



APRENDERAPROGRAMAR.COM

AJAX: EJEMPLO BÁSICO  
CÓDIGO OBJETO  
XMLHTTPREQUEST Y  
RECUPERAR DATOS DESDE  
ARCHIVO (CU01206F)

Sección: Cursos

Categoría: Tutorial básico del programador web: Ajax desde cero

Fecha revisión: 2031

**Resumen:** Entrega nº6 del Tutorial básico "Ajax desde cero".

Autor: Alex Rodríguez

## EJEMPLO BÁSICO CÓDIGO AJAX

Hemos discutido distintos aspectos conceptuales sobre Ajax. Todavía nos faltan algunas cuestiones por exponer o discutir, pero vamos a tratar de ir combinando la exposición de ideas y conceptos con el desarrollo de código y la visualización de resultados.



En este momento vamos a centrarnos en crear un ejemplo básico de ejecución de Ajax, sobre el que vamos a explicar diferentes conceptos. En primer lugar vamos a crear los archivos necesarios y comprobar que todo funcione correctamente. Sigue las instrucciones que vamos a ir dando a continuación. Crea un archivo denominado cursoAjaxCU01206F.html con el contenido que indicamos a continuación. No te preocupes ahora por entender todo el código, lo explicaremos después:

```
<!DOCTYPE html><html><head><title>Cursos aprende a programar</title><meta charset="utf-8">

<style type="text/css">
*{font-family:sans-serif;} a:link {text-decoration:none;} select{font-size:18px;}
div div {color: blue; background-color:#F1FEC6; font-size: 20px; float:left; border: solid; margin: 20px;
padding:15px;}
</style>

<script>
function mostrarSugerencia(str) {
var xmlhttp;
var contenidosRecibidos = new Array();
var nodoMostrarResultados = document.getElementById('listaCiudades');
var contenidosAMostrar = "";

if (str.length==0) { document.getElementById("txtInformacion").innerHTML="";
nodoMostrarResultados.innerHTML = ""; return; }

xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
if (xmlhttp.readyState==4 && xmlhttp.status==200) {
contenidosRecibidos = xmlhttp.responseText.split(",");
document.getElementById("txtInformacion").innerHTML=contenidosRecibidos[0];
for (var i=1; i<contenidosRecibidos.length;i++) {
contenidosAMostrar = contenidosAMostrar+'<div id="ciudades'+i+"> <a
href="http://aprenderaprogramar.com">' + contenidosRecibidos[i]+ '</a></div>';
}
nodoMostrarResultados.innerHTML = contenidosAMostrar;
}
}

xmlhttp.open("GET"," datosCU01206F.php?pais="+str);
xmlhttp.send();
}

</script>
</head>
```

```

<body style="margin:20px;">
<h2>Elige un país:</h2>
<form action="">
<select onchange="mostrarSugerencia(this.value)">
<option value="none">Elige</option>
<option value="spain">España</option>
<option value="mexico">México</option>
<option value="argentina">Argentina</option>
<option value="colombia">Colombia</option>
</select>
</form>
<br/>
<p>Informacion sobre operacion en segundo plano con Ajax: <span style="color:brown;"
id="txtInformacion"></span></p>
<div id="listaCiudades"> </div>
</body>
</html>

```

Creas un archivo denominado datosCU01206F.php con el siguiente contenido:

```

<meta charset = "utf-8" />
<?php // Datos

$pais[0]="spain"; $pais[1]="mexico"; $pais[2]="argentina"; $pais[3]="colombia";

$ciudad[0][0]="Madrid"; $ciudad[0][1]="Barcelona"; $ciudad[0][2]="Valencia"; $ciudad[0][3]="Sevilla";
$ciudad[0][4]="Zaragoza"; $ciudad[0][5]="Málaga"; $ciudad[0][6]="Murcia";
$ciudad[1][0]="México D.F."; $ciudad[1][1]="Ecatepec"; $ciudad[1][2]="Guadalajara"; $ciudad[1][3]="Puebla";
$ciudad[1][4]="Juárez"; $ciudad[1][5]="Tijuana"; $ciudad[1][6]="León"; $ciudad[1][7]="Zapopan";
$ciudad[2][0]="Buenos Aires"; $ciudad[2][1]="Córdoba"; $ciudad[2][2]="Rosario"; $ciudad[2][3]="La Plata";
$ciudad[2][4]="Mar del Plata"; $ciudad[2][5]="San Miguel de Tucumán"; $ciudad[2][6]="Ciudad de Salta";
$ciudad[3][0]="Bogotá"; $ciudad[3][1]="Medellín"; $ciudad[3][2]="Cali"; $ciudad[3][3]="Barranquilla";
$ciudad[3][4]="Cartagena"; $ciudad[3][5]="Cúcuta"; $ciudad[3][6]="Soledad"; $ciudad[3][7]="Ibagué";

// Rescatamos el parámetro pais que nos llega mediante la url que invoca xmlhttp
$paisRecibido=$_REQUEST["pais"]; $ciudadesDevueltas="";
$existePais = false;
for ($i = 0; $i<count($pais); $i++) {      if ($paisRecibido == $pais[$i]) {$indicePais = $i; $existePais=true;}}

$msg = 'El pais recibido por get en segundo plano es '.$paisRecibido ;
if ($existePais) {$msg = $msg. ' y tiene indice '.$indicePais;}
$ciudadesRespuesta = "";

// Creamos el array a devolver
for ($i = $indicePais; $i<count($ciudad[$indicePais]); $i++) {
    $ciudadesRespuesta .= ",".$ciudad[$indicePais][$i];
}

echo $msg.$ciudadesRespuesta;
?>

