



iesperemariaorts

Desarrollo web en entorno cliente

Tema 2

Ajax



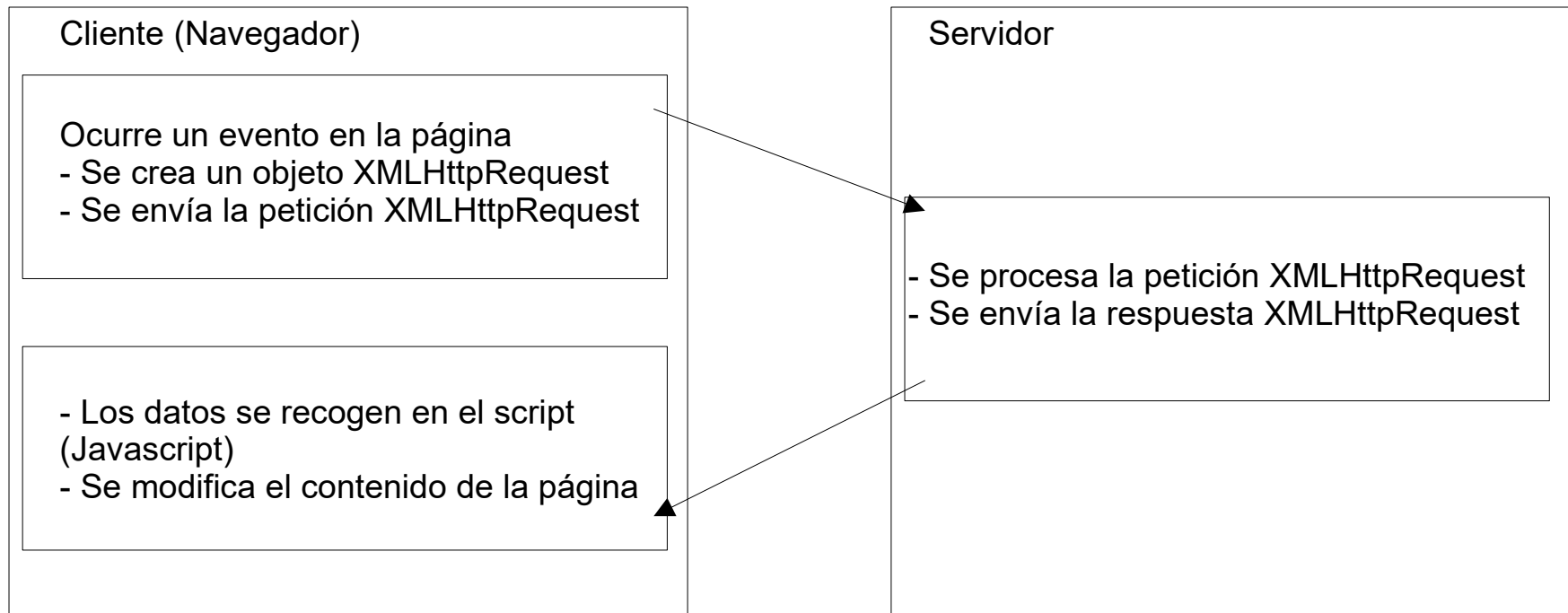
AJAX

- Asynchronous JavaScript and XML
- Petición asíncrona al servidor para no recargar entera la página web
- Solo se intercambian datos (ahorro recursos, ancho de banda...)
- Antigüamente: recargar páginas enteras.



AJAX

- AJAX se basa en el objeto XMLHttpRequest para intercambio de datos con servidor de forma asíncrona





iesperemariaorts

AJAX

- Las peticiones que se envían/reciben son mediante HTTP
 - Petición al servidor: Se envían cabeceras con métodos, datos...
 - Métodos: GET, POST, PUT, DELETE



iesperemariaorts

AJAX

MÉTODO GET

- Recupera datos del servidor
- Normalmente no modifica nada
- Sería como un SELECT en SQL
- Los datos serán enviados en la URL



iesperemariaorts

AJAX

MÉTODO POST

- Inserta o crea nuevos datos en el servidor
 - Sería como un INSERT en SQL
- Los datos se envían dentro de la petición HTTP



AJAX

MÉTODO PUT

- Actualiza datos en el servidor
- Sería como un UPDATE en SQL
- Los datos que identifican al objeto que va a ser actualizado se envían en la URL
- Los nuevos datos del objeto se envían aparte, como en un método POST.



iesperemariaorts

AJAX

MÉTODO DELETE

- Borra datos en el servidor
- Sería como un DELETE en SQL
- Los datos que identifican al objeto que se va a borrar se envían en la URL



AJAX – XMLHttpRequest (XHR)

- Crear petición al servidor, instanciaremos a XHR
`obj_ajax = new XMLHttpRequest()`
- **Método OPEN** (`open(método,url,asíncrono)`):
`obj_ajax.open("GET","http://localhost/xxxx",true)`
- **Método:** Tipo de método usado (POST,GET)
- **URL:** Fichero que contiene el código que procesa la solicitud (web, php, xml, json...)
- **Asíncrono:** Si síncrona (`false`), si asíncrona (`true`)



AJAX – XMLHttpRequest (XHR)

- **Método SEND(datos):** Inicia la solicitud de AJAX.
obj_ajax.send()
- El parámetro que recibe (datos) puede ser de diferentes tipos.
- Será vacía si es un GET o un DELETE o con un string con los datos si se trata de un POST o un PUT



AJAX – XMLHttpRequest (XHR)

- **Eventos:** Los eventos con los que podemos trabajar son:
 - **loadstart:** Es lanzado cuando empieza la solicitud
 - **progress:** Cada 50 milisegundos
 - **abort:** Se lanza cuando se aborta la solicitud
 - **timeout:** Si se indica un tiempo, cuando venza lanza este evento
 - **loadend:** Se lanza al completar la solicitud.
 - **load:** Se lanza al completarse con éxito



AJAX – XMLHttpRequest (XHR)

- **Eventos:** Los eventos con los que podemos trabajar son:
 - **readystatechange:** Se lanza cuando cambia el valor de la propiedad **readyState**, que toma los siguientes valores:
 - 1 – opened – El método open() ha sido llamado
 - 2 – headers-received – El método send() ha sido llamado y cabeceras y estados estan disponibles
 - 3 – loading – bajando, responseText() contiene datos parciales
 - 4 – done – la operación ha terminado



AJAX – XMLHttpRequest (XHR)

- **Propiedades "status"**: Nos indica si todo ha ido correctamente o no con los códigos:
 - 200 – OK
 - 404 – Recurso no encontrado
 - 500 – Error interno del servidor
- Cuando `readyState == 4 && status == 200` entonces todo correcto



AJAX – XMLHttpRequest (XHR)

- **Propiedades "response"**: Recoge los datos devueltos por el servidor:

- response – Retorna la respuesta de la solicitud
- responseText – Retorna respuesta en formato texto
- responseXML – Retorna respuesta en formato XML



AJAX – XMLHttpRequest (XHR)

- Evento "**progress**" propiedades:
 - LengthComputable: Devuelve true si se puede realizar cálculo del progreso
 - loaded – Retorna total de byte que se han enviado o recibido de la petición ajax
 - total – Retorna el total de bytes que se van a transmitir.



iesperemariaorts

