	UNIVERSIDAD DON BOSCO **FACULTAD DE INGENIERÍA** **COMPUTACIÓN**
CICLO 2 **2020**	*GUÍA DE LABORATORIO #11* Nombre de la Práctica: **Introducción a AJAX** Lugar de Ejecución: **Centro de Cómputo** Tiempo Estimado: **2 horas con 30 minutos** MATERIA: **Lenguajes Interpretados en el Cliente**

I. OBJETIVOS

Que el estudiante:

- Adquiera dominio en la técnica básica de desarrollo de aplicaciones web con AJAX en el lado del cliente.
- Pueda crear código JavaScript necesario para crear un objeto XMLHttpRequest para realizar peticiones asíncronas al servidor; independientemente del navegador que utilice el usuario.
- Maneje los métodos y propiedades del objeto XMLHttpRequest para trabajar con peticiones asíncronas al servidor web.

II. INTRODUCCIÓN TEÓRICA

El término AJAX es un acrónimo de Asynchronous JavaScript And XML, que se puede traducir como "JavaScript y XML Asíncrono". Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

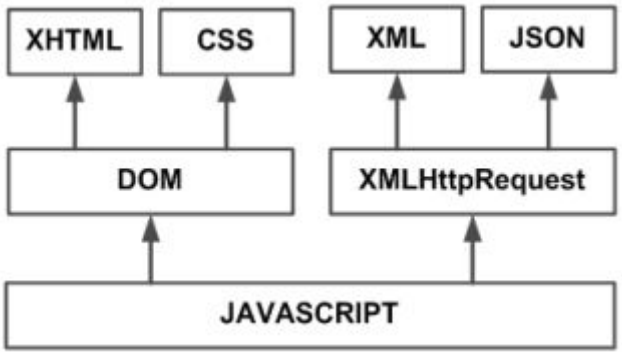


Figura 1.1 Tecnologías agrupadas bajo el concepto de AJAX

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

En el siguiente esquema, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX:

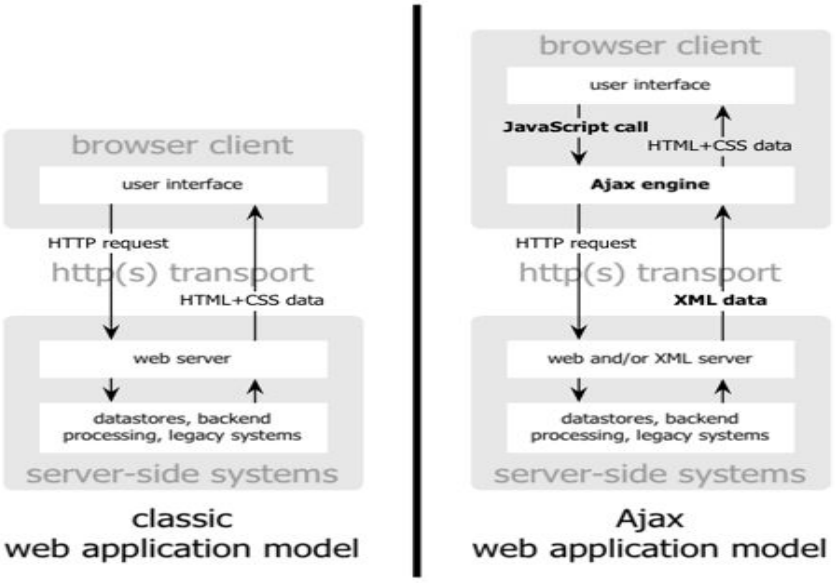


Figura 1.2 Comparación gráfica del modelo tradicional de aplicación web y del nuevo modelo propuesto por AJAX. (Imagen original creada por Adaptive Path y utilizada con su permiso)

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

Métodos y propiedades del objeto XMLHttpRequest

El objeto XMLHttpRequest posee muchas otras propiedades y métodos diferentes a las manejadas por la primera aplicación de AJAX. A continuación se incluye la lista completa de todas las propiedades y métodos del objeto y todos los valores numéricos de sus propiedades.

Las propiedades definidas para el objeto XMLHttpRequest son:

Propiedad	Descripción
readyState	Valor numérico (entero) que almacena el estado de la petición
responseText	El contenido de la respuesta del servidor en forma de cadena de texto
responseXML	El contenido de la respuesta del servidor en formato XML. El objeto devuelto se puede procesar como un objeto DOM
status	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.)
statusText	El código de estado HTTP devuelto por el servidor en forma de cadena de texto: "OK", "Not Found", "Internal Server Error", etc.

Los valores definidos para la propiedad readyState son los siguientes:

Valor	Descripción
0	No inicializado (objeto creado, pero no se ha invocado el método open)
1	Cargando (objeto creado, pero no se ha invocado el método send)
2	Cargado (se ha invocado el método send, pero el servidor aún no ha respondido)
3	Interactivo (se han recibido algunos datos, aunque no se puede emplear la propiedadresponseText)
4	Completo (se han recibido todos los datos de la respuesta del servidor)

Los métodos disponibles para el objeto XMLHttpRequest son los siguientes:

Método	Descripción
abort()	Detiene la petición actual
getAllResponseHeaders()	Devuelve una cadena de texto con todas las cabeceras de la respuesta del servidor
getResponseHeader("cabecera")	Devuelve una cadena de texto con el contenido de la cabecera solicitada
onreadystatechange	Responsable de manejar los eventos que se producen. Se invoca cada vez que se produce un cambio en el estado de la petición HTTP. Normalmente es una referencia a una función JavaScript
open("metodo", "url")	Establece los parámetros de la petición que se realiza al servidor. Los parámetros necesarios son el método HTTP empleado y la URL destino (puede indicarse de forma absoluta o relativa)
send(contenido)	Realiza la petición HTTP al servidor
setRequestHeader("cabecera", "valor")	Permite establecer cabeceras personalizadas en la petición HTTP. Se debe invocar el método open() antes quesetRequestHeader()

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]);

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.

No obstante, las peticiones síncronas son justamente contrarias a la filosofía de AJAX. El motivo es que una petición síncrona *congela* el navegador y no permite al usuario realizar ninguna acción hasta que no se haya recibido la respuesta completa del servidor. La sensación que provoca es que el navegador se ha *colgado* por lo que no se recomienda el uso de peticiones síncronas salvo que sea imprescindible.

Los últimos dos parámetros opcionales permiten indicar un nombre de usuario y una contraseña válidos para acceder al recurso solicitado.

Por otra parte, 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 anull. En otro caso, se puede indicar como parámetro una cadena de texto, un array de bytes o un objeto XML DOM.

Ejemplo práctico de petición por método GET

Vamos a ver un ejemplo sencillo para entender mejor la potencia del objeto XMLHttpRequest:

```
<html>
<head>
<title>AJAX</title>
<script type="text/javascript">
window.onload = function() {
document.getElementById('boton').onclick = enviarPeticion;
function enviarPeticion() {
 var peticion;
 if(window.XMLHttpRequest) {
    peticion = new XMLHttpRequest();
 }
 else if(window.ActiveXObject) {
    peticion = new ActiveXObject('Microsoft.XMLHTTP');
 }
 peticion.onreadystatechange = holaMundo;
 peticion.open('GET','http://localhost/Ajax/datos.txt');
 peticion.send(null);
 function holaMundo() {
   if(peticion.readyState==4)
   { if(peticion.status==200) {
        alert(peticion.responseText);
     }
   }
   else {
      document.getElementById('resultado').innerHTML = 'Esperando...';
   }
 }
}
}
</script>
</head>
<body>
<input type="button" id="boton" value="Enviar peticion" />
<p id="resultado"></p>
</body>
</html>
```

El archivo datos.txt del servidor es tan simple como este:

LIC

Ejemplo práctico de petición por método POST

Ahora vamos a ver un ejemplo para enviar una petición al servidor con parámetros mediante el método POST de HTTP.

```
<script type="text/javascript">
 window.onload = function() {
    document.getElementById('boton').onclick = enviarPeticion;
    function enviarPeticion() {
       var peticion;
       if(window.XMLHttpRequest) {
          peticion = new XMLHttpRequest();
       }
       else if(window.ActiveXObject) {
          peticion = new ActiveXObject('Microsoft.XMLHTTP');
       }
       peticion.onreadystatechange = holaUsuario;
       peticion.open('POST','http://localhost/pruebas/ajaxPost.php');
       peticion.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
       var nombre = document.getElementById('nombre');
       var apellidos = document.getElementById('apellidos')
       var informacion = "nombre=" + encondeURIComponent(nombre.value) +
                         "&apellidos=" + encondeURIComponent(apellidos.value);
       peticion.send(informacion);
```

```
    function holaUsuario() {
      if(peticion.readyState==4)
      { if(peticion.status==200) {
          alert(peticion.responseText);
        }
      }
      else {
         document.getElementById('resultado').innerHTML = 'Esperando...';
      }
    }
  }
}
</script>
```

III. MATERIAL Y EQUIPO

Para la realización de la guía de práctica se requerirá lo siguiente:

No.	Requerimiento	Cantidad
1	Guía de práctica #11: Introducción a AJAX	1
2	Computadora con editor HTML instalado y navegadores	1
3	Memoria USB.	1

IV. PROCEDIMIENTO

Ejercicio #1: El siguiente ejemplo ilustra cómo utilizar AJAX para generar ayudas contextuales al estilo Tooltip, al colocar el puntero del ratón sobre cada enlace presente en una página que apunta a un recurso en el servidor. La ayuda contextual muestra información del archivo obtenida directamente de las cabeceras HTTP que son utilizadas por el navegador para comunicarse con el servidor web. Esto se hace a través de AJAX mediante peticiones HTTP específicas que obtienen la información del recurso solicitado y luego se construye con esta información el tooltip correspondiente a cada enlace. Se requieren varias clases JavaScript que se incluyen en la sección HEAD del documento.

Guión 1: contenidoslic.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Enlaces</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/fonts.css" />
    <link rel="stylesheet" href="css/tooltip.css" />
    <script src="js/http.js"></script>
    <script src="js/tooltip.js"></script>
    <script src="js/geometry.js"></script>
    <script src="js/linkdetails.js"></script>
</head>
<body>
<header>
    <h1>Lenguajes Interpretados en el Cliente</h1>
</header>
<section>
<article id="content">
    <p>
        En el presente curso de la materia se van a desarrollar los siguientes
        contenidos:
    </p>
    <ol>
        <li>
            <a href="data/guia1LIC.docx" target="_blank">
              Fundamentos de páginas web dinámicas en el cliente con JavaScript.
            </a>
        </li>
        <li>
```

```
            <a href="data/tutorial-GitHub-LIC.flv" target="_blank">
                Plataformas de desarrollo colaborativo.
            </a>
        </li>
        <li>
                        <a  href="data/clase03-Sentencias-repetitivas-y-matrices.pdf"
target="_blank">
                Sentencias repetitivas y matrices.
            </a>
        </li>
        <li>
            <a href="data/clase04-Funciones.pptx" target="_blank">
                Funciones en JavaScript.
            </a>
        </li>
    </ol>
</article>
</section>
</body>
</html>
```

Guión 2: http.js

```
/**********************************************************************
 * http.js: utilities for scripted HTTP requests                      *
 *                                                                    *
 * From the book JavaScript: The Definitive Guide, 5th Edition,       *
 * by David Flanagan. Copyright 2006 O'Reilly Media,                  *
 * Inc. (ISBN: 0596101996)                                            *
 **********************************************************************/

// Asegurarse que no ha sido cargado el objeto HTTP
var HTTP;
if (HTTP && (typeof HTTP != "object" || HTTP.NAME))
    throw new Error("Namespace 'HTTP' already exists");

// Crear el objeto HTTP como un espacio de nombres y
// especificar alguna meta-información
HTTP = {};
// El nombre de este espacio de nombres
HTTP.NAME = "HTTP";
// La versión del espacio de nombres
// en este caso la versión del protocolo HTTP
HTTP.VERSION = 1.2;

// Crear un listado de los distintos objetos XMLHttpRequest
// de acuerdo al navegador que se esté utilizando
HTTP._factories = [
    function() {
        return new XMLHttpRequest();
    },
    function() {
        return new ActiveXObject("Msxml2.XMLHTTP");
    },
    function() {
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
];

// Si se encuentra un objeto XMLHttpRequest que funciona correctamente
// se almacena en HTTP._factory
HTTP._factory = null;

/*
 Crear y retornar un objeto XMLHttpRequest.
 La primera vez que invocamos al objeto XMLHttpRequest
 probamos con toda la lista de fabricantes de navegadores
 definidos en HTTP._factories hasta que encontramos uno
```

```
 que retorne un valor nonull sin lanzar una excepción.
 Una vez que se retorna un objeto XMLHttpRequest que funcione
 se almacena para su futuro uso.
 */
HTTP.newRequest = function() {
    if (HTTP._factory != null) return HTTP._factory();
    for(var i = 0; i < HTTP._factories.length; i++) {
        try {
            var factory = HTTP._factories[i];
            var request = factory();
            if (request != null) {
                HTTP._factory = factory;
                return request;
            }
        }
        catch(e) {
            continue;
        }
    }

    // If we get here, none of the factory candidates succeeded,
    // so throw an exception now and for all future calls.
    HTTP._factory = function() {
        throw new Error("XMLHttpRequest not supported");
    }
    HTTP._factory(); // Throw an error
}

/**
 * Use XMLHttpRequest to fetch the contents of the specified URL using
 * an HTTP GET request.  When the response arrives, pass it (as plain
 * text) to the specified callback function.
 *
 * This function does not block and has no return value.
 */
HTTP.getText = function(url, callback) {
    var request = HTTP.newRequest();
    request.onreadystatechange = function() {
        if (request.readyState == 4 && request.status == 200)
            callback(request.responseText);
    }
    request.open("GET", url);
    request.send(null);
};

/**
 * Use XMLHttpRequest to fetch the contents of the specified URL using
 * an HTTP GET request.  When the response arrives, pass it (as a parsed
 * XML Document object) to the specified callback function.
 *
 * This function does not block and has no return value.
 */
HTTP.getXML = function(url, callback) {
    var request = HTTP.newRequest();
    request.onreadystatechange = function() {
        if (request.readyState == 4 && request.status == 200)
            callback(request.responseXML);
    }
    request.open("GET", url);
    request.send(null);
};

/**
 * Use an HTTP HEAD request to obtain the headers for the specified URL.
 * When the headers arrive, parse them with HTTP.parseHeaders() and pass the
 * resulting object to the specified callback function. If the server returns
 * an error code, invoke the specified errorHandler function instead.  If no
 * error handler is specified, pass null to the callback function.
```

```
 */
HTTP.getHeaders = function(url, callback, errorHandler) {
    var request = HTTP.newRequest();
    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            if (request.status == 200) {
                callback(HTTP.parseHeaders(request));
            }
            else {
                if (errorHandler) errorHandler(request.status,
                                               request.statusText);
                else callback(null);
            }
        }
    }
    request.open("HEAD", url);
    request.send(null);
};

/**
 * Parse the response headers from an XMLHttpRequest object and return
 * the header names and values as property names and values of a new object.
 */
HTTP.parseHeaders = function(request) {
    var headerText = request.getAllResponseHeaders();  // Text from the server
    var headers = {}; // This will be our return value
    var ls = /^\s*/;  // Leading space regular expression
    var ts = /\s*$/;  // Trailing space regular expression

    // Break the headers into lines
    var lines = headerText.split("\n");
    // Loop through the lines
    for(var i = 0; i < lines.length; i++) {
        var line = lines[i];
        if (line.length == 0) continue;  // Skip empty lines
        // Split each line at first colon, and trim whitespace away
        var pos = line.indexOf(':');
        var name = line.substring(0, pos).replace(ls, "").replace(ts, "");
        var value = line.substring(pos+1).replace(ls, "").replace(ts, "");
        // Store the header name/value pair in a JavaScript object
        headers[name] = value;
    }
    return headers;
};

