

**Ajax** es una tecnología que nos permite desde una aplicación web en ejecución obtener datos de una aplicación servidora en segundo plano, pudiendo seguir utilizando dicha aplicación web. Es decir, la aplicación no se queda bloqueada mientras llega la respuesta de la aplicación servidora.

**Mostrar ejercicio ej9UT7.html** sin explicar cómo se ejecuta.

**Vamos a ver cómo se ejecutaría una aplicación web** (ej9UT7.html) que hace una petición de datos a una aplicación servidora:

- El cliente solicita los datos a una aplicación servidora mediante una petición HTTP-> Ajax
- Esta aplicación debe tener una URL.
- Al ser invocada dicha aplicación se abre una petición HTTP que tiene que ser atendida por el servidor web en el que está alojada la aplicación servidora a la que le estamos pidiendo los datos.
- En los ejercicios en los que la aplicación servidora está alojada en un sitio web externo que no es nuestro podemos ejecutar nuestra aplicación sin tener arrancado XAMPP.
- En los ejercicios en los que la aplicación servidora es nuestra tenemos que tenerla alojada en un sitio web (por ejemplo localhost) de un servidor web (por ejemplo XAMPP).
- Si la aplicación cliente y servidora pertenecen al mismo sitio web (por ejemplo localhost), podemos no poner en la URL la dirección del sitio web (localhost/)

**Para crear la petición Ajax** en nuestra aplicación cliente se utiliza el objeto XMLHttpRequest:

```
- Var xhr= new XMLHttpRequest()  
- xhr.open("GET", url, true);  
- xhr.send();
```

Mientras se puede seguir utilizando la aplicación web.

**¿Cómo sabemos cuándo la aplicación servidora nos ha respondido?**

La petición Ajax va pasando por varios estados, que se guardan en la propiedad readyState del objeto xhr:

- 1: Conexión establecida con el servidor.
- 2: Solicitud recibida por el servidor.
- 3: El servidor está procesando la petición.
- 4: El servidor ha dado una respuesta al cliente. Esta respuesta puede ser positiva o no, esto se guarda en la propiedad status del objeto xhr, que puede valer:
  - o 200: la aplicación servidora se ha ejecutado correctamente. La respuesta se recibe en las propiedades del objeto xhr siguientes:.responseText y responseXML
  - o 404: la aplicación servidora no se ha encontrado (por ejemplo la URL de la aplicación servidora es incorrecta).
  - o 403: forbidden: la ejecución de la aplicación servidora ha dado un error de permisos.

Cuando cambia de valor la propiedad readyState, del objeto xhr a través del cual se ha hecho la petición Ajax, se produce un evento llamadoreadystatechange, por lo que utilizando la gestión de eventos podemos hacer que cada vez que se produzca este evento se ejecute una función manejadora del evento que recibirá la respuesta del servidor.

```
var xhr= new XMLHttpRequest()  
xhr.open("GET", url, true);
```

```
function functionAjax() {  
    if (this.readyState == 4 && this.status == 200) {  
        document.getElementById("demo1").innerHTML = this.responseText;  
    }  
}  
xhr.send();
```

- `responseText` la utilizaremos cuando la respuesta esperada sea:
  - o Una cadena, por ejemplo de una aplicación servidora php (OK, NOK, un código, etc) se recibirá en la propiedad `responseText`
  - o Un fichero HTML.
  - o Una cadena que guarda un objeto JSON. En el cliente habrá que parsearlo de cadena a objeto JSON.
- `responseXML` la utilizaremos cuando la respuesta esperada sea un fichero XML. En ese caso también estará la misma respuesta en `responseText` pero no será en formato XML, y por tanto, no podremos utilizar los métodos de manejo del DOM XML de JavaScript.

- Herramientas del desarrollador.
- Monitor de red
- Seleccionar XHR.
- Al seleccionar una de las peticiones XHR (la línea azul que aparece más abajo) podemos consultar la respuesta dada por la aplicación:

Estado	Método	Dominio	Archivo	Protocolo	Iniciador	Tipo	Cookies	Set-Cookies	Tamaño
200	GET	localhost	eg9U7.php?provincia=B	HTTP/1.1	eg9U7.html:58 (xhr)	html			279 B
200	GET	localhost	eg9U7.php?provincia=B	HTTP/1.1	eg9U7.html:58 (xhr)	html			278 B

Badajoz, Barcelona, Burgos