

# TEMA 7

## Utilización de mecanismos de comunicación asíncrona

### 1 MECANISMOS DE COMUNICACIÓN ASÍNCRONA

#### Ejecución de una aplicación web: modelo moderno (AJAX)

¿De qué tipo de comunicación estamos hablando (síncrona o asíncrona) cuando el cliente envía una petición y permanece bloqueado esperando la respuesta del receptor?

Comunicación asíncrona.

¿Qué significan las siglas AJAX?

Asynchronous JavaScript And XML, que en español significan: JavaScript asíncrono y XML.

¿Qué es AJAX?

Es una técnica de desarrollo de aplicaciones web que permite la creación de aplicaciones interactivas.

### 1.1. DEFINICIÓN DE AJAX

#### Introducción a AJAX - Parte 1

Traduce las siguientes frases al castellano:

1. **AJAX is not a new programming language, but a new way to use existing standards.**
  2. **AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.**
- 
1. AJAX no es un nuevo lenguaje de programación, sino una nueva forma de utilizar los estándares existentes.
  2. AJAX es el arte de intercambiar datos con un servidor, y actualizar partes de una página web, sin recargar la página completa.

### **¿Qué objetivo tiene AJAX?**

Intercambiar pequeñas entidades de datos entre el cliente y el servidor recargando partes de la página web sin la necesidad de recargar todo el contenido de la misma.

### **¿Es AJAX un nuevo lenguaje de programación?**

No.

### **¿Qué tecnologías incluye AJAX?**

- XHTML y CSS.
- DOM
- XML y XSLT
- XMLHttpRequest
- JavaScript

## **1.2. ELECCIÓN DE AJAX**

## **1.3. REPASO A LAS TECNOLOGÍAS INVOLUCRADAS**

XHTML y CSS

### **¿Cuál es la principal diferencia entre XHTML y HTML?**

La principal diferencia entre XHTML y HTML es su grado de especificación. XHTML es un HTML estándar especificado mediante un documento XML. XHTML es más riguroso con su estructura que HTML (justamente como un fichero XML).

DOM

JavaScript

XML y XSLT

### **¿Para qué se utiliza el lenguaje XML?**

El lenguaje XML es utilizado para describir y estructurar datos.

El objeto XMLHttpRequest

### **¿Qué función tiene el objeto XMLHttpRequest?**

Sirve como una interfaz con la que se realizan peticiones a servidores web.