/**
 * Send an HTTP POST request to the specified URL, using the names and values
 * of the properties of the values object as the body of the request.
 * Parse the server's response according to its content type and pass
 * the resulting value to the callback function.  If an HTTP error occurs,
 * call the specified errorHandler function, or pass null to the callback
 * if no error handler is specified.
 **/
HTTP.post = function(url, values, callback, errorHandler) {
    var request = HTTP.newRequest();
    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            if (request.status == 200) {
                callback(HTTP._getResponse(request));
            }
            else {
                if (errorHandler) errorHandler(request.status,
                                               request.statusText);
                else callback(null);
            }
        }
    }
```

```
    request.open("POST", url);
    // This header tells the server how to interpret the body of the request
    request.setRequestHeader("Content-Type",
                             "application/x-www-form-urlencoded");
    // Encode the properties of the values object and send them as
    // the body of the request.
    request.send(HTTP.encodeFormData(values));
};

/**
 * Encode the property name/value pairs of an object as if they were from
 * an HTML form, using application/x-www-form-urlencoded format
 */
HTTP.encodeFormData = function(data) {
    var pairs = [];
    var regexp = /%20/g; // A regular expression to match an encoded space

    for(var name in data) {
        var value = data[name].toString();
        // Create a name/value pair, but encode name and value first
        // The global function encodeURIComponent does almost what we want,
        // but it encodes spaces as %20 instead of as "+". We have to
        // fix that with String.replace()
        var pair = encodeURIComponent(name).replace(regexp,"+") + '=' +
            encodeURIComponent(value).replace(regexp,"+");
        pairs.push(pair);
    }

    // Concatenate all the name/value pairs, separating them with &
    return pairs.join('&');
};

/**
 * Parse an HTTP response based on its Content-Type header
 * and return the parsed object
 */
HTTP._getResponse = function(request) {
    // Check the content type returned by the server
    switch(request.getResponseHeader("Content-Type")) {
    case "text/xml":
        // If it is an XML document, use the parsed Document object
        return request.responseXML;

    case "text/json":
    case "application/json":
    case "text/javascript":
    case "application/javascript":
    case "application/x-javascript":
        // If the response is JavaScript code, or a JSON-encoded value,
        // call eval() on the text to "parse" it to a JavaScript value.
        // Note: only do this if the JavaScript code is from a trusted server!
        return eval(request.responseText);

    default:
        // Otherwise, treat the response as plain text and return as a string
        return request.responseText;
    }
};

/**
 * Send an HTTP GET request for the specified URL.  If a successful
 * response is received, it is converted to an object based on the
 * Content-Type header and passed to the specified callback function.
 * Additional arguments may be specified as properties of the options object.
 *
 * If an error response is received (e.g., a 404 Not Found error),
 * the status code and message are passed to the options.errorHandler
 * function.  If no error handler is specified, the callback
```

```
 * function is called instead with a null argument.
 *
 * If the options.parameters object is specified, its properties are
 * taken as the names and values of request parameters.  They are
 * converted to a URL-encoded string with HTTP.encodeFormData() and
 * are appended to the URL following a '?'.
 *
 * If an options.progressHandler function is specified, it is
 * called each time the readyState property is set to some value less
 * than 4.  Each call to the progress handler function is passed an
 * integer that specifies how many times it has been called.
 *
 * If an options.timeout value is specified, the XMLHttpRequest
 * is aborted if it has not completed before the specified number
 * of milliseconds have elapsed.  If the timeout elapses and an
 * options.timeoutHandler is specified, that function is called with
 * the requested URL as its argument.
 **/
HTTP.get = function(url, callback, options) {
    var request = HTTP.newRequest();
    var n = 0;
    var timer;
    if (options.timeout)
        timer = setTimeout(function() {
                               request.abort();
                               if (options.timeoutHandler)
                                   options.timeoutHandler(url);
                           },
                           options.timeout);

    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            if (timer) clearTimeout(timer);
            if (request.status == 200) {
                callback(HTTP._getResponse(request));
            }
            else {
                if (options.errorHandler)
                    options.errorHandler(request.status,
                                         request.statusText);
                else callback(null);
            }
        }
        else if (options.progressHandler) {
            options.progressHandler(++n);
        }
    }

    var target = url;
    if (options.parameters)
        target += "?" + HTTP.encodeFormData(options.parameters)
    request.open("GET", target);
    request.send(null);
};

HTTP.getTextWithScript = function(url, callback) {
    // Create a new script element and add it to the document
    var script = document.createElement("script");
    document.body.appendChild(script);

    // Get a unique function name
    var funcname = "func" + HTTP.getTextWithScript.counter++;

    // Define a function with that name, using this function as a
    // convenient namespace.  The script generated on the server
    // invokes this function
    HTTP.getTextWithScript[funcname] = function(text) {
        // Pass the text to the callback function
```

```
        callback(text);
        // Clean up the script tag and the generated function
        document.body.removeChild(script);
        delete HTTP.getTextWithScript[funcname];
    }

    // Encode the URL we want to fetch and the name of the function
    // as arguments to the jsquoter.php server-side script.  Set the src
    // property of the script tag to fetch the URL
    script.src = "jsquoter.php" +
                 "?url=" + encodeURIComponent(url) + "&func=" +
                 encodeURIComponent("HTTP.getTextWithScript." + funcname);
}

// Usamos esto para generar una única función de retrollamada
// en caso de que haya más de una solicitud al mismo tiempo.
HTTP.getTextWithScript.counter = 0;
```

Guión 3: linkdetails.js

```
/*********************************************************************
 * Archivo: linkdetails.js                                           *
 * Descripción: Este módulo de JavaScript no obstruviso añade        *
 *              manejadores de evento a los enlaces presentes en     *
 *              el documento de modo que muestren textos de ayuda    *
 *              cuando el ratón se posicione encima de éstos por     *
 *              un tiempo de medio segundo.                          *
 *              Si el enlace apunta a un documento en el mismo       *
 *              servidor el texto de ayuda incluirá el tipo, tamaño  *
 *              y fecha obtenidos con una petición XMLHttpRequest    *
 *                                                                   *
 * Information: Este módulo requiere los módulos tooltip.js,         *
 *              HTTP.js, y geometry.js que se proporcionan           *
 *              en la carpeta de recursos de la guía.                *
 *********************************************************************/
(function() {  // Función anómina que engloba toda la lógica del script
    // Creación del objeto Tooltip que usaremos
    var tooltip = new Tooltip();

    // Asociar los manejadores de evento del evento load para que
    // se llame a la función init() al cargarse el documento
    if(window.addEventListener){
        window.addEventListener("load", init, false);
    }
    else if (window.attachEvent){
        window.attachEvent("onload", init);
    }

    // Función init() que será llamada cuando el documento haya cargado
    function init() {
        var links = document.getElementsByTagName('a');
        // Ciclo for con el que se recorren todos los enlaces de la página web
        // y se les agrega el manejador de evento correspondiente
        for(var i = 0; i < links.length; i++){
            if(links[i].href) addTooltipToLink(links[i]);
        }
    }

    // Función que agrega los manejadores de eventos a los enlaces
    // enviados como argumentos
    function addTooltipToLink(link) {
        // Agregando los manejadores de eventos
        if(link.addEventListener) {  // Técnica estándar
            link.addEventListener("mouseover", mouseover, false);
            link.addEventListener("mouseout", mouseout, false);
        }
        else if(link.attachEvent) {  // Técnica específica para IE
            link.attachEvent("onmouseover", mouseover);
```

```
            link.attachEvent("onmouseout", mouseout);
        }

        // Se usará esta variable con setTimeout/clearTimeout
        var timer;

        function mouseover(event) {
            // Menejo de evento de ratón para cualquier navegador
            var e = event || window.event;
            // Obtener las posiciones específicas del ratón
            // cuando se posicione encima de los enlaces convirtiendo
            // a coordenadas del documento añadiendo un desplazamiento (offset)
            var x = e.clientX + Geometry.getHorizontalScroll() + 20;
            var y = e.clientY + Geometry.getVerticalScroll() + 12;