```

## ANÁLISIS DEL FICHERO PHP

Para seguir este curso son necesarios unos conocimientos previos que incluyen HTML, CSS, JavaScript y PHP. Si no tienes estos conocimientos previos dirígete a las entregas iniciales del curso y lee las recomendaciones.

Nuestro fichero php es un fichero que contiene información en forma de arrays sobre países y ciudades que pertenecen a esos países. En concreto se utilizan cuatro países: España, México, Argentina y Colombia, y para cada país se definen las principales ciudades utilizando un array o arreglo. Así por ejemplo `$ciudad[3][4]` es el elemento del array ciudad con primer índice 3 y segundo índice 4. En este ejemplo el primer índice indica el país (0 España, 1 México, 2 Argentina y 3 Colombia) y el segundo índice la ciudad (en este caso la ciudad con índice 4). `$ciudad[3][4]` es "Cartagena". Estamos usando arrays con índice inicial cero, por tanto `$ciudad[3][4]` se lee como "quinta ciudad (índice cuatro) del cuarto país (índice 3)".

Para cada país se definen un grupo de ciudades.

En el fichero php se recupera (o se trata de recuperar) un parámetro usando REQUEST. Request nos permite recuperar tanto parámetros pasados por post como por get. Una vez identificado el parámetro esperado, que es el nombre de un país entre none, spain, mexico, argentina, colombia, se procede a comprobar si el parámetro país recibido coincide con alguno de estos valores. En caso de recibirse none o un valor extraño, el resultado o ejecución del fichero devuelve un código HTML que se define en la última línea mediante una sentencia echo. El código HTML devuelto es la concatenación de una variable `$msg` y otra variable `$ciudadesRespuesta`.

Si se ha recibido un país en el formato esperado, la respuesta será del tipo:

```
El pais recibido por get en segundo plano es spain y tiene indice 0, Madrid, Barcelona, Valencia, Sevilla, Zaragoza, Málaga, Murcia
```

Sube el fichero php y comprueba que obtienes esa respuesta al invocar la ruta del mismo, que será del tipo: `http://aprenderaprogramar.com/datosCU01206F.php?pais=spain`

Donde debes sustituir `aprenderaprogramar.com` por la ruta que corresponda en tu servidor.

Si no obtienes esa respuesta debes revisar y solucionar el problema antes de seguir avanzando.

En esta respuesta la primera frase corresponde a la variable `$msg` y la lista de ciudades empezando por una coma y separadas por comas corresponde a la variable `$ciudadesRespuesta`.

Supón que pasamos por get un país para el que no está preparado el fichero, por ejemplo supón que escribiéramos algo como `http://aprenderaprogramar.com/datosCU01206F.php?pais=chile`

El resultado que se debe mostrar por pantalla es: El pais recibido por get en segundo plano es chile

Bien, hasta ahora hemos visto lo que hace el archivo php: recibe un dato por get ó post, y devuelve un código HTML, en este caso una frase donde se incluye una lista de ciudades si el parámetro país tiene un valor determinado.

Comprueba que según la ruta invocada obtenemos la lista de ciudades del país recibido como parámetro. Por ejemplo `http://aprenderaprogramar.com/datosCU01206F.php?pais=colombia` nos devolverá una cadena que incluirá una lista de ciudades separadas por comas.

### ANÁLISIS DEL FICHERO HTML

En el fichero HTML tenemos un contenido HTML que incluye un formulario donde se permite elegir entre varios países (España, México, Argentina, Colombia) y el atributo value asociado a cada opción es exactamente el que espera recibir el fichero php.

Además tenemos algunos estilos definidos con CSS.

Finalmente tenemos un código JavaScript con una función denominada mostrarSugerencia(str). Esta función se ejecuta como respuesta al evento onchange que se produce cuando el usuario elige un país en el combobox desplegable del formulario. Al elegirse un país se pasa a la función this.value, que contiene el atributo value del elemento del combobox elegido. Los valores posibles son none, spain, mexico, argentina y colombia.

En el código html se ha dejado un div con id listaCiudades que se encuentra vacío (<div id="listaCiudades"> </div>). En este div es donde tenemos previsto mostrar las ciudades que recuperemos del servidor en función de la opción elegida por el usuario, usando Ajax para ello.

El código HTML y CSS debes ser capaz de comprenderlo. Si no es así, dirígete a las entregas iniciales del curso y lee las recomendaciones.

Nos vamos a centrar en el código JavaScript, que es donde se encuentra Ajax incorporado.

### RECUPERACIÓN DE INFORMACIÓN DESDE EL SERVIDOR EN SEGUNDO PLANO

Sube el fichero HTML al servidor (en la misma ruta donde tengas el fichero PHP) e invoca la ruta donde se encuentre, que será del tipo http://aprenderaprogramar.com/cursoAjaxCU01206F.html.

Ahora debes ser capaz de elegir opciones del combobox desplegable y visualizar resultados por pantalla. Por ejemplo, si eliges la opción "Argentina", el resultado esperado será que por pantalla se visualice:

Informacion sobre operacion en segundo plano con Ajax: El pais recibido por get en segundo plano es argentina y tiene indice 2				
Rosario	La Plata	Mar del Plata	San Miguel de Tucumán	Ciudad de Salta



Si cambias la selección del país por otro país, debes observar un refresco “casi instantáneo” de modo que se modifica el texto y las ciudades que se muestran, viéndose en cada caso las correspondientes al país seleccionado.

Si eres capaz de visualizar esto, ya estás viendo Ajax en acción. Cada vez que seleccionamos un país, hay una comunicación con el servidor en segundo plano y la información recibida del servidor la utilizamos para hacer cambios en la página web sin necesidad de hacer una recarga tradicional.

## COMPRENDIENDO LOS CONCEPTOS

En la siguiente entrega del curso haremos un análisis pormenorizado del código. Ahora vamos a centrarnos en explicar y entender a nivel de conceptos qué es lo que estamos haciendo. Para ello vamos a fijarnos en algunos aspectos:

`xmlhttp=new XMLHttpRequest();` define que la variable `xmlhttp` contiene un objeto `XMLHttpRequest`. Este objeto es un objeto predefinido disponible en JavaScript. Recordar que en JavaScript existen dos tipos de datos básicos: los tipos primitivo y los objetos. Dentro de los objetos existen algunos objetos especiales: el “objeto global”, el “objeto prototipo”, los arrays, las funciones, las clases definidas por el programador, los objetos predefinidos JavaScript (como `Date` para manejo de fechas, `RegExp` para manejo de expresiones regulares, etc.) y otros.

Un objeto tiene propiedades y métodos. El objeto `XMLHttpRequest` lo estudiaremos con detalle más adelante. De momento simplemente saber que es un objeto que nos permite comunicar con el servidor para obtener información de él.

Mediante la propiedad `onreadystatechange` indicamos el código que debe ejecutarse una vez se haya completado la petición y transferencia de datos desde el servidor.

Mediante el método `open` estamos creando una petición HTTP sencilla al servidor: se hace una petición GET sobre la url indicada, pasando como parámetro el país elegido por el usuario. Seguidamente la petición se envía al servidor con el método `send`.

Una vez enviada la petición, tan solo resta comprobar que se ha completado correctamente y ejecutar la función indicada utilizando la respuesta del servidor, que vendrá en la propiedad `responseText` del objeto `XMLHttpRequest`.

Lo interesante hasta el momento es comprender a grandes rasgos qué es lo que hemos hecho: comunicar con el servidor cuando se produce un evento y obtener respuesta del servidor en segundo plano sin necesidad de hacer una recarga de página completa.

Todavía nos quedan muchas cosas por aclarar y las iremos explicando poco a poco.

## EJERCICIO

Tomando como punto de partida el código de ejemplo que hemos visto en esta entrega, realiza las modificaciones oportunas en los archivos php y html para:

- a) Definir una url asociada a cada ciudad (usa la url de la ciudad en wikipedia, por ejemplo para la ciudad Medellín será <https://es.wikipedia.org/wiki/Medell%C3%ADn>)
- b) Hacer que cuando se elige un país en el combobox, cada ciudad aparezca con el link a su url en wikipedia y que si se pulsa en ese link se abra una nueva ventana en el navegador mostrando la url correspondiente.

Para comprobar si tus respuestas son correctas puedes consultar en los foros [aprenderaprogramar.com](http://aprenderaprogramar.com).

**Próxima entrega:** CU01207F

**Acceso al curso completo** en [aprenderaprogramar.com](http://aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:

[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=83&Itemid=212](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=83&Itemid=212)