**¿Cómo podemos instanciar el objeto XMLHttpRequest para asegurarnos de que se crea correctamente en cualquier navegador?**

```
Var httpRequest;  
if (window.XMLHttpRequest) {  
    httpRequest = new XMLHttpRequest();  
} else if (window.ActiveXObject) {  
    httpRequest = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

**Ordena las siguientes acciones, según la secuencia de utilización normal del objeto XMLHttpRequest:**

- **Enviar la petición con el método send().**
  - **Obtener la respuesta en los diferentes atributos del objeto XMLHttpRequest (como una cadena, como un documento XML o como un array de bytes).**
  - **Agregar la función que se disparará en el evento onreadystatechange.**
  - **Abrir un canal de comunicación y especificar la petición (de conexión) con el método open().**
1. Abrir un canal de comunicación y especificar la petición (de conexión) con el método open().
  2. Enviar la petición con el método send().
  3. Agregar la función que se disparará en el evento onreadystatechange.
  4. Obtener la respuesta en los diferentes atributos del objeto XMLHttpRequest (como una cadena, como un documento XML o como un array de bytes).

**Indica cuáles son los valores y su significado correspondiente, que podemos encontrar en el atributo readyState del objeto XMLHttpRequest.**

Devuelve el estado del objeto como sigue:

- 0 = sin inicializar
- 1 = abierto
- 2 = cabeceras recibidas
- 3 = cargando
- 4 = completado

**¿Si se espera recibir la respuesta del servidor en forma de documento XML, en qué atributo del objeto XMLHttpRequest encontraremos dicha respuesta?**

En el atributo responseXML.

**¿Qué encontraremos en el atributo/propiedad status del objeto XMLHttpRequest? Indica algunos posibles valores y sus significados.**

El código de estado HTTP devuelto por el servidor (200 para una respuesta correcta, 404 para "No encontrado", 500 para un error de servidor, etc.)

### ¿Cuántos parámetros se deben pasar al método open?

El método `open()` requiere dos parámetros (método HTTP y URL) y acepta de forma opcional otros tres parámetros.

### Definición formal del método `open()`:

`open(string metodo, string URL [,boolean asincrono, string usuario, string password]);`

### ¿Qué significado tiene el tercer parámetro del método open?

Por defecto, las peticiones realizadas son asíncronas. Si se indica un valor `false` al tercer parámetro, la petición se realiza de forma síncrona, esto es, se detiene la ejecución de la aplicación hasta que se recibe de forma completa la respuesta del servidor.

### ¿Qué función tiene el método send y qué parámetro hay que pasarle?

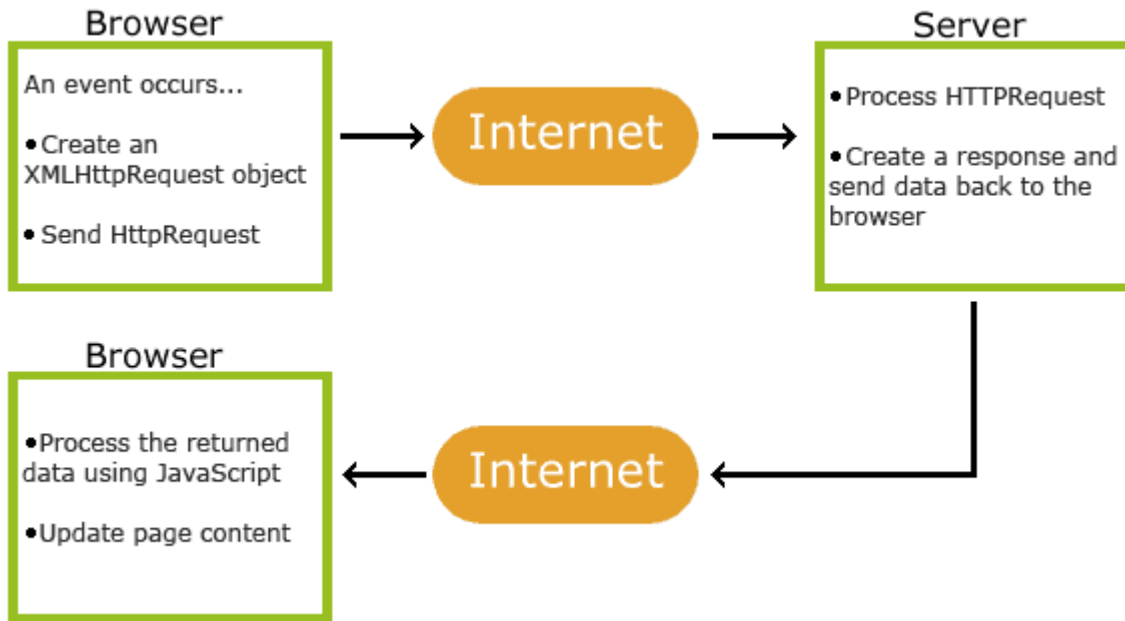
El método `send()` requiere de un parámetro que indica la información que se va a enviar al servidor junto con la petición HTTP. Si no se envían datos, se debe indicar un valor igual a `null`. En otro caso, se puede indicar como parámetro una cadena de texto, un array de bytes o un objeto XML DOM.

### ¿Cuál es el evento del objeto XMLHttpRequest que se produce cada vez que se produce un cambio en el estado de la petición HTTP?

`Onreadystatechange`

## 1.4. PERSPECTIVA GLOBAL DE UN DESARROLLO AJAX

Dibuja un esquema que represente el funcionamiento de AJAX.



## 2 FORMATOS PARA EL ENVÍO Y RECEPCIÓN DE INFORMACIÓN. XML Y JSON.

### Introducción a JSON

¿Qué significan las siglas JSON?  
JavaScript Object Notation

¿Qué es JSON?  
Por un lado es un formato ligero para el intercambio de datos y, por otro lado, es una manera de almacenar información.

### 2.1. SINTAXIS DE JSON.

¿Cómo se llama el elemento base de la sintaxis de JSON?  
El elemento base de la sintaxis es el object.

¿Qué caracteres delimitan un object de JSON?  
Comienza con una llave de apertura ' { ' y finaliza con una llave de cierre ' } '.

¿Qué elemento de JSON queda delimitado entre corchetes?  
Un array.

### 3 Ejemplo de comunicación asíncrona.

#### Ejemplo del libro. Código.

```
<html>
<head>
<meta charset="utf-8" />
<title>Ejemplo de AJAX</title>
<script language = "javascript">
    var objetoXHR = false;

    if (window.XMLHttpRequest) {
        objetoXHR = new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        objetoXHR = new ActiveXObject("Microsoft.XMLHTTP");
    }

    function obtenerDatosServidor(orien,elemento) {
        if (objetoXHR) {
            var objeto_destino = document.getElementById(elemento);
            objetoXHR.open("GET", origen);
            objetoXHR.onreadystatechange = respuesta();
            objetoXHR.send(null);
        }
    }

    function respuesta() {
        if (objetoXHR.readyState == 4 &&
            objetoXHR.status == 200) {
            objeto_destino.innerHTML = objetoXHR.responseText;
        }
    }

</script>
</head>

<body>
    <h1>Ejemplo de AJAX</h1>
    <form>
        <input type = "button" value = "Buscar información"
            onclick = "obtenerDatosServidor('http://Web/datos.txt',
'elemento_destino') ">
    </form>
    <div id="elemento_destino">
        <p>La información aparecerá aquí</p>
    </div>
</body>
</html>
```

**En la función obtenerDatosServidor hay dos parámetros. ¿Qué es lo que se le pasa como primer parámetro?**

Es la URL de un archivo de texto (datos.txt), que contiene el texto plano que vamos a mostrar en la página web, que proviene del servidor web y que es descargado en segundo plano.

**¿Qué nombre recibe, en nuestro ejemplo, la función que será llamada cada vez que el estado del objeto XMLHttpRequest cambie?**

En este caso, la función se llama “respuesta”.

**¿En qué propiedad del objeto XMLHttpRequest comprobamos los cambios en el estado del objeto XMLHttpRequest?**

En la propiedad onreadystatechange.

**¿Qué función se llamará cuando los datos terminen de cargarse?**

Se llamará a la función “respuesta”.

**¿Qué propiedades debemos tener en cuenta una vez que se han terminado de cargar los datos?**

Las propiedades readyState y status.

**¿Qué es lo que nos indicará la propiedad readyState? ¿y con qué valor?**

La propiedad readyState nos indicará cuando se han terminado de recibir los datos, tomando el valor “4”.

**¿En qué propiedad podemos comprobar si la comunicación con la URL ha sido correcta?**

Con la propiedad status, si se recibe el valor “200”, quiere decir que la comunicación es correcta.

**¿Cómo podemos leer los datos obtenidos del servidor?**

A partir de una de las propiedades del objeto XMLHttpRequest, en este caso responseText.



## **3.1. COMUNICACIÓN CON XML**

**Si queremos enviar datos al servidor, en formato XML, ¿qué es lo primero que es necesario hacer?**

Lo primero es indicar el formato de envío a partir del método `setRequestHeader()`.

**¿Cómo se enviará después el texto XML?**

Utilizando el método `send()`.

## 4. Librerías de actualización dinámica.

### 4.1 JQUERY