            // Si existe algún tooltip pendiente de cancelar se limpia
            if(timer) window.clearTimeout(timer);

               // Programar el tooltip para que aparezca hasta transcurrido medio
segundo
            timer = window.setTimeout(showTooltip, 500);

            function showTooltip() {
                // If it is an HTTP link, and if it is from the same host
                // as this script is, we can use XMLHttpRequest
                // to get more information about it.
              if(link.protocol == "http:" && link.host == location.host) {
            // Make an XMLHttpRequest for the headers of the link
            HTTP.getHeaders(link.href, function(headers) {
                // Usar encabezados HTTP para construir
                // la cadena de texto con la información
                var tip = "URL: " + link.href + "<br>" +
                          "Type: " + headers["Content-Type"] + "<br>" +
                          "Size: " + headers["Content-Length"] + "<br>" +
                          "Date: " + headers["Last-Modified"];
                          // Desplegar el texto anterior como un tooltip
                          tooltip.show(tip, x, y);
                    });

                }
                else {
                    // Otherwise, if it is an offsite link, the
                    // tooltip is just the URL of the link
                    tooltip.show("URL: " + link.href, x, y);
                }
            }
        }

        function mouseout(e) {
            // Cuando el ratón se retire de encima del enlace, clear any
                  // limpiar cualquier tooltip pendiente o esconderlo si se está
mostrando
            if (timer) window.clearTimeout(timer);
            timer = null;
            tooltip.hide();
        }
    }
})();
```

Guión 4: tooltip.js

```
/**********************************************************************
 * Tooltip.js: simple CSS tooltips with drop shadows.                 *
 *                                                                    *
 * Este módulo define una clase Tooltip creando un objeto Tooltip     *
 * con el constructor Tooltip(), haciéndolo visible con una llamada   *
 * al método show() y ocultándolo con una llamada al método hide().   *
 *                                                                    *
```

```
 * Debe notar que necesitará crear los estilos apropiados para       *
 * mostrar u ocultar apropiadamente el elemento Tooltip. Los         *
 * siguientes estilos muestran un ejemplo de cómo hacerlo.           *
 *   .tooltipShadow {                                                *
 *      background: url(shadow.png);  /* translucent shadow * /      *
 *   }                                                               *
 *                                                                   *
 *   .tooltipContent {                                               *
 *      left: -4px; top: -4px;        /* how much of the shadow shows * /
 *      background-color: #ff0;       /* yellow background * /
 *      border: solid black 1px;      /* thin black border * /
 *      padding: 5px;                 /* spacing between text and border * /
 *      font: bold 10pt sans-serif;   /* small bold font * /
 *   }                                                               *
 *                                                                   *
 * En navegadores que soportan imágenes PNP transparentes es posible *
 * mostrar sombras con transparencia. Otros navegadores deberán usar *
 * colores sólidos o simular transparencia con un GIF interpolado    *
 * alterne píxeles sólidos y transparentes.                          *
 *******************************************************************/
// Método constructor para la clase Tooltip
function Tooltip() {
    this.tooltip = document.createElement("div"); // create div for shadow
    this.tooltip.style.position = "absolute";     // absolutely positioned
    this.tooltip.style.visibility = "hidden";     // starts off hidden
    this.tooltip.className = "tooltipShadow";     // so we can style it

    this.content = document.createElement("div"); // create div for content
    this.content.style.position = "relative";     // relatively positioned
    this.content.className = "tooltipContent";    // so we can style it

    this.tooltip.appendChild(this.content);       // add content to shadow
}

// Establecer el contenido y posición del Tooltip y, luego, mostrarlo
Tooltip.prototype.show = function(text, x, y) {
    this.content.innerHTML = text;             // Set the text of the tooltip.
    this.tooltip.style.left = x + "px";        // Set the position.
    this.tooltip.style.top = y + "px";
    this.tooltip.style.visibility = "visible"; // Make it visible.

    // Add the tooltip to the document if it has not been added before
    if (this.tooltip.parentNode != document.body)
        document.body.appendChild(this.tooltip);
};

// Esconder el Tooltip
Tooltip.prototype.hide = function() {
    this.tooltip.style.visibility = "hidden";  // Hacerlo invisible.
};
```

Guión 5: geometry.js

```
/*********************************************************
 * Geometry.js: portable functions for querying window    *
 * and document geometry.                                 *
 * This module defines functions for querying window and  *
 * document geometry.                                     *
 *                                                        *
 * getWindowX/Y(): return the position of the window on   *
 * the screen getViewportWidth/Height(): return the size  *
 * of the browser viewport area.                          *
 * getDocumentWidth/Height(): return the size of the      *
 * document.                                              *
 * getHorizontalScroll(): return the position of the      *
 * horizontal scrollbar.                                  *
 * getVerticalScroll(): return the position of the        *
 * vertical scrollbar.                                    *
```

```
 *                                                          *
 * Note that there is no portable way to query the overall*
 * size of the                                              *
 * browser window, so there are no getWindowWidth/Height()*
 * functions.                                               *
 *                                                          *
 * IMPORTANT: This module must be included in the <body>  *
 * of a document instead of the <head> of the document.   *
 ************************************************************/
var Geometry = {};

if (window.screenLeft) { // IE and others
    Geometry.getWindowX = function() { return window.screenLeft; };
    Geometry.getWindowY = function() { return window.screenTop; };
}
else if (window.screenX) { // Firefox and others
    Geometry.getWindowX = function() { return window.screenX; };
    Geometry.getWindowY = function() { return window.screenY; };
}

if (window.innerWidth) { // All browsers but IE
    Geometry.getViewportWidth = function() { return window.innerWidth; };
    Geometry.getViewportHeight = function() { return window.innerHeight; };
    Geometry.getHorizontalScroll = function() { return window.pageXOffset; };
    Geometry.getVerticalScroll = function() { return window.pageYOffset; };
}
else if (document.documentElement && document.documentElement.clientWidth) {
    // These functions are for IE6 when there is a DOCTYPE
    Geometry.getViewportWidth =
        function() { return document.documentElement.clientWidth; };
    Geometry.getViewportHeight =
        function() { return document.documentElement.clientHeight; };
    Geometry.getHorizontalScroll =
        function() { return document.documentElement.scrollLeft; };
    Geometry.getVerticalScroll =
        function() { return document.documentElement.scrollTop; };
}
else if (document.body.clientWidth) {
    // These are for IE4, IE5, and IE6 without a DOCTYPE
    Geometry.getViewportWidth =
        function() { return document.body.clientWidth; };
    Geometry.getViewportHeight =
        function() { return document.body.clientHeight; };
    Geometry.getHorizontalScroll =
        function() { return document.body.scrollLeft; };
    Geometry.getVerticalScroll =
        function() { return document.body.scrollTop; };
}

// These functions return the size of the document.  They are not window
// related, but they are useful to have here anyway.
if (document.documentElement && document.documentElemnet.scrollWidth) {
    Geometry.getDocumentWidth =
        function() { return document.documentElement.scrollWidth; };
    Geometry.getDocumentHeight =
        function() { return document.documentElement.scrollHeight; };
}
else if (document.body.scrollWidth) {
    Geometry.getDocumentWidth =
        function() { return document.body.scrollWidth; };
    Geometry.getDocumentHeight =
        function() { return document.body.scrollHeight; };
}
```

Guión 6: fonts.css

```
@font-face {
    font-family: 'Tenderness';
```

```
    src: url('../fonts/tenderness.eot');
    src: url('../fonts/tenderness.eot?#iefix') format('embedded-opentype'),
         url('../fonts/tenderness.woff') format('woff'),
         url('../fonts/tenderness.ttf') format('truetype'),
         url('../fonts/tenderness.svg#Tenderness') format('svg');
    font-weight: normal;
    font-style: normal;
    text-rendering: optimizeLegibility;
}
```

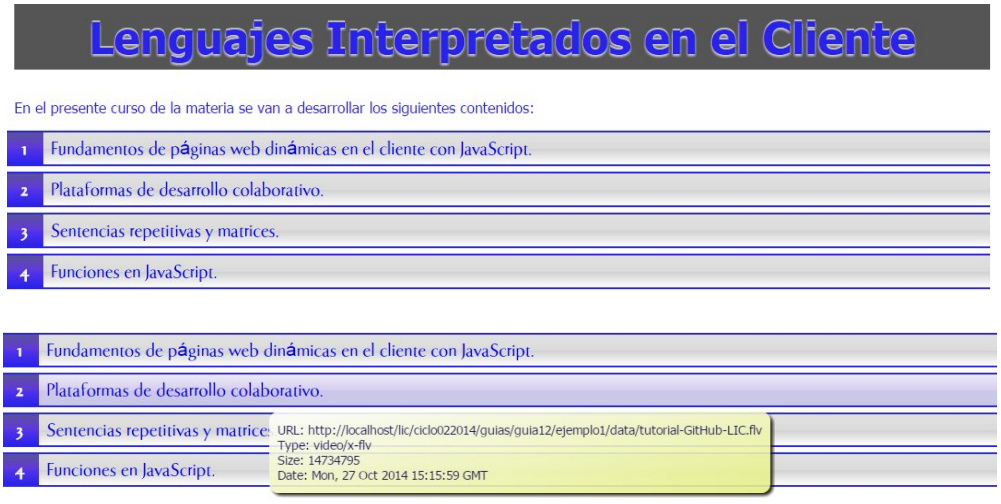
Guión 7: tooltip.css

```
* {
    margin: 0;
    padding: 0;
}

html, body {
    height: 100%;
    outline: none;
}

header h1 {
    background-color: #555;
    color: #2a23eb;
    font: Bold 3em Tahoma,Helvetica,"Liberation Sans";
    line-height: 1.5;
    text-align: center;
    text-shadow: 0px -2px 0px #bbb,
                 0px 2px 3px #ddd;
}

.tooltipShadow {
    /* IE10 Consumer Preview */
    background-image: -ms-linear-gradient(top left, #FFFFFF 0%, #E4EF81 100%);
    /* Mozilla Firefox */
    background-image: -moz-linear-gradient(top left, #FFFFFF 0%, #E4EF81 100%);
    /* Opera */
    background-image: -o-linear-gradient(top left, #FFFFFF 0%, #E4EF81 100%);
    /* Webkit (Safari/Chrome 10) */
          background-image:  -webkit-gradient(linear,  left  top,  right  bottom,
color-stop(0, #FFFFFF), color-stop(1, #E4EF81));
    /* Webkit (Chrome 11+) */
    background-image: -webkit-linear-gradient(top left, #FFFFFF 0%, #E4EF81 100%);
    /* W3C Markup, IE10 Release Preview */
    background-image: linear-gradient(to bottom right, #FFFFFF 0%, #E4EF81 100%);
    -webkit-border-radius: 10px;
    -moz-border-radius: 10px;
    -ms-border-radius: 10px;
    -o-border-radius: 10px;
    border-radius: 10px;
    -webkit-box-shadow: 3px 3px 4px 0px rgba(26,24,7,1);
    -moz-box-shadow: 3px 3px 4px 0px rgba(26,24,7,1);
    -ms-box-shadow: 3px 3px 4px 0px rgba(26,24,7,1);
    -o-box-shadow: 3px 3px 4px 0px rgba(26,24,7,1);
    box-shadow: 3px 3px 4px 0px rgba(26,24,7,1);
    color: #080248;
    font: Normal 0.85em Tahoma,Helvetica,"Liberation Sans";
    /* IE 8 */
    -ms-filter: "progid:DXImageTransform.Microsoft.Alpha(Opacity=90)";
    /* IE 5-7 */
    filter: alpha(opacity=90);
    /* Netscape */
    -moz-opacity: 0.9;
    /* Webkit */
    -webkit-opacity: 0.9;
    /* Safari 1.x */
    -khtml-opacity: 0.9;
```

```
    /* Good browsers */
    opacity: 0.9;
    padding: 12px 9px;
}

#content p {
    color: #2a23eb;
    font: Normal 1em Tahoma,Helvetica,"Liberation Sans";
    text-align: justify;
}

#content ol {
    counter-reset: li; /* Initiate a counter */
    margin-left: 0; /* Remove the default left margin */
    padding-left: 0; /* Remove the default left padding */
}

#content ol > li {
    position:relative; /* Create a positioning context */
     margin:0 0 6px 2em; /* Give each list item a left margin to make room for the
numbers */
    padding:4px 8px; /* Add some spacing around the content */
    list-style:none; /* Disable the normal item numbering */
    /* list-style-type: upper-roman; */
    border-bottom:2px solid #2a23eb;
    border-top:2px solid #2a23eb;
    background: rgba(226,226,226,1);
                background:    -moz-linear-gradient(top,    rgba(226,226,226,1)    0%,
rgba(219,219,219,1) 50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%);
           background:   -webkit-gradient(left   top,   left   bottom,   color-stop(0%,
rgba(226,226,226,1)),    color-stop(50%,    rgba(219,219,219,1)),    color-stop(51%,
rgba(209,209,209,1)), color-stop(100%, rgba(254,254,254,1)));
              background:    -webkit-linear-gradient(top,    rgba(226,226,226,1)    0%,
rgba(219,219,219,1) 50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%);
     background: -o-linear-gradient(top, rgba(226,226,226,1) 0%, rgba(219,219,219,1)
50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%);
                 background:    -ms-linear-gradient(top,    rgba(226,226,226,1)    0%,
rgba(219,219,219,1) 50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%);
              background:   linear-gradient(to   bottom,   rgba(226,226,226,1)   0%,
rgba(219,219,219,1) 50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%);
       filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#e2e2e2',
endColorstr='#fefefe', GradientType=0 );
}

#content ol > li:hover {
    background: rgba(235,233,249,1);
                background:    -moz-linear-gradient(top,    rgba(235,233,249,1)    0%,
rgba(216,208,239,1) 50%, rgba(206,199,236,1) 51%, rgba(201,200,230,1) 100%);
           background:   -webkit-gradient(left   top,   left   bottom,   color-stop(0%,
rgba(235,233,249,1)),    color-stop(50%,    rgba(216,208,239,1)),    color-stop(51%,
rgba(206,199,236,1)), color-stop(100%, rgba(201,200,230,1)));
              background:    -webkit-linear-gradient(top,    rgba(235,233,249,1)    0%,
rgba(216,208,239,1) 50%, rgba(206,199,236,1) 51%, rgba(201,200,230,1) 100%);
     background: -o-linear-gradient(top, rgba(235,233,249,1) 0%, rgba(216,208,239,1)
50%, rgba(206,199,236,1) 51%, rgba(201,200,230,1) 100%);
                 background:    -ms-linear-gradient(top,    rgba(235,233,249,1)    0%,
rgba(216,208,239,1) 50%, rgba(206,199,236,1) 51%, rgba(201,200,230,1) 100%);
              background:   linear-gradient(to   bottom,   rgba(235,233,249,1)   0%,
rgba(216,208,239,1) 50%, rgba(206,199,236,1) 51%, rgba(201,200,230,1) 100%);
       filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#ebe9f9',
endColorstr='#c9c8e6', GradientType=0 );
    color: #300690;
}

#content ol > li a {
    text-decoration: none;
    font-family: "Tenderness","Helvetica Neue", Arial, sans-serif;
    font-size: 1.3em;
```

```
}

#content ol > li:before {
    content:counter(li); /* Use the counter as content */
    counter-increment:li; /* Increment the counter by 1 */
    /* Position and style the number */
    position:absolute;
    top:-2px;
    left:-2em;
    -moz-box-sizing:border-box;
    -webkit-box-sizing:border-box;
    box-sizing:border-box;
    width:2em;
    /* Some space between the number and the content in browsers that support
       generated content but not positioning it (Camino 2 is one example) */
    margin-right:8px;
    padding:4px;
    border-bottom:2px solid #2a23eb;
    border-top:2px solid #2a23eb;
    color:#fff;
    background: rgba(91,83,156,1);
      background: -moz-linear-gradient(top, rgba(91,83,156,1) 0%, rgba(97,57,219,1)
48%, rgba(85,52,235,1) 51%, rgba(42,35,235,1) 100%);
          background: -webkit-gradient(left top, left bottom, color-stop(0%,
rgba(91,83,156,1)), color-stop(48%, rgba(97,57,219,1)), color-stop(51%,
rgba(85,52,235,1)), color-stop(100%, rgba(42,35,235,1)));
              background: -webkit-linear-gradient(top, rgba(91,83,156,1) 0%,
rgba(97,57,219,1) 48%, rgba(85,52,235,1) 51%, rgba(42,35,235,1) 100%);
       background: -o-linear-gradient(top, rgba(91,83,156,1) 0%, rgba(97,57,219,1)
48%, rgba(85,52,235,1) 51%, rgba(42,35,235,1) 100%);
      background: -ms-linear-gradient(top, rgba(91,83,156,1) 0%, rgba(97,57,219,1)
48%, rgba(85,52,235,1) 51%, rgba(42,35,235,1) 100%);
     background: linear-gradient(to bottom, rgba(91,83,156,1) 0%, rgba(97,57,219,1)
48%, rgba(85,52,235,1) 51%, rgba(42,35,235,1) 100%);
       filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#5b539c',
endColorstr='#2a23eb', GradientType=0 );
    font-weight: bold;
    font-family: "Tenderness","Helvetica Neue", Arial, sans-serif;
    font-size: 1.3em;
    text-align:center;
}

#content li ol,
#content li ul {
    margin-top:6px;
}

#content ol ol li:last-child {
    margin-bottom:0;
}

#content ol ol li:last-child a {
    text-decoration: none;
    font-family: "Tenderness","Helvetica Neue", Arial, sans-serif;
    font-size: 1.3em;
}
```

Resultado:



Ejercicio #2: El siguiente ejemplo muestra la realización de una página usando AJAX en donde se carga el contenido relacionado con el libro cuando se posiciona el puntero del ratón encima de la imagen de su portada. El contenido se carga en un elemento DIV haciendo una petición asíncrona al servidor justo en el momento que se coloca el puntero del ratón encima de la imagen agregando dinámicamente un manejador de evento al evento mouseover. Del mismo modo, se borra el contenido del DIV cuando se retira el puntero del ratón de encima de la portada de cada libro asociando al evento mouseout el manejador de evento que realiza esa tarea.

Guión 1: librosprogramacion.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Libros de programación web</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/books.css" />
    <script src="js/booksajax.js"></script>
</head>
<body>
<header>
    <h1>Libros de programación</h1>
</header>
<section>
<article>
<div id="covers">
       <img  src="img/css-tecnicas-profesionales.jpg"  id="csstecprof"  alt="Portada
Técnicas profesionales para el diseño moderno" />
    <img src="img/java8.jpg" id="java8" alt="Portada Java 8" />
     <img src="img/javascript-ninja.jpg" id="jsninja" alt="Portada JavaScript Ninja"
/>
    <img src="img/nodejs.jpg" id="nodejs" alt="Portada Node.js" />
     <img src="img/php-practico.jpg" id="phppract" alt="Portada Fundamentos de PHP
Práctico" />
     <img src="img/programacion-html5.jpg" id="proghtml5" alt="Portada Programación
HTML5" />
</div>
<div id="descriptions" class="content-box"></div>
</article>
</section>
</body>
</html>
```

Guión 2: booksajax.js

```
//Variable para el objeto XMLHttpRequest
var solicitudAsinc;

//Establecer los manejadores de evento para las imágenes con
//las portadas de los libros
```

```
function registrarManejadores(){
    var img;
    var contentbook;
    //Creando el contenido y borrándolos al producirse los eventos
    //mouseover y mouseout sobre las imágenes capturadas por su id
    //Primera imagen
    img = document.getElementById("csstecprof");
    if(img.addEventListener){
        img.addEventListener("mouseover", function(){
            obtenerContenido("css.html");
        }, false);
    }
    else if(img.attachEvent){
        img.attachEvent("onmouseover", function(){
            obtenerContenido("css.html");
        });
    }
    if(img.addEventListener){
        img.addEventListener("mouseout", borrarContenido, false);
    }
    else if(img.attachEvent){
       img.attachEvent("onmouseout", borrarContenido);
    }
    //Segunda imagen
    img = document.getElementById("java8");
    if(img.addEventListener){
        img.addEventListener("mouseover", function(){
            obtenerContenido("java8.html");
        }, false);
    }
    else if(img.attachEvent){
       img.attachEvent("mouseover", function(){
            obtenerContenido("java8.html");
       });
    }
    if(img.addEventListener){
        img.addEventListener("mouseout", borrarContenido, false);
    }
    else if(img.attachEvent){
        img.attachEvent("onmouseout", borrarContenido);
    }
    //Tercera imagen
    img = document.getElementById("jsninja");
    if(img.addEventListener){
        img.addEventListener("mouseover", function(){
            obtenerContenido("jsninja.html");
        }, false);
    }
    else if(img.attachEvent){
        img.attachEvent("onmouseover", function(){
            obtenerContenido("jsninja.html");
        });
    }
    if(img.addEventListener){
        img.addEventListener("mouseout", borrarContenido, false);
    }
    else if(img.attachEvent){
       img.attachEvent("onmouseout", borrarContenido);
    }
    //Cuarta imagen
    img = document.getElementById("nodejs");
    if(img.addEventListener){
        img.addEventListener("mouseover", function(){
            obtenerContenido("nodejs.html");
        }, false);
    }
    else if(img.attachEvent){
       img.attachEvent("onmouseover", function(){
```

```
            obtenerContenido("nodejs.html");
        });
    }
    if(img.addEventListener){
        img.addEventListener("mouseout", borrarContenido, false);
    }
    else if(img.attachEvent){
        img.attachEvent("onmouseout", borrarContenido);
    }
    //Quinta imagen
    img = document.getElementById("phppract");
    if(img.addEventListener){
        img.addEventListener("mouseover", function(){
            obtenerContenido("phppract.html");
        }, false);
    }
    else if(img.attachEvent){
        img.attachEvent("onmouseover", function(){
             obtenerContenido("phppract.html");
        });
    }
    if(img.addEventListener){
        img.addEventListener("mouseout", borrarContenido, false);
    }
    else if(img.attachEvent){
       img.attachEvent("onmouseout", borrarContenido);
    }
    //Sexta imagen
    img = document.getElementById("proghtml5");
    if(img.addEventListener){
        img.addEventListener("mouseover", function(){
            obtenerContenido("proghtml5.html");
        }, false);
    }
    else if(img.attachEvent){
        img.attachEvent("onmouseover", function(){
            obtenerContenido("proghtml5.html");
        });
    }
    if(img.addEventListener){
        img.addEventListener("mouseout", borrarContenido, false);
    }
    else if(img.attachEvent){
        img.attachEvent("onmouseout", borrarContenido);
    }
} //Fin de la función que registra eventos sobre las imágenes

function obtenerContenido(url){
    //Intentar crear objeto XMLHtttpRequest y realizar la petición
    try {
       //Crear objeto petición XMLHttpRequest
       //Cambiar esto por una función multinavegador para construir el objeto
XMLHttpRequest
       solicitudAsinc = new XMLHttpRequest();
       //Registrar el manejador de eventos
       if(solicitudAsinc.addEventListener){
                  solicitudAsinc.addEventListener("readystatechange", cambiarEstado,
false);
        }
        else if(solicitudAsinc.attachEvent){
            solicitudAsinc.attachEvent("onreadystatechange", cambiarEstado);
        }
        //Preparar la solicitud
        solicitudAsinc.open("GET", url, true);
        //Enviar la solicitud
        solicitudAsinc.send(null);
    }
    catch(exception){
```

```
        alert("No se procesó la petición AJAX");
    }
}

function borrarContenido(){
    var contenido = document.getElementById("descriptions");
    contenido.innerHTML = "";
}

function cambiarEstado(){
    var contenido;
    if(solicitudAsinc.readyState == 4 && solicitudAsinc.status == 200){
        contenido = document.getElementById("descriptions");
        //Coloca el contenido devuelto en la petición en el div descriptions
        contenido.innerHTML = solicitudAsinc.responseText;
    }
}

if(window.addEventListener){
    window.addEventListener("load", registrarManejadores, false);
}
else if(window.attachEvent){
    window.attachEvent("onload", registrarManejadores);
}
```

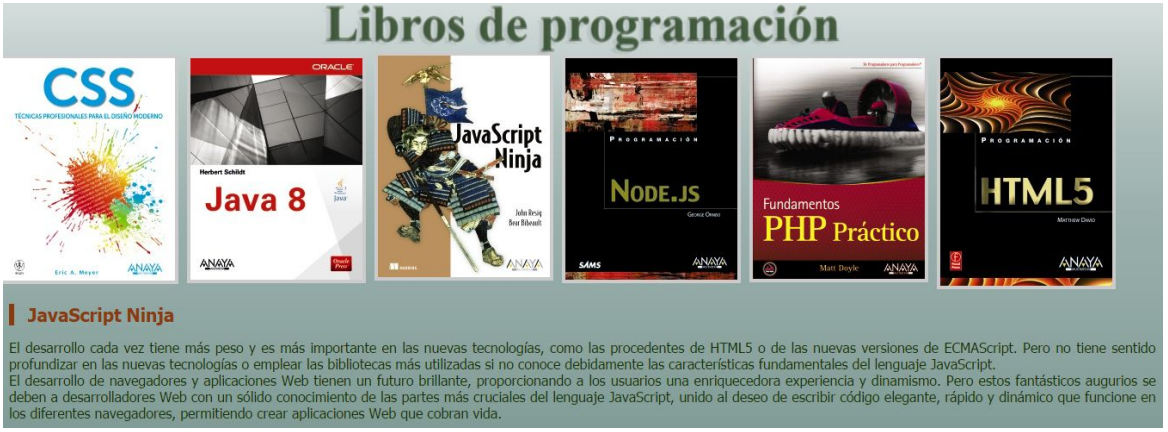
Guión 3: books.css

```
* {
    margin: 0;
    padding: 0;
}

html, body {
    height: 100%;
}

body {
    /* Old browsers */
    background: #d8e0de;
    /* FF3.6+ */
      background: -moz-linear-gradient(top, #d8e0de 0%, #aebfbc 22%, #99afab 40%,
#8ea6a2 59%, #829d98 75%, #4e5c5a 90%, #303d31 100%);
    /* Chrome,Safari4+ */
            background:   -webkit-gradient(linear,   left   top,   left   bottom,
color-stop(0%,#d8e0de),      color-stop(22%,#aebfbc),      color-stop(40%,#99afab),
color-stop(59%,#8ea6a2),      color-stop(75%,#829d98),      color-stop(90%,#4e5c5a),
color-stop(100%,#303d31));
    /* Chrome10+,Safari5.1+ */
        background:  -webkit-linear-gradient(top,  #d8e0de  0%,#aebfbc  22%,#99afab
40%,#8ea6a2 59%,#829d98 75%,#4e5c5a 90%,#303d31 100%);
    /* Opera 11.10+ */
     background: -o-linear-gradient(top, #d8e0de 0%,#aebfbc 22%,#99afab 40%,#8ea6a2
59%,#829d98 75%,#4e5c5a 90%,#303d31 100%);
    /* IE10+ */
     background: -ms-linear-gradient(top, #d8e0de 0%,#aebfbc 22%,#99afab 40%,#8ea6a2
59%,#829d98 75%,#4e5c5a 90%,#303d31 100%);
    /* W3C */
        background:  linear-gradient(to  bottom,  #d8e0de  0%,#aebfbc  22%,#99afab
40%,#8ea6a2 59%,#829d98 75%,#4e5c5a 90%,#303d31 100%);
    /* IE6-9 */
      filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#d8e0de',
endColorstr='#303d31',GradientType=0 );
    color: #163303;
    font-family: Tahoma,Helvetica,"Liberation Sans";
    font-size: 16px;
}

header h1 {
```

```
    color: transparent;
         filter:progid:DXImageTransform.Microsoft.DropShadow(color=#1E3C05,  offX=1,
offY=1);
    font-family: "Anton", serif;
    font-size: 3.6em;
    text-align: center;
    text-shadow: rgba(30, 60, 5, 0.35) 0 0px 0px,
                 rgba(0, 30, 0, 0.08) 0px 2px 2px,
                 rgba(0, 30, 0, 0.20) 0px 2px 1px,
                 rgba(0, 30, 0, 0.40) 0px 2px 1px,
                 rgba(0, 0, 0, 0.08) -5px 5px 2px;
}

#descriptions h3 {
    border-left: 6px solid #8A3C10;
    color: #8A3C10;
    font-size: 1.3em;
    margin-bottom: 16px;
    padding-left: 16px;
}

img{
    border: 4px solid #ccc;
    float: left;
    margin: 5px;
    -webkit-transition: margin 0.5s ease-out;
    -moz-transition: margin 0.5s ease-out;
    -o-transition: margin 0.5s ease-out;
}

img:hover {
    margin-top: 2px;
}

.content-box {
    padding: 12px 16px;
    text-align: justify;
    clear: both;
}
```

NOTA: Los archivos estáticos css.html, java8.html, jsninja.html, nodejs.html, phppract.html y proghtml5.html estarán disponibles en los recursos de descarga que se adjuntan en el sitio web junto con la guía de práctica.

Resultado:



Ejercicio #3: El siguiente ejemplo muestra cómo crear un menú de navegación que usa peticiones AJAX cada vez que se desea ir a uno de los enlaces. El contenido está disponible como contenido

HTML almacenado en archivos completamente estáticos que son cargados asíncronamente usando la propiedad responseText el objeto XMLHttpRequest.

Guión 1: navegacion.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title></title>
    <meta charset="utf-8" />
    <link rel="stylesheet"
href="http://fonts.googleapis.com/css?family=Anton:regular" />
    <link rel="stylesheet" href="css/browsers.css" />
    <script src="js/xmlhttp.js"></script>
    <script src="js/eventhandler.js"></script>
<body>
<header>
    <h1>Navegadores web</h1>
</header>
<section>
<article>
<div id="cssmenu">
       <ul>
           <li class="active">
               <span><a href="content1.html" id="1">Chrome</a></span>
           </li>
           <li>
               <span><a href="content2.html" id="2">Firefox</a></span>
           </li>
           <li>
               <span><a href="content3.html" id="3">Internet Explorer</a></span>
           </li>
           <li>
               <span><a href="content4.html" id="4">Opera</a></span>
           </li>
       </ul>
</div>
<div id="content"></div>
</body>
</article>
</section>
</html>
```

Guión 2: xmlhttp.js

```
//Crear una variable de Bool para comprobar si se está usando Internet Explorer.
var xmlhttp = false;

//Comprobar si se está usando IE.
try {
       //Si la versión de Internet Explorer es superior a la 5.0
       xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
       //Si no, utilizar el tradicional objeto ActiveX.
       try {
              //Si se está usando Internet Explorer.
              xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
       } catch (E) {
              //En caso contrario no se debe estar usando Internet Explorer.
              xmlhttp = false;
       }
}

//Si no se está usando IE, crear una instancia ActiveX del objeto.
if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
       xmlhttp = new XMLHttpRequest();
}

function makerequest(serverPage, objID) {
       var obj = document.getElementById(objID);
```

```
	xmlhttp.open("GET", serverPage);
	xmlhttp.onreadystatechange = function() {
		if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
			obj.innerHTML = xmlhttp.responseText;
		}
	}
	xmlhttp.send(null);
}
```

Guión 3: eventhandler.js

```
function init(){
    var links;
    var numlinks;
    links = document.getElementsByTagName("a");
    numlinks = links.length;
    for(var i=0; i<numlinks; i++){
        if(links[i].addEventListener){
            links[i].addEventListener("click", function(event){
             if(event.preventDefault){
                 event.preventDefault();
             }
             else if(event.stopPropagation){
                    event.stopPropagation();
               }
               else{
                   event.returnValue = false;
               }
               makerequest('content' + this.id + '.html','content');
               return false;
            }, false);
        }
        else if(links[i].attachEvent){
            links[i].attachEvent("onclick", function(event){
             if(event.preventDefault){
                 event.preventDefault();
             }
             else if(event.stopPropagation){
                    event.stopPropagation();
               }
               else{
                   event.returnValue = false;
               }
             makerequest('content' + this.id + '.html','content');
               return false;
            });
        }
    }
    makerequest('content1.html','content');
}

if(window.addEventListener){
    window.addEventListener("load", init, false);
}
else if(window.attachEvent){
    window.attachEvent("load", init);
}
```

Guión 4: browsers.css

```
@import url(http://fonts.googleapis.com/css?family=Open+Sans:700);

* {
    margin: 0;
    padding: 0;
}

html, body {
    height: 100%;
```

```
}

body {
    font-size: 16px;
    font-family: Arial,Helvetica,"Liberation Sans";
}

header {
    background-color: rgba(200,0,0,.5) !important;
    background-image: linear-gradient(0, rgba(250,110,90,.5) 50%, transparent 50%),
                                    linear-gradient(rgba(200,0,0,.5) 50%, transparent
50%)!important;
          background-image:  -webkit-linear-gradient(0,  rgba(250,110,90,.5)  50%,
transparent 50%),
                             -webkit-linear-gradient(rgba(200,0,0,.5) 50%, transparent
50%)!important;
     background-image: -moz-linear-gradient(0, rgba(250,110,90,.5) 50%, transparent
50%),
                           -moz-linear-gradient(rgba(200,0,0,.5) 50%, transparent 50%)
!important;
      background-image: -o-linear-gradient(0, rgba(250,110,90,.5) 50%, transparent
50%),
                            -o-linear-gradient(rgba(200,0,0,.5) 50%, transparent 50%)
!important;
    background-size: 50px 50px!important;
}

header h1 {
    font-family: "Anton",Helvetica,"Liberation Sans",serif;
    font-size: 4em;
    color: #990F0A;
    text-align: center;
    text-shadow: 5px 5px 0px #EEE, 7px 7px 0px #707070;
}

#cssmenu {
  background: #f96e5b;
  width: auto;
}

#cssmenu ul {
  list-style: none;
  margin: 0;
  padding: 0;
  line-height: 1;
  display: block;
  zoom: 1;
}

#cssmenu ul:after {
  content: " ";
  display: block;
  font-size: 0;
  height: 0;
  clear: both;
  visibility: hidden;
}

#cssmenu ul li {
  display: inline-block;
  padding: 0;
  margin: 0;
}

#cssmenu.align-right ul li {
  float: right;
}
```

```
#cssmenu.align-center ul {
  text-align: center;
}

#cssmenu ul li a {
  color: #ffffff;
  text-decoration: none;
  display: block;
  padding: 15px 25px;
  font-family: 'Open Sans', sans-serif;
  font-weight: 700;
  text-transform: uppercase;
  font-size: 14px;
  position: relative;
  -webkit-transition: color .25s;
  -moz-transition: color .25s;
  -ms-transition: color .25s;
  -o-transition: color .25s;
  transition: color .25s;
}

#cssmenu ul li a:hover {
  color: #333333;
}

#cssmenu ul li a:hover:before {
  width: 100%;
}

#cssmenu ul li a:after {
  content: "";
  display: block;
  position: absolute;
  right: -3px;
  top: 19px;
  height: 6px;
  width: 6px;
  background: #ffffff;
  opacity: .5;
}

#cssmenu ul li a:before {
  content: "";
  display: block;
  position: absolute;
  left: 0;
  bottom: 0;
  height: 3px;
  width: 0;
  background: #333333;
  -webkit-transition: width .25s;
  -moz-transition: width .25s;
  -ms-transition: width .25s;
  -o-transition: width .25s;
  transition: width .25s;
}

#cssmenu ul li.last > a:after,
#cssmenu ul li:last-child > a:after {
  display: none;
}

#cssmenu ul li.active a {
  color: #333333;
}

#cssmenu ul li.active a:before {
  width: 100%;
```

```
}

#cssmenu.align-right li.last > a:after,
#cssmenu.align-right li:last-child > a:after {
  display: block;
}

#cssmenu.align-right li:first-child a:after {
  display: none;
}

@media screen and (max-width: 768px) {
  #cssmenu ul li {
    float: none;
    display: block;
  }
  #cssmenu ul li a {
    width: 100%;
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
    border-bottom: 1px solid #fb998c;
  }
  #cssmenu ul li.last > a,
  #cssmenu ul li:last-child > a {
    border: 0;
  }
  #cssmenu ul li a:after {
    display: none;
  }
  #cssmenu ul li a:before {
    display: none;
  }
}

/* Estilos para el bloque de contenido */
.browser-box {
    padding: 16px 20px;
}

.browser-box h1 {
    font: Bold 2em Tahoma,Helvetica,"Liberation Sans";
    color: #930C0A;
    margin-bottom: 12px;
    text-align: center;
}

.browser-box p {
    font: Normal 0.9em Tahoma,Helvetica,"Liberation Sans";
    color: Brown;
    margin-bottom: 8px;
    text-align: justify;
}

#content img {
    border: 2px solid #d3bfbc;
    -moz-box-shadow: 4px 4px 5px #b68d4c;
    -webkit-box-shadow: 4px 4px 5px #b68d4c;
    box-shadow: 4px 4px 5px #b68d4c;
    -moz-border-radius: 25px;
    -webkit-border-radius: 25px;
    border-radius: 25px;
    float: left;
    margin: 0 15px 0 0;
}
```

NOTA: Los archivos estáticos content1.html, content2.html, content3.html y content4.html estarán disponibles en los recursos de descarga que se adjuntan en el sitio web junto con la guía de práctica.

Resultado:

Ejemplo #4: El siguiente ejemplo ilustra cómo utilizar peticiones asíncronas con AJAX para cargar contenido en un elemento DIV proveniente de un objeto JSON. Este objeto JSON está almacenado en un archivo con extensión JSON, que bien podría ser cualquier objeto JavaScript válido de forma declarativa, pero son declaración de variable, lo que significa que debe comenzar con llave de apertura ({) y terminar con llave de cierre (}). Para convertir el código del objeto JSON que es asumido como cadena cuando se pasa con el método responseText, debe utilizar el método parse() para convertirlo en objeto de JavaScript y así después, poder acceder a todas las propiedades definidas en el mismo.

Guión 1: enlacesjson.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Enlaces con JSON</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/linkbox.css" />
    <script src="js/linksjson.js"></script>
</head>
<body>
<header>
    <h1>Favoritos de la Liga de Campeones de Europa</h1>
</header>
<div id="result"></div>
</body>
</html>
```

Guión 2: linksdata.json

```
{
    "title": "UEFA Champions League",
    "link": "http://es.uefa.com/uefachampionsleague/",
    "description": "Liga de campeones de Europa",
    "language": "es",
    "items": [
        {
            "title" : "Real Madrid CF",
            "link"  : "http://www.realmadrid.com/",
            "logo" : "logos/real-madrid.png",
             "description": "El Real Madrid es una entidad polideportiva con sede en
Madrid, España."
        },
        {
            "title": "FC Barcelona",
            "link": "http://www.fcbarcelona.com/",
            "logo" : "logos/fc-barcelona.png",
            "description": "FC Barcelona - Más que un club."
        },
        {
            "title": "FC Bayern München",
            "link": "http://www.fcbayern.de/es/",
            "logo" : "logos/bayern-munich.png",
```

```
            "description": "Bayern Munich - Nosotros somos nosotros."
        },
        {
            "title": "Paris Saint Germain",
            "link": "http://www.psg.fr/es/accueil/0/home",
            "logo" : "logos/paris-saint-germain.png",
            "description": "Paris Saint Germain - Aquí es París."
        }
    ]
}
```

Guión 2: linksjson.js

```
function ajaxRequest(){
    //Crear array() con cadenas para creación de objeto ActiveX
    //en caso de navegadores antiguