ejemplo donde debe usarse el método POST en la función open y pasar los parámetros desde la función send.

```
<script>  
parámetro = 'Datos pasados por POST'; // Se crea la variable parámetro.  
oXML = AJAXCrearObjeto () ; // Se crea el objeto.  
oXML.open( `POST`, ` pagina.php` ) ; // Se prepara la petición con parámetros.  
oXML.onreadystatechange = leerDatos ; // Se prepara la recepción.  
oXML.send( 'parametro=' + escape(parametro)); //Se realiza la petición.  
</script>
```

Es muy importante hablar del método escape que devuelve la cadena pasada por parámetro con las transformaciones necesarias para ser enviada en una transacción HTTP. Cuando se utiliza debe ponerse así:

```
sCodificada = escape (sDescodificada);
```

donde sCodificada no es más que la cadena preparada para ser enviada y sDescodificada es la cadena antes de las transformaciones.

En este método todos los caracteres no ASCII ya sean signos de puntuación, espacios y acentos de sDescodificada son sustituidos por su notación %XX, donde XX es su equivalente hexadecimal.

Este es un ejemplo donde se decodifica una cadena no ASCII:

```
<script language="JavaScript">  
  
cadena = "Hola! mundo"; //Se crea la cadena.  
cadena = escape ( cadena ); //Se codifica.  
alert ( cadena ); //Se alertan los datos.  
    //Resultado: "Hola mundo"  
  
</script>
```

2.3.4 Cómo recibir la petición Ajax

Este es un ejemplo donde se recibe la petición realizada en el capítulo anterior donde se había hecho una petición indicando que cuando cambie el estado de la misma, se ejecute la función leerDatos, es decir un ejemplo que une todo lo visto hasta este momento:

```
<script>  
function leerDatos ()  
{  
    if (oXML.readyState == 4)  
    {  
        alert (oXML.responseText);  
    }  
}
```

```
    }  
  }  
function AJAXCrearObjeto()  
{  
  var obj;  
  if(window.XMLHttpRequest) //Si no es IE  
  {  
    obj = new XMLHttpRequest();  
  }  
  else // Si es IE o no tiene el objeto.  
  {  
    try  
    {  
      obj = new ActiveXObject("Microsoft.XMLHTTP");  
    }  
    catch (e)  
    {  
      alert('El navegador utilizado no está soportado');  
    }  
  }  
  return obj;  
}  
oXML = AJAXCrearObjeto();  
oXML.open('GET', 'archivo.txt');  
oXML.onreadystatechange = leerDatos;  
oXML.send("");  
</script>
```

2.3.5 Respuesta Ajax

Como fue planteado anteriormente Ajax utiliza XML para organizar los datos transmitidos entre el servidor y el navegador y para que el navegador sea capaz de interpretar estos datos hay que identificarlos como XML y su contenido un XML bien definido, sino los datos no podrán ser utilizables.

A la hora de recibir los datos lo primero es que el navegador interprete el contenido obtenido y esto es posible si el encabezado es de contexto XML (text/xml), esto es posible con el encabezado HTTP.

Es muy importante adicionar que como la respuesta XML será habitualmente generada de manera dinámica, se recomienda enviar encabezamientos de control de caché para asegurar que la aplicación siempre esté trabajando con los contenidos que solicita y no con una caché almacenada en su navegador.

Cuando se van a mandar estos encabezados de los que se habló anteriormente se deben poner estos códigos al principio de todo documento, si con el lenguaje del lado del servidor que se va realizar es PHP se pone así:

```
<?php  
header("Content-Type: text/xml");  
header("Cache-Control: no-cache, must-revalidate");  
header("Expires: Mon, 13 May 2007 07:00:00 GMT");  
?>
```

con el lenguaje ASP así es como debe ponerse:

```
<%  
response.ContentType="text/xml"  
response.CacheControl="no-cache, must-revalidate"  
response.Expires="Mon, 13 May 2007 07:00:00 GMT"  
%>
```

En cuanto al contenido cuando el navegador lo recibe lo analiza para estructurar los datos recibidos y así poder manipularlos desde la aplicación, para que funcione el contenido del documento debe ser como se había dicho antes XML válido por lo que debe empezar con la declaración de la versión:

```
<?xml version="1.0"?>
```

Se debe tener cuidado con la declaración de XML cuando se trabaja con archivos PHP, porque PHP interpreta <? como inicio de su código cuando tiene las short tags activadas.

Seguidamente se pueden enviar los datos en formato XML por ejemplo:

```
<?xml version="1.0"?>  
<xml>  
  <mensaje>  
    <color> #000000 </color>  
    <texto> Hola </texto>  
  </mensaje>  
</xml>
```

2.3.6 Tratamiento de la respuesta Ajax

Una vez recibida la petición Ajax debe saberse interpretar los datos XML recibidos usando JavaScript y para ello se utiliza responseXML en lugar de responseText entonces es posible comenzar a parsear el documento recibido de la siguiente manera:

```
<script>  
function leerDatos()  
{  
  if (oXML.readyState == 4)  
  {  
    var xml = oXML.responseXML.documentElement;  
  }  
}  
</script>
```

Este es un ejemplo completo que explica todo lo anterior de parsear el documento XML:

```
<html>  
<head>  
<title>ProgramaciónWeb - Ejemplo</title>  
<script>  
function leerDatos() // Esta función fue explicada anteriormente pero no se  
  // había visto como queda elaborada....  
{  
  if (oXML.readyState == 4)
```

```
{
  var xml = oXML.responseXML.documentElement;
  for (i = 0; i < xml.getElementsByTagName('mensaje').length; i++)
  {
    var item = xml.getElementsByTagName('mensaje')[i];
    var txt = item.getElementsByTagName('texto')[0].firstChild.data;
    alert(txt);
  }
}
function AJAXCrearObjeto()
{
  // Como mismo fue explicada anteriormente.....
}
</script>
</head>
</html>
```

En este ejemplo dentro de la función leerDatos () se utiliza la función getElementsByTagName () la cual devuelve un arreglo de todos los elementos con el nombre de las etiquetas indicado, en este ejemplo lo hace con la etiqueta <mensaje>, además con esta función se puede acceder a un elemento determinado poniendo entre corchetes un número, es decir devuelve una lista de todos los subelementos del elemento actual. Cuando se utiliza esta función debe ponerse así:

```
aElementos = getElementsByTagName (sNombre);
```

donde *aElementos* es la cadena de subobjetos del elemento y *sNombre* no es mas que el nombre de los elementos que serán devueltos.

Además es importante decir que para acceder a un texto entre las etiquetas `<texto>` y `</texto>` se utiliza la función `firstChild.data ()`.

2.3.7 Cómo mostrar los datos al usuario

Es necesario mostrarles a los usuarios los datos obtenidos del servidor, esto no debe hacerse como se había visto hasta ahora en los ejemplos anteriores con `alert`, sino que debe hacerse sin refrescar la página es decir en la misma página que el usuario está observando en ese momento, aquí casi siempre se utiliza un objeto `div` al que se le puede modificar su contenido, un ejemplo de lo planteado anteriormente es la siguiente sentencia de código:

```
<div id="miDiv1">Aquí aparecen los datos</div>.
```

Para ver cómo se van a mostrar los datos obtenidos en el `div` creado anteriormente lo primero que se hace es acceder al objeto a través de su `id` (identificador) y esto se realiza con el método `getElementById` y posteriormente se puede utilizar `innerHTML` para indicar el contenido en formato HTML que tendrá este `div` en su interior, como por ejemplo:

```
<script>  
miDiv = document.getElementById('miDiv1'); //Se accede al div  
miDiv.innerHTML = '<b>Este es el nuevo contenido</b>'; //Se modifica su contenido  
</script>
```

La manera de cómo pasar los datos recibidos del servidor al `div` depende de cada caso, pero en este ejemplo hay que imaginarse que se tiene una lista de usuarios en XML.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xml>
  <usuario>
    <id>1</id>
    <nombre>Yanaisy</nombre>
  </usuario>
  <usuario>
    <id>2</id>
    <nombre>Yenivet</nombre>
  </usuario>
  <usuario>
    <id>3</id>
    <nombre>Maykel</nombre>
  </usuario>
</xml>
```

Luego se supone que se han pedido estos datos al servidor y que se ha indicado como readystatechange handler la función leerDatos, ahora hay que centrarse en el contenido de dicha función para que muestre una lista de usuarios con enlace a su perfil en el div que se ha creado con id miDiv1 y el ejemplo de esto es lo siguiente:

```
<script>
function leerDatos() //Se reciben y muestran los datos
{
  if (oXML.readyState == 4) //Se comprueba que los datos se han recibido.
  {
    var xml =oXML.responseXML.documentElement;//Se accede
                                                // al XML recibido
```

```
var miDiv = document.getElementById('miDiv1'); //Se accede al Div
miDiv.innerHTML = ""; //Se vacía el Div
for (i = 0; i < xml.getElementsByTagName('usuario').length; i++)//Se
    //itera
    // cadausuario
{
var item = xml.getElementsByTagName('usuario')[i]; //Se accede
    //al objeto XML usuario
var id = item.getElementsByTagName('id')[0].firstChild.data; //Se
    //recoje el id
var nombre = item.getElementsByTagName('nombre')[0].firstChild.data;
    //Se
    // recoge el nombre
miDiv.innerHTML += '<a href="/perfil/'+id+'/">'+nombre+'</a><br>';
    //Se
    //muestra el enlace
}
}
}
</script>
```

2.3.8 Implementaciones de Ajax

Existen diversidades de implementaciones Ajax que facilitan el desarrollo de aplicaciones con comunicación con el servidor por ejemplo en las “Librerías de Cliente” que son las que permiten trabajar con mayor facilidad cuando se realizan llamadas al servidor y la principal ventaja de estas librerías es que no dependen de un lenguaje de servidor se encuentra prototype que es

una interesante librería de JavaScript que permite el uso de las aplicaciones web dinámicas, ofreciendo una caja de herramientas única, es fácil de utilizar y además tiene a su alrededor la librería más grande de Ajax. Otro ejemplo son las “Librerías de cliente/servidor” que son las que permiten hacer llamadas a funciones del servidor desde el cliente usando las funciones JavaScript autogeneradas que son llamadas proxies, dentro de esta clasificación se encuentran Ajax.Net Professional más conocido por AjaxPro que permite crear un proxy para llamar a métodos .Net del servidor desde JavaScript, xajax que es un framework escrito en PHP de código abierto y permite crear aplicaciones web que utilizan Ajax, XOAD que es un proxy de PHP que usa XMLHttpRequest y JSON el cual es un formato de datos muy ligero basado en un subconjunto de sintaxis de JavaScript.

Conclusiones

En este capítulo se trata lo relacionado a cuándo debe usarse Ajax, cuando no usar esta técnica, además se elaboró un manual para que los usuarios pueda utilizarlo para el desarrollo de aplicaciones web.

CONCLUSIONES

El desarrollo de este trabajo investigativo permitió dar cumplimiento a todos los objetivos que fueron trazados desde el inicio de este documento. Se realizó un estudio de una nueva tecnología nombrada Ajax, en este estudio se define la importancia y necesidad del uso de esta técnica para el perfeccionamiento de aplicaciones web ya que esta presenta un intermediario entre el usuario y el servidor el cual es conocido como motor Ajax que permite que las aplicaciones elaboradas con esta tecnología sean cada día más eficientes, es decir mejora las aplicaciones posibilitando que los usuarios nunca tengan que esperar una petición mirando una página en blanco.

También tiene numerosas ventajas las cuales hacen que cada día esta técnica sea más utilizada a nivel mundial ya que Ajax es la combinación de muchas tecnologías ya existentes que se unen para formar una nueva era en el mundo de la web. Dentro de estas técnicas que Ajax incluye tenemos al HTML, CSS y XML para la presentación de información, el DOM y JavaScript para el intercambio directo con la información y XMLHttpRequest que es el protocolo sobre el que está apoyado. También es compatible con lenguajes dinámicos de programación como ASP, PHP, ASP.Net, Perl, los cuales hacen que sea mucho mejor y más fácil de utilizar.

Además se llegó a conclusiones con respecto a las desventajas que trae consigo la utilización de esta técnica así como las preferencias de su uso. Y finalmente se realizó la elaboración de un manual el cual brinda la posibilidad de que los desarrolladores o usuarios que tiene poco conocimiento de esta técnica y quieran crear aplicaciones web puedan consultarlo para mejorar sus conocimientos y además poder crear aplicaciones con más calidad y mucho más eficientes.

RECOMENDACIONES

- Realizar pruebas prácticas de los elementos teóricos que se exponen en esta explicación.
- Seguir adelante con esta investigación para poder cada día más aprender más.
- Utilizar este documento como fuente de consulta y apoyo.

REFERENCIAS BIBLIOGRÁFICAS

1. GARRET, J. J. *Ajax: Un nuevo acercamiento a las aplicaciones web*. 2005, consultado: 15/12/06, Disponible en: http://gpi.uci.cu/index.php?option=com_remository&Itemid=31&func=select&id=339.
2. QUIJADO, J. L. *DOMINE HTML Y DHTML*. 2003, consultado: 27/1/2007, Disponible en: <http://www.agapea.com/Domine-HTML-y-DHTML--n11712i.htm>.
3. ALVAREZ, M. A. *Programación en JavaScript*. Consultado: 3/2/2007, Disponible en: <http://www.desarrolloweb.com/articulos/490.php>.
4. IDEM 3
5. IDEM 1
6. IDEM 1
7. CEBRIAN, M. *Puntos a tener en cuenta cuando uses Ajax*. 2006, consultado: 12/1/2007, Disponible en: <http://www.ingeniuz.com/2006/05/08/puntos-a-tener-en-cuenta-cuando-uses-ajax>.
8. ANÓNIMO. *El framework más popular de Ajax*. 2007, consultado: 20/3/2007, Disponible en: <http://www.tufuncion.com/mejor-framework-ajax>.
9. ---. *Places To Use Ajax*. Consultado: 12/6/2007, Disponible en: <http://swik.net/Ajax/Places+To+Use+Ajax>

BIBLIOGRAFÍA

1. ALFONSO, X. C. G. Y. J. M. *Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto*. publicado el: 07/03/2007 de 2004, última actualización: 07/03/2007. Disponible en: <http://mosaic.uoc.edu/articulos/cms1204.html>.
2. ALVAREZ, M. A. *Que es DHTML o HTML Dinámico*. publicado el: 27/01/2007 de última actualización: 27/01/2007. Disponible en: <http://www.desarrolloweb.com/articulos/391.php>.
3. ANÓNIMO. *Qué es Ajax?* publicado el: 15/12/2006 de última actualización: 15/12/2006. Disponible en: http://www.mctekk.com/index.php/seccion/ajax/que_es_ajax/321.htm.
4. ---. *Puntos a tener en cuenta cuando uses Ajax*. publicado el: 12/01/2007 de última actualización: 12/01/2007. Disponible en: <http://www.ingeniuz.com/2006/05/08/puntos-a-tener-en-cuenta-cuando-uses-ajax>.
5. ---. *Google Suggest*. publicado el: 06/03/2007 de última actualización: 06/03/2007. Disponible en: <http://google.dirson.com/post/966/>.
6. ---. *Google Groups - los grupos de discusión de Google*. publicado el: 15/03/2007 de última actualización: 15/03/2007. Disponible en: <http://google.dirson.com/o.a/google-groups/>.
7. ---. *Ajax acelera el desarrollo de interfaces web en las aplicaciones interactivas. Representa una amplia colección de las últimas tendencias en tecnologías de Internet*. publicado el: 10/02/2007 de última actualización: 10/02/2007. Disponible en: <http://www.idg.es/computerworld/articulo.asp?id=179064>.
8. ---. *Ajax*. publicado el: 15/02/2007 de última actualización: 15/02/2007. Disponible en: <http://www.ajax.com.es/index.php?title=Portada>.

Bibliografías

9. ---. *Charla general, opiniones, debates, otros CMS*. publicado el: 24/03/2007 de última actualización: 24/03/2007. Disponible en: <http://drupalmania.com/forums/charla-general-opiniones-debates-otros-cms-0>.
10. ---. *CMS: Sistema de gestión de contenido*. publicado el: 25/03/2007 de última actualización: 25/03/2007. Disponible en: <http://www.ubuntu-es.org/wiki/doku.php/web:cms>.
11. ---. *Tutorial de JavaScript*. publicado el: 03/05/2007 de última actualización: 03/05/2007. Disponible en: <http://www.unav.es/cti/manuales/TutorialJavaScript/indices/>.
12. ---. *La web del programador* publicado el: 27/04/2007 de última actualización: 27/04/2007. Disponible en: <http://www.lawebdelprogramador.com/cursos/enlace.php?idp=3298&id=45&texto=JavaScript>.
13. ---. *El universo de JavaScript en tus páginas*. publicado el: 05/05/2007 de última actualización: 05/05/2007. Disponible en: <http://www.terra.com.ar/canales/informesespeciales/11/11823.html>.
14. ---. *Lenguajes Dinámicos en .NET*. publicado el: 30/04/2007 de última actualización: 30/04/2007. Disponible en: <http://blogs.msdn.com/ricardoj/archive/2006/10/25/lenguajes-dinamicos-en-net.aspx>.
15. ---. *DHTML*. publicado el: 04/02/2007 de última actualización: 04/02/2007. Disponible en: <http://www.utem.cl/web/dhtml.htm>.
16. ---. *Que es XML*. publicado el: 26/02/2007 de última actualización: 26/02/2007. Disponible en: <http://www.desarrolloweb.com/articulos/449.php>.
17. ---. *AJAX: Creating Web Pages with Asynchronous JavaScript and XML (Bruce Perens Open Source)*. publicado el: 10/05/2007 de última actualización: 10/05/2007. Disponible en: <http://www.flazx.com/ebook5945.php>.

Bibliografías

18. ---. *XML en 10 puntos*. publicado el: 23/02/2007 de 2000, última actualización: 23/02/2007. Disponible en: <http://www.sidar.org/recur/desdi/traduc/es/xml/xml10p/xml10p.htm>.
19. ---. *Drupal*. publicado el: 23/03/2007 de 2003, última actualización: 23/03/2007. Disponible en: <http://libertonia.escomposlinux.org/story/2003/4/10/135935/144>.
20. ---. *Evolución de la web*. publicado el: 20/01/2007 de 2005, última actualización: 20/01/2007. Disponible en: <http://www.adrformacion.com/cursos/ajax/leccion2/tutorial1.html>.
21. ---. *Google Suggest Beta*. publicado el: 10/03/2007 de 2006, última actualización: 10/03/2007. Disponible en: <http://www-dejavu.blogspot.com/2006/01/google-suggest-beta.html>.
22. ---. *Las mejores aplicaciones Ajax*. publicado el: 28/01/2007 de 2006, última actualización: 28/01/2007. Disponible en: <http://www.euroresidentes.com/Blogs/internet/2006/04/las-mejores-aplicaciones-ajax.html>.
23. ---. *Buscador con Ajax y moo.fx en un plugin para WordPress 2006*. publicado el: 17/03/2007 de 2006, última actualización: 17/03/2007. Disponible en: <http://sigt.net/archivo/buscador-con-ajax-y-moofx-en-un-plugin-para-wordpress.xhtml>.
24. ---. *Ejemplos reales donde usar Ajax*. publicado el: 03/04/2007 de 2006, última actualización: 03/04/2007. Disponible en: <http://www.anieto2k.com/2006/08/17/5-ejemplos-reales-de-donde-si-usar-ajax/>.
25. ---. *AJAX, la palabra de moda*. publicado el: 15/02/2007 de 2006, última actualización: 15/02/2007. Disponible en: <http://asp.net.do/foros/ShowPost.aspx?PostID=3127>
26. ---. *IBM AJAX Toolkit Framework*. publicado el: 11/03/2007 de 2006, última actualización: 11/03/2007. Disponible en: <http://www.programacion.net/noticia/1467/>

Bibliografías

27. ---. *Frameworks para Ajax*. publicado el: 17/03/2007 de 2006, última actualización: 17/03/2007. Disponible en: <http://despuesdegoogle.com/2006/09/25/frameworks-para-ajax/>.
28. ---. *Nuevo Google Groups sale de beta*. publicado el: 23/03/2007 de 2007, última actualización: 23/03/2007. Disponible en: http://tecnicalia.com/2007/01/25/tec_nuevo-google-groups-sale-de-beta/.
29. ---. *Curso de Ajax en programación web*. publicado el: 23/04/2007 de 2007, última actualización: 23/04/2007. Disponible en: <http://www.programacionweb.net/cursos/curso.php?num=2>.
30. ---. *Infoxicación*. publicado el: 26/03/2007 de 2007, última actualización: 26/03/2007. Disponible en: <http://www.infoxicacion.cl/blog/category/cms/>.
31. ARANGO, R. Z. *Ajax ¿Moda pasajera o el verdadero futuro?* publicado el: 20/01/2007 de 2005, última actualización: 20/01/2007. Disponible en: <http://www.metodus.com/articulos/ajax/index.jsp>.
32. ARCADI. *Cómo funciona 'Google Suggest'?* publicado el: 03/03/2007 de 2004, última actualización: 03/03/2007. Disponible en: <http://google.dirson.com/post/0970/>.
33. ARENAS, M. I. G. *Curso XML* publicado el: 07/02/2007 de última actualización: 07/02/2007. Disponible en: <http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml>.
34. CUERVOS, J. J. M. *Introducción a los sistemas de gestión de contenidos*. publicado el: 20/03/2007 de 2005, última actualización: 20/03/2007. Disponible en: <http://geneura.ugr.es/~jmerelo/tutoriales/cms/>.

Bibliografías

35. DAVE CRANE, E. P., DARREM JAMES. *Ajax in action*. publicado el: 11/05/07 de 2006, última actualización: 11/05/07. Disponible en: http://gpi.uci.cu/index.php?option=com_remository&Itemid=31&func=select&id=339.
36. FAEL. *Introducción a Ajax con PHP y formularios*. publicado el: 27/04/2007 de 2006, última actualización: 27/04/2007. Disponible en: <http://www.cristalab.com/tutoriales/165/introduccion-a-ajax-con-php-y-formularios>.
37. GARRET, J. J. *Ajax: Un nuevo acercamiento a las Aplicaciones Web*. publicado el: 15/12/2006 de 2005, última actualización: 15/12/2006. Disponible en: http://gpi.uci.cu/index.php?option=com_remository&Itemid=31&func=select&id=339
38. JUSTIN GEHTLAND, B. G., DION ALMER. *Pragmatic Bookshelf*. publicado el: 27/04/07 de 2005, última actualización: 27/04/07. Disponible en: Pragmating.
39. LÓPEZ, G. R. *HTML Dinámico*. publicado el: 31/01/2007 de 2007, última actualización: 31/01/2007. Disponible en: <http://geneura.ugr.es/~gustavo/css/css.html#introduccion>.
40. MORNACCO, D. H. *Ajax*. publicado el: 10/04/2007 de 2003, última actualización: 10/04/2007. Disponible en: <http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=AJAX&PHPSESSID=3111051a90f62943d3a316c19e88ae82>.
41. NICHOLAS C. ZACAS, J. M., JOE FAWCETT. *Professional Ajax, 2nd Edition (Programmer to Programmer)* publicado el: 06/05/2007 de última actualización: 06/05/2007. Disponible en: <http://www.flazx.com/ebook7424.php>.
42. QUIJADO, J. L. *DOMINE HTML Y DHTML*. publicado el: 05/02/2007 de última actualización: 05/02/2007. Disponible en: <http://www.agapea.com/Domine-HTML-y-DHTML--n11712i.htm>.

Bibliografías

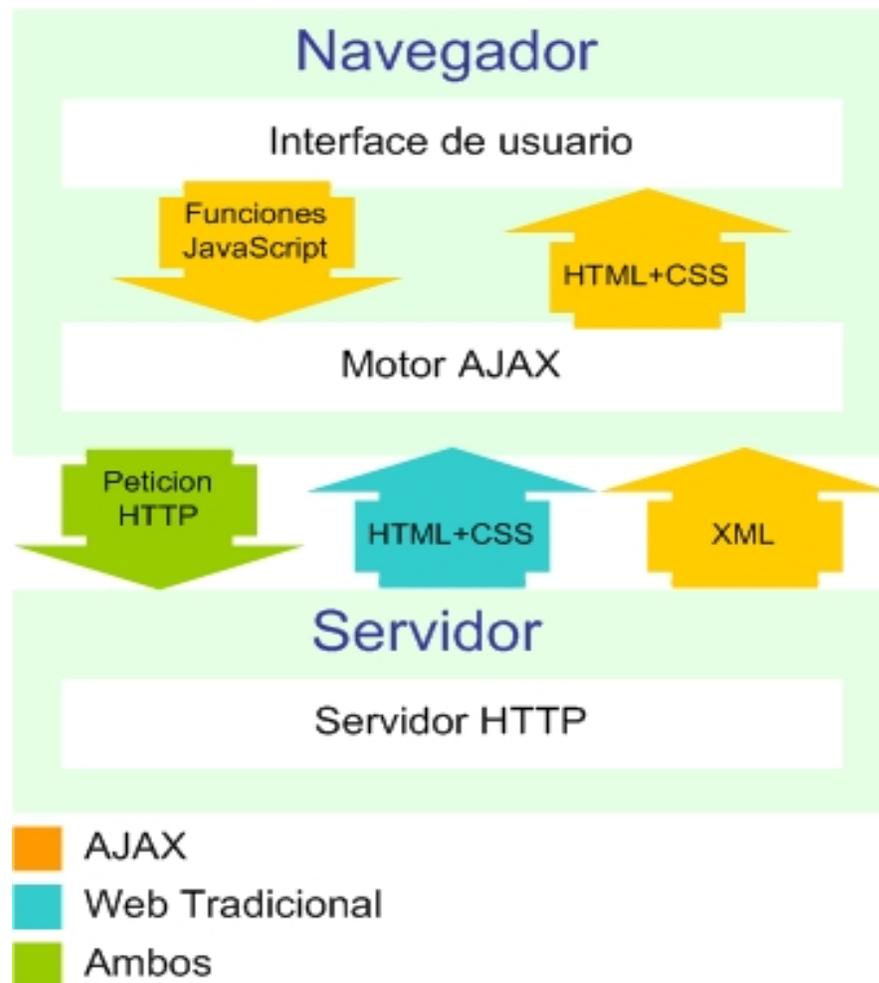
43. REYERO, J. A. *¿Por qué Drupal?* publicado el: 23/03/2007 de 2006, última actualización: 23/03/2007. Disponible en: <http://www.reyero.net/es/node/102>.

44. RYAN ASLESON, N. T. S. *Foundations of Ajax*. publicado el: 25/04/07 de 2006, última actualización: 25/04/07. Disponible en: http://gpi.uci.cu/index.php?option=com_remository&Itemid=31&func=select&id=339.

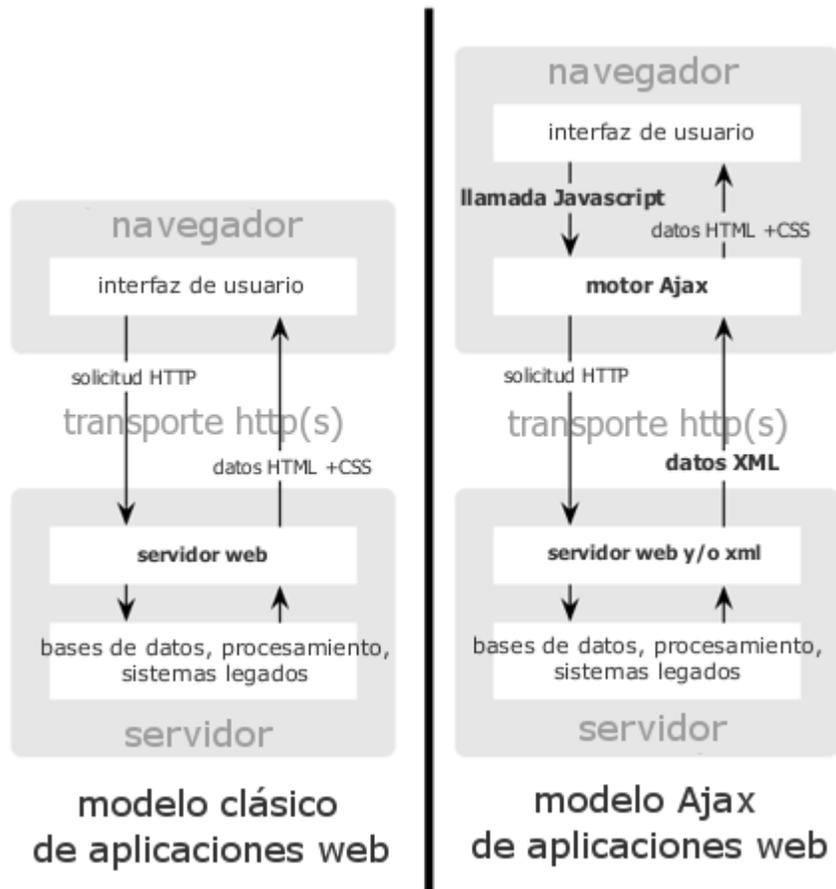
45. VALENTE, M. *Efectos DHTML para Msle 1/5*. publicado el: 04/02/2007 de última actualización: 04/02/2007. Disponible en: <http://www.htmlpoint.com/dhtml/12/index.html>.

ANEXOS

Anexo 1 “Funcionamiento de Ajax”

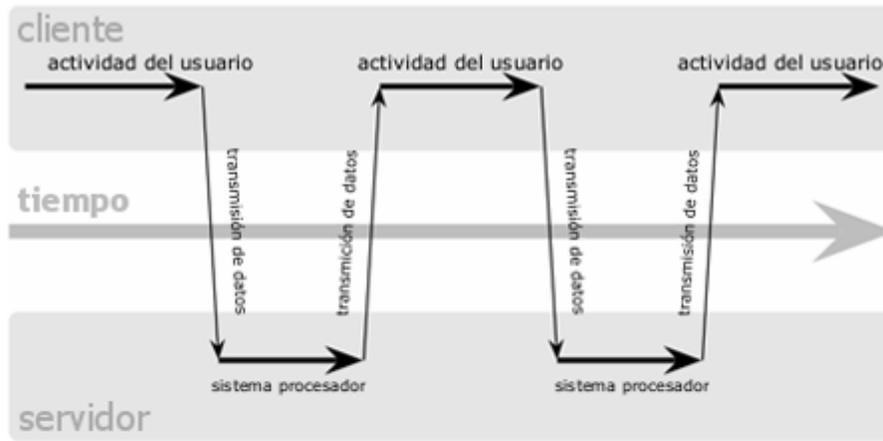


Anexo 2: Modelo tradicional para las aplicaciones Web (izquierda) comparado con el modelo de Ajax (derecha).

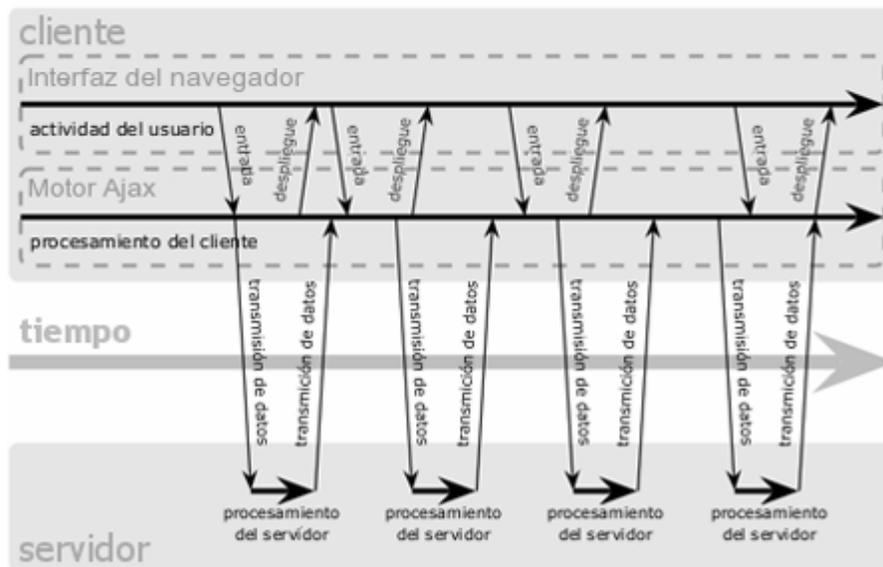


Anexo 3: Patrón de interacción sincrónica de una aplicación Web tradicional (arriba) comparada con el patrón asincrónico de una aplicación AJAX (abajo).

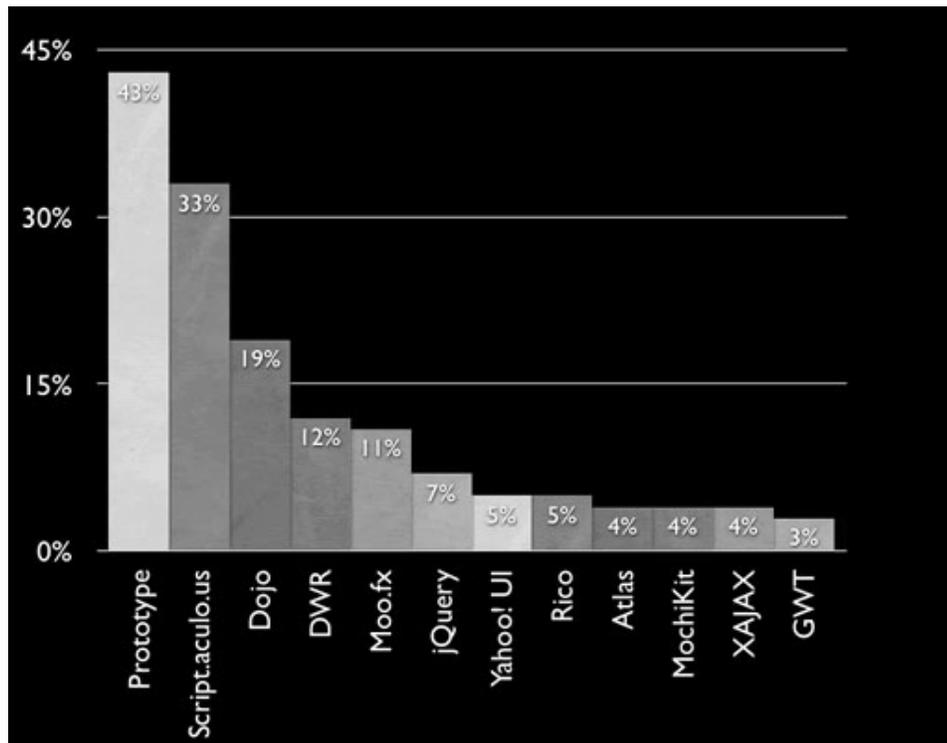
modelo clásico de aplicaciones web (síncrono)



modelo Ajax de aplicaciones web (asíncrono)



Anexo 4: Gráfico que muestra una comparación entre los frameworks más populares de Ajax.



Anexo 5: Lista con los posibles códigos y sus significados, es decir el estado en que se encuentra:

- 100 - Continúa.
- 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.

Anexos

- 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 disponibles.
- 411 - Requiere longitud.
- 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.

Anexo 6: Este es un ejemplo en el cual se adiciona un proyecto a una lista de proyectos ya existentes:

```
[Ajax.AjaxMethod()]
public void AddProyecto(string[] list)
{
    Proyecto p = new Proyecto(list[0], DateTime.Parse(list[1]),
DateTime.Parse(list[2]),
                int.Parse(list[3]), list[4]);
    Horario h = new Horario(DateTime.Parse(list[5]), DateTime.Parse(list[6]));
    Moment_Dia md = new Moment_Dia(list[7]);
    bool soft;
    string comp;
    Lenguaje l = new Lenguaje();
    if (list[8] == ".NET")
    {
        soft = false;
        comp = "Microsoft";
        l.nombre = "C#";
    }
    else
        if (list[8] == "Java")
        {
            soft = true;
            comp = "Sun";
            l.nombre = "Java";
        }
}
```

```
    }  
    else  
    {  
        soft = true;  
        comp = "Linux";  
        l.nombre = "Python";  
    }  
    Plataforma pl = new Plataforma (list[8], comp, soft);  
  
    gestion.InsertProyecto(p, h, md, pl, l);  
}
```

Anexo 7: Este es un ejemplo donde se selecciona la Compañía a la cual pertenece la plataforma de desarrollo especificada.

```
[Ajax.AjaxMethod()]  
public System.Web.UI.HtmlControls.HtmlSelect Compannia(string plat)  
{  
    System.Web.UI.HtmlControls.HtmlSelect control = new  
    System.Web.UI.HtmlControls.HtmlSelect();  
  
    switch (plat)  
    {  
        case ".NET":  
            ListItem l = new ListItem("Microsoft", "Microsoft");
```

```
        control.Items.Add(l);
        break;

    case "Java":
        ListItem l1 = new ListItem("Sun", "Sun");
        control.Items.Add(l1);
        break;

    case "Python":
        ListItem l2 = new ListItem("Linux", "Linux");
        control.Items.Add(l2);
        break;

    case "":
        throw new System.Security.SecurityException("Seleccione una
Plataforma");
    }

    return control;
}
```

Anexo 8: Este es un ejemplo donde se muestran los proyectos existentes en una lista de proyectos.

```
[Ajax.AjaxMethod()]
public System.Web.UI.HtmlControls.HtmlSelect ListarProyectos()
{
```

```
        System.Web.UI.HtmlControls.HtmlSelect control = new
System.Web.UI.HtmlControls.HtmlSelect();

        List<Proyecto> list = gestion.ListadoProyectos();

        control.Items.Add("Proyectos");

        foreach (Proyecto var in list)
        {
            ListItem l = new ListItem(var.nombre, var.nombre);
            control.Items.Add(l);
        }

        return control;
    }
```

Anexo 9: Muestra un listado con el nombre de las personas que pertenecen a un proyecto dado.

```
public System.Web.UI.HtmlControls.HtmlSelect ListarPersPro(string pro)
{
    System.Web.UI.HtmlControls.HtmlSelect control =
        new
System.Web.UI.HtmlControls.HtmlSelect();
    List<Persona> list = gestion.listarPersonasProyecto(pro);
    control.Items.Add("Personal");
    foreach (Persona var in list)
    {
        control.Items.Add(var.nombre);
    }
}
```

```
    }  
  
    return control;  
}
```

Anexo 10: Devuelve un texto en código HTML, el cual contiene nombre, apellido, correo electrónico y un comentario, pasados por parámetros.

[Ajax.AjaxMethod]

```
public static string Test1(string firstName, string familyName, string email, string  
comment)  
{  
    return "Hello " + firstName + " " + familyName + "<br>Thank you for your  
comment (" + DateTime.Now.ToString() + "):<br><b>" +  
System.Web.HttpUtility.HtmlEncode(comment).Replace("\r", "<br>") + "</b>.";  
}
```

Anexo 11: Devuelve el día posterior a la fecha pasada por parámetros.

[Ajax.AjaxMethod]

```
public string Test2(DateTime d)  
{  
    d = d.AddDays(1);  
  
    return "The next day will be " + d.ToLongDateString() + ".";  
}
```

Anexo 12: En este ejemplo, siempre que se invoque devuelve un error.

```
[Ajax.AjaxMethod]  
public void Test4()  
{  
    throw new System.Security.SecurityException("Este Usuario no tiene acceso a  
este  
recurso");  
}
```

Anexo 13: Devuelve el valor siguiente almacenado en la variable aleatoria "r".

```
[Ajax.AjaxMethod]  
public int Test8()  
{  
    Random r = new Random(System.DateTime.Now.Second);  
    return r.Next(0, 100);  
}
```

Anexo 14: Devuelve una colección de elementos de tipo cadena, utilizando las librerías del sistema, en este caso del tipo StringCollection.

```
[Ajax.AjaxMethod]  
public System.Collections.Specialized.StringCollection Test9()  
{  
    System.Collections.Specialized.StringCollection s = new  
System.Collections.Specialized.StringCollection();  
  
    s.Add("Yenivel");  
    s.Add("Yanaisy");  
}
```

```
    return s;  
}
```

Anexo 15: Devuelve una colección de objetos, utilizando las librerías del sistema.

```
[Ajax.AjaxMethod]  
public object[] Test10()  
{  
    object[] o = new object[3];  
  
    o[0] = "Cesar";  
    o[1] = DateTime.Now;  
    o[2] = true;  
  
    return o;  
}
```

Anexo 16: Devuelve una colección de elementos, utilizando las librerías del sistema, en este caso del tipo ArrayList.

```
[Ajax.AjaxMethod]  
public System.Collections.ArrayList Test11()  
{  
    System.Collections.ArrayList a = new System.Collections.ArrayList();  
  
    a.Add("Michael");  
    a.Add(DateTime.Now);  
    a.Add(true);  
}
```

```
Person p1 = new Person();
p1.FirstName = "Cesar";

SpecialPerson p2 = new SpecialPerson();           //Clase que hereda de
Person.
p2.FirstName = "Alejandro";
p2.Comment = "Comentario de esta Persona!";

a.Add(p1);
a.Add(p2);

return a;
}
```

Anexo 17: Devuelve todos los datos de una persona.

[Ajax.AjaxMethod]

```
public Person Test12()
{
    Person p = new Person();
    p.FirstName = "Alejandro";
    p.FamilyName = "Martinez";
    p.Age = 28;
    Person mj = p.NewChild();
    mj.FirstName = "Marcos Julian";
    mj.Age = 3;
    Person jo = p.NewChild();
    jo.FirstName = "Jean Oliver";
}
```

```
jo.Age = 1;
p.Children = new Person[2];
p.Children[0] = mj;
p.Children[1] = jo;
//Devuelve la persona creada con todos sus datos.
return p;
}
```

Anexo 18: Devuelve los elementos encontrados que se derivan de la consulta realizada a la base de datos.

[Ajax.AjaxMethod]

```
public System.Data.DataSet Test15()
```

```
{
```

```
    System.Data.SqlClient.SqlConnection conn = new
```

```
System.Data.SqlClient.SqlConnection("server=(local);uid=sa;password=*;Initial
    Catalog=Northwind;");
```

```
    System.Data.SqlClient.SqlCommand cmd = new
```

```
System.Data.SqlClient.SqlCommand("SELECT TOP 10 * FROM Customers C INNER
    JOIN Orders O ON O.CustomerID = C.CustomerID",
```

```
conn);
```

```
    System.Data.SqlClient.SqlDataAdapter da = new
```

```
System.Data.SqlClient.SqlDataAdapter(cmd);
```

```
System.Data.DataSet ds = new System.Data.DataSet();
try
{
    conn.Open();
    try
    {
        da.Fill(ds);
    }
    catch (Exception)
    {}
    finally
    {
        conn.Close();
        conn.Dispose();
    }
}
catch (Exception)
{}

    return ds;
}
```

Anexo 19: Devuelve la suma de los valores contenidos en un arreglo dado de valores enteros.

```
[Ajax.AjaxMethod]
public int Test17(int[] i)
{
    int r = 0;
    foreach (int ii in i)
        r += ii;
}
```

Anexos

```
    return r;  
}
```

Anexo 20: Concatena los valores contenidos en el arreglo dado, codificado en código HTML.

```
[Ajax.AjaxMethod]  
public string Test18(string[] s)  
{  
    string r = "";  
    foreach (string ss in s)  
        r += "<p>" + ss + "</p>\r\n";  
  
    return r;  
}
```

Anexo 21: Devuelve la fecha actual pasado unos 30 segundos aproximadamente.

```
[Ajax.AjaxMethod(30)]  
public System.DateTime Test20()  
{  
    return DateTime.Now;  
}
```

GLOSARIO DE TÉRMINOS

1. Apache: Es un software libre de código abierto para plataformas UNIX.
2. API (Application Programming Interfaces): Interfaz de Programación de Aplicaciones, es del conjunto de funciones y procedimientos o métodos si se refiere a programación orientada a objetos que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.
3. ASP: Páginas de servidor activas, es una tecnología para hacer scripts del lado del servidor.
4. C: Lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix, es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.
5. C++: Lenguaje de programación, que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.
6. CGI (Common Gateway Interface): No es más que el cambio de la forma de manipular información en la web, es utilizado comúnmente para contadores, bases de datos, motores de búsqueda, formularios y generadores de email automático.
7. Chilisoft: Compañía que ofrece la tecnología ASP a través de Chilisoft ASP, es el mismo funcionamiento que el ASP de Microsoft pero corre en sistemas operativos como Unix, Solaris entre otros.
8. CMS: Sistemas de gestión de contenidos.

9. CSS: Hojas de Estilo en Cascada.
10. DB2: Marca comercial, propiedad de IBM, bajo la cual se comercializa el sistema de gestión de base de datos.
11. DHTML: El HTML Dinámico es un conjunto de técnicas que posibilitan la creación de Sitios Web interactivos utilizando una combinación entre el lenguaje HTML estático, Java Script y CSS.
12. DOM (Document Object Model): Modelo de Objeto del Documento.
13. ECMA-262: Lenguaje de Script estándar de la Oficina de Estándares de la Unión Europea.
14. Flickr: Sistema de gestión de fotografías digitales
15. Frameworks: Estructura de soporte, en la cual cualquier otro proyecto puede ser desarrollado y organizado.
16. FTP (File Transfer Protocol): Protocolo de transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor.
17. GTK (GIMP Toolkit): Es una biblioteca para crear interfaces gráficas de usuario.
18. Handler: Es un manejador o puntero que hace referencia a algo determinado.
19. HTML: Lenguaje de marcas para la creación de hipertextos.
20. HTTP (HyperText Transfer Protocol): Protocolo usado en cada transacción de la Web, el hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido.

21. IBM (International Business Machines): Empresa que fabrica y comercializa hardware, software, y servicios relacionados con la informática.
22. IIS: Servidor de Información de Internet.
23. InstantAsp: Tecnología que permite ejecutar ASP fuera de un sistema operativo Windows.
24. ISAPI: Interfaz de programación de aplicaciones de servidor para Internet. Una interfaz para el desarrollo de aplicaciones de servidor Web, desarrollado por Process Software y Microsoft Corporation, que se utiliza en lugar de CGI.
25. Java: Lenguaje de programación orientado a objetos, toma mucha de su sintaxis del lenguaje C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros.
26. JavaScript: Lenguaje de programación del lado del cliente utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web.
27. JSON: Formato de datos ligero basado en un subconjunto de sintaxis de JavaScript.
28. LAMP: Es la plataforma más común para servicios web, es decir se refiere a un conjunto de subsistemas software necesarios para alcanzar una solución global, en este caso configurar sitios web o servidores dinámicos con un esfuerzo reducido.
29. LIVECONNECT: Tecnología de Netscape que permite comunicar códigos de JavaScript y Java.
30. MacOSX: Sistema Operativo de la familia UNIX.
31. Metalenguaje: Lenguaje utilizado para describir un sistema de lenguaje de programación.

32. MicrosoftSQLServer: Sistema de gestión de base de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.
33. Multitarea: Cuando varios procesos son ejecutados al mismo tiempo compartiendo uno o más procesadores.
34. Multiusuario: Cuando es apto para ser utilizado por varios usuarios.
35. MySQL: Sistema de gestión de Base de Datos multiusuario y es la base de datos de código fuente abierto más usada del mundo.
36. ODBC: Estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation.
37. Oracle: Sistema de gestión de base de datos relacional.
38. Página Web: Fuente de información adaptada para la World Wide Web y accesible mediante un navegador de Internet.
39. PDF (Portable Document Format): Formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.
40. Perl (Practical Extraction and Report Language): Es un lenguaje de scripts que corre en la mayoría de los principales sistemas disponibles.
41. PHP: Es un lenguaje de programación interpretado y multiplataforma muy usado para la creación de aplicaciones para servidores y contenido dinámico para sitios Web.
42. PHP-GTK: es una extensión para el lenguaje de programación PHP que permite la utilización de GTK+. Provee una interface orientada a objetos a las clases y funciones de

GTK+ permitiendo simplificar la creación de programas multiplataformas con interfaz gráfica.

43. Plugins: Aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.
44. Portable: La portabilidad de un software en cuán dependiente es de la plataforma en la que corre, la portabilidad es mayor cuando la dependencia con el software de plataforma disminuye.
45. Postgres: Es un motor de base de datos, es servidor de base de datos relacional libre.
46. PWS: Servidor de Web Personal.
47. RSS: Formato de noticias usado para describir el contenido de páginas Web.
48. Sistema Operativo: Es un administrador de programas y contiene archivos que utilizan estos para poder funcionar. Un sistema operativo es imprescindible en una computadora personal, hace de intermediario entre el usuario y los componentes de la PC.
49. Software Libre: es el software que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.
50. TIC: Tecnologías de la Información y las Comunicaciones.
51. UNIX: Sistema Operativo portable, multitarea y multiusuario.
52. World Wide Web (WWW): Sistema de navegador web para extraer elementos de información llamados "documentos" o "páginas web".
53. XHTML (eXtensible Hypertext Markup Language): Lenguaje extensible de marcado de hipertexto, pensado para sustituir a HTML como estándar para las páginas web.

54. XML: Lenguaje de anotación extensible.

55. XMLHttpRequest: es un API que puede invocarse desde JavaScript, y otros lenguajes de script incluidos en un navegador web, es una parte fundamental de la técnica de desarrollo web conocida como Ajax.