



Departamento de Lenguajes y  
Sistemas Informáticos



Universitat d'Alacant  
Universidad de Alicante

# AJAX

Programación en Internet  
Curso 2009-2010

Programación en Internet – Curso 2009-2010

## Índice

- Introducción
- Definición y conceptos
- Como funciona
- Ventajas y desventajas
- El objeto XMLHttpRequest
- Ciclo de vida de una transferencia con XMLHttpRequest
- Ejemplo
- Bibliografía
- Ejercicios

Programación en Internet – Curso 2009-2010

## Introducción

- Arquitectura cliente-servidor
  - El cliente solicita un servicio y el servidor lo provee.
  - La parte de cliente solo tiene la lógica de presentación y en algunos casos un mínimo de lógica de negocio.
- Aplicaciones en entorno web:
  - Cada interacción del usuario genera una transacción contra el servidor.
  - En algunos casos, según las características de la aplicación, el interface, etc. una pantalla HTML puede generar demasiadas peticiones.
  - Problemas:
    - Esperas del cliente
    - Errores en la transmisión

Programación en Internet – Curso 2009-2010

## Introducción

- ¿Qué aporta AJAX?
  - Interacción diferente con el servidor:
    - Anticipo de transferencias con el servidor para disponer de los datos más rápidamente. (interacción en background)
    - Permitir al usuario poder hacer otras cosas en el cliente mientras una transferencia está en curso (interacción asíncrona)
  - Nueva filosofía:
    - La aplicación web se apoya más en el cliente, al cual se le dota de más lógica de negocio.
    - Se consigue una interacción en el cliente similar a las aplicaciones autónomas.
- ¿Y el XML?
  - Es el formato en el que se transmiten los datos el servidor y el cliente.
- Usado en: Google Maps, Gmail, etc.

## Definición y conceptos

- *Asynchronous Javascript And XML*
  - XML y JavaScript asíncronos
- No es una nueva tecnología,
  - sino un conjunto de tecnologías, el uso de las cuales permite una nueva filosofía de desarrollo de aplicaciones web.
- Combinación de HTML/XHTML, CSS, Javascript, DOM, XML, XSLT, HTTP.
  - Todas ellas son estándares abiertos, tecnologías implantadas, difundidas y conocidas en Internet.

## Definición y conceptos

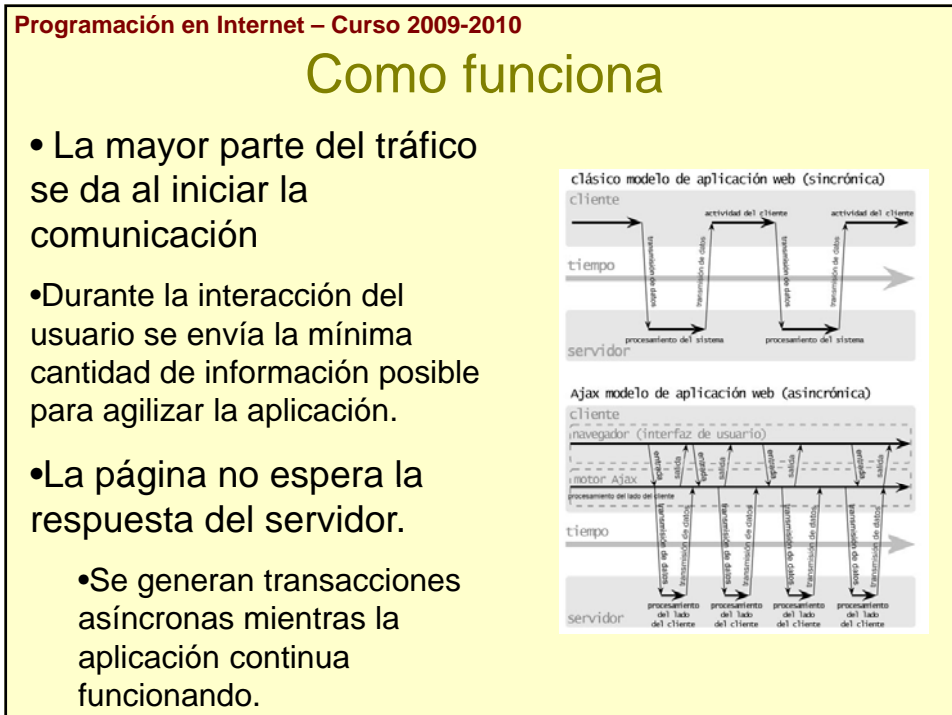
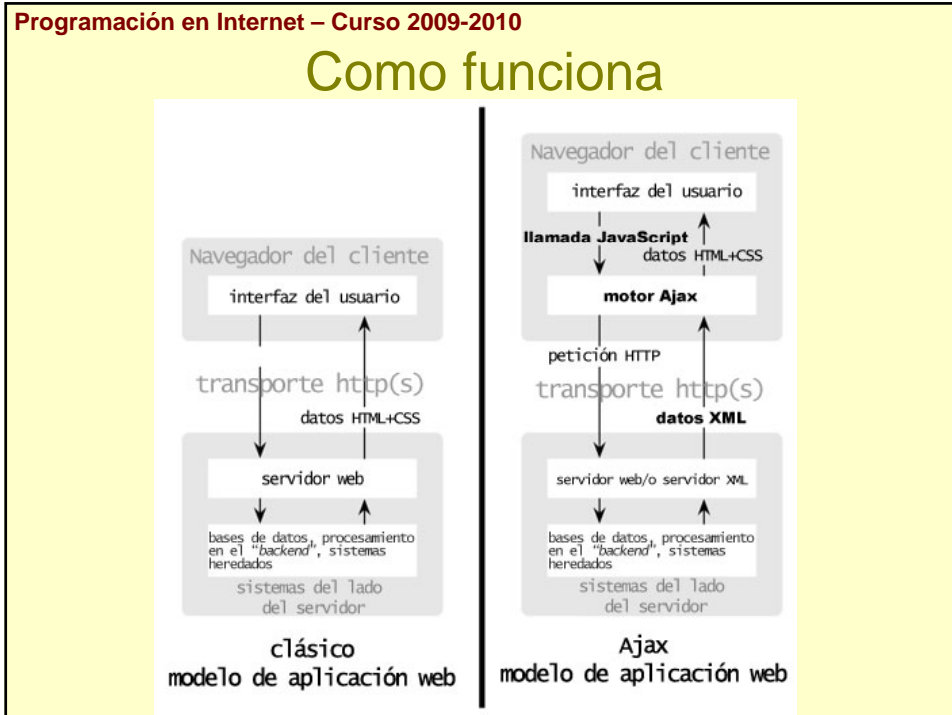
- La filosofía que se presenta con AJAX es:
  - Cargar y renderizar una página para luego mantenerse en esa misma página mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-dibujando la página y mostrando u ocultando porciones de la misma.

## Definición y conceptos

- Tecnologías usadas con Ajax:
  - **HTML/XHTML:** Presentación del contenido
  - **DOM:** Interacción en el cliente.
  - **JavaScript:** Lenguaje empleado para programar el motor Ajax.
    - Recoger la interacción del usuario.
    - Para realizar la comunicación con el servidor.
    - Para cambiar el contenido de la pantalla del cliente.
- Se aporta un nuevo enfoque al desarrollo de aplicaciones web.
  - Funcionamiento similar a las aplicaciones autónomas.
  - Cambio del concepto de programación clásico.
  - Reducción del tráfico que generan las aplicaciones web en cuanto al número y tamaño de las transacciones.

## Como funciona

- Se introduce una nueva capa entre el cliente y el servidor, llamada “Motor Ajax”
  - Se usa un objeto del DOM llamado XMLHttpRequest
  - Por medio de este objeto se realiza la comunicación con el servidor
  - Permite obtener información del servidor y actualizar la interfaz de usuario
- La respuesta que se obtiene del servidor puede estar en varios formatos:
  - Texto plano
  - HTML
  - XML
  - JSON
  - ...



Programación en Internet – Curso 2009-2010

## Ventajas

- Transacciones asíncronas, el usuario no tiene que esperar después de una petición.
- Menor cantidad de información.
- Tecnologías basadas en estándares abiertos.
- Significa una aproximación a las aplicaciones tradicionales en cuanto a interactividad y velocidad de respuesta.

Programación en Internet – Curso 2009-2010

## Desventajas

- Puede aumentar el número de peticiones contra el servidor.
- Existen incompatibilidades entre navegadores.
  - El objeto XMLHttpRequest no se invoca de la misma forma en los navegadores.
- Pérdida de control de navegación.
  - Los botones atrás y adelante del navegador dejan de funcionar.
  - No se puede crear un favorito (marcador) a una página, porque siempre es la misma página.

## XMLHttpRequest

- Permite la comunicación asíncrona con el servidor.
- En proceso de estandarización por el W3C:
  - The XMLHttpRequest Object W3C Working Draft 15 April 2008:
    - <http://www.w3.org/TR/XMLHttpRequest/>

## XMLHttpRequest

- Métodos:
  - **abort()**
  - **getAllResponseHeaders()**
  - **getResponseHeader(etiqueta)**
  - **open(metodo, url, asincrona, usuario, password):**
    - Métodos: GET, POST, HEAD, POST o DELETE.
  - **send(contenido)**
  - **setRequestHeader(etiqueta, valor)**

## XMLHttpRequest

- Propiedades:
  - **onreadystatechange**: Asignación de manejador al evento de cambio de estado.
  - **readyState**: Estado de la conexión
    - 0: no inicializada
    - 1: cargando
    - 2: cargada
    - 3: interactiva
    - 4: completada
  - **responseText**: La respuesta en forma de texto plano.
  - **responseXML**: La respuesta en formato XML.
  - **status**: Código de respuesta del servidor.
    - Códigos HTTP. 200: Transacción completada con éxito.
  - **statusText**: Mensaje asociado a la respuesta.

## Ciclo de vida de una transferencia con XMLHttpRequest

- Se **crea el objeto** XMLHttpRequest
- Se le **asigna un manejador** del evento de cambio de estado
- Se **invoca la petición** en función de la interacción del usuario
- Al **recibir un cambio de estado**, se gestiona la respuesta.
- Si la respuesta es correcta, se **obtienen los datos** y se muestran al cliente.
- Se debe vincular a algún objeto de la página, la invocación de la petición al servidor, ya sea mediante un click, un link o cualquier otro evento.



## Creación del objeto XMLHttpRequest

- Según el navegador se crea de dos formas:

- Internet Explorer:

```
peticion = new ActiveXObject("Microsoft.XMLHTTP");
```

- Otros navegadores:

```
peticion = new XMLHttpRequest();
```

- Código fuente independiente del navegador

```
if (window.XMLHttpRequest)
    peticion = new XMLHttpRequest();
else if(window.ActiveXObject)
    peticion = new ActiveXObject("Microsoft.XMLHTTP");
else {
    alert("Su navegador no soporta AJAX");
    return;
}
```

## Creación del objeto XMLHttpRequest

- Otra forma, mediante excepciones:

```
try {
    peticion = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
    try {
        peticion = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (ee) {
        try {
            peticion = new XMLHttpRequest();
        } catch(eee) {
            alert("Su navegador no soporta AJAX");
            return;
        }
    }
}
```

Programación en Internet – Curso 2009-2010

## Asignar un manejador

- Se debe asignar una función que se invoque cuando cambie el estado del objeto XMLHttpRequest, es decir, cuando el servidor responda a nuestra petición:

```
peticion.onreadystatechange = nombreDeLaFuncion;
```

Sin paréntesis, solo el nombre.

Programación en Internet – Curso 2009-2010

## Invocar la petición

- Se debe especificar el tipo de petición, la URL que deseamos invocar y si la transacción es asíncrona o no.
  - Si no es asíncrona, el navegador se bloquea hasta la respuesta.
  - La URL tiene que ser del mismo dominio que la página actual

```
peticion.open('GET', 'http://www.ua.es/ajax.php', true);
```

- Después se envía la transacción con la función send:

```
peticion.send(parámetros de la petición o null);
```

## Gestión de la respuesta

- El manejador se invoca al recibir un cambio de estado de la transacción.
  - 0: no inicializada
  - 1: cargando
  - 2: cargada
  - 3: interactiva
  - 4: completada

- Fragmento de código del manejador:

```
if (peticion.readyState == 4) { //fin
  if (peticion.status==200) { // todo OK
    //procesaríamos la petición
  }
} else {
  // Todavía no está...
}
```

## Tratamiento de los datos

- La respuesta puede estar en varios formatos: texto plano, HTML, XML, etc.
- La propiedad `responseText` nos da acceso al texto plano de la misma:

```
peticion.responseText
```

- Si es XML, con la propiedad `responseXML` obtenemos un objeto de tipo `XMLDocument`

## Ejemplo (1)

- Una página web con un botón
- Al pulsar el botón, mediante AJAX se llama a un fichero PHP
- El fichero PHP devuelve la hora en el servidor
- En el cliente se muestra el resultado devuelto con una ventana de aviso

## Ejemplo (1): código HTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Transitional//EN" "xhtml1-transitional.dtd">
<html>
<head>
<title>Sentido Web - AJAX</title>
</head>
<body>
<input type="button" onclick="holaMundo()" value="Hola
  mundo" />
</body>
</html>
```

Programación en Internet – Curso 2009-2010

## Ejemplo (1): funciones JavaScript

```
function ajaxobj() {
  try {
    _ajaxobj = new ActiveXObject("Msxml2.XMLHTTP");
  } catch (e) {
    try {
      _ajaxobj = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (ee) {
      _ajaxobj = false;
    }
  }

  if (!_ajaxobj && typeof XMLHttpRequest!='undefined')
    _ajaxobj = new XMLHttpRequest();

  return _ajaxobj;
}
```

Programación en Internet – Curso 2009-2010

## Ejemplo (1): funciones JavaScript

```
function holaMundo() {
  ajax = ajaxobj();
  ajax.onreadystatechange=function() {
    if (ajax.readyState==4) {
      alert(ajax.responseText);
    }
  }

  ajax.open("GET", "holamundo.php", true);
  ajax.send(null);
}
```

Programación en Internet – Curso 2009-2010

## Ejemplo (1): fichero PHP

```
<?php
    echo "Hola, son las ".date("H:i:s");
?>
```

Programación en Internet – Curso 2009-2010

## Ejemplo (2)

- Diseño de una página web con un listado de poblaciones.
- En función de la población que se seleccione, por medio de un click se mostrará a la derecha (en un layer) un listado de calles de la misma.
- Las calles se solicitarán al servidor por medio de una petición asíncrona (AJAX), accediendo a un fichero XML con las calles de la población.

Programación en Internet – Curso 2009-2010

## Ejemplo (2): fichero de datos

```
<?xml version="1.0"?>
<callejero>
  <poblacion>Benimagrell</poblacion>
  <calles>
    <calle tipo="avenida">del pais valencia</calle>
    <calle tipo="calle">alacant</calle>
    <calle tipo="calle">sant vicent</calle>
    <calle tipo="calle">del tordo</calle>
    <calle tipo="calle">del pulpo</calle>
    <calle tipo="plaza">del ayuntamiento</calle>
    <calle tipo="plaza">nova del ravalet</calle>
  </calles>
</callejero>
```

Programación en Internet – Curso 2009-2010

## Ejemplo (2): funciones en JavaScript

```
function fInvoca(pURL) {
  if (window.XMLHttpRequest) {
    oPetición = new XMLHttpRequest();
  } else if (window.ActiveXObject) {
    oPetición = new ActiveXObject("Microsoft.XMLHTTP");
  }
  if (oPetición) {
    oPetición.onreadystatechange=fGestionPetición;
    oPetición.open('GET', pURL, true);
    oPetición.send(null);
  }
}
function fGestionPetición() {
  if (oPetición.readyState==4) {
    if (oPetición.status==200) {
      //alert(oPetición.responseText);
      fProcesaDatos(oPetición.responseXML);
    } else {
      alert("Error en la petición de datos al servidor!\nError:
"+oPetición.status);
    }
  }
};
}
```

Programación en Internet – Curso 2009-2010

## Ejemplo (2): funciones en JavaScript

```
function fProcesaDatos(pDoc) {
    oCapa = document.getElementById("calles");
    //Nombre de la población:
    oPoblacion = pDoc.getElementsByTagName("poblacion").item(0);
    sHTML = "<h1>" + oPoblacion.text + "</h1>";
    //Lista de calles:
    sHTML = sHTML + "<br><ul>";
    oCalles = pDoc.getElementsByTagName("calle");
    for (var i=0; i<oCalles.length; i++) {
        // Ponemos el tipo de vía y el nombre:
        sHTML = sHTML + "<li>" + oCalles.item(i).getAttribute("tipo")
+ " " + oCalles.item(i).text + "</li>";
    }
    sHTML = sHTML + "</ul>";
    oCapa.innerHTML = sHTML;
}
```

Programación en Internet – Curso 2009-2010

## Ejemplo (2): código HTML

```
<body>
<div id="poblaciones">
<ul>
<li><a
href="javascript:fInvoca('benimagrell.xml');">Benimag
rell</a></li>
<li><a
href="javascript:fInvoca('mutxamel.xml');">Mutxamel</
a></li>
<li><a
href="javascript:fInvoca('villena.xml');">Villena</a>
</li>
</div>
<hr>
<div id="calles">
</div>
</body>
```



## Bibliografía

- AJAX (Wikipedia):
  - <http://es.wikipedia.org/wiki/AJAX>
- Ajax: A New Approach to Web Applications:
  - <http://adaptivepath.com/publications/essays/archives/000385.php>
- AJAX un nuevo acercamiento a aplicaciones web (traducción del anterior):
  - <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>
- “Hola mundo” en Ajax:
  - <http://www.maestrosdelweb.com/editorial/ajaxpaso/>
- Introducción a Ajax:
  - <http://www.librosweb.es/ajax/index.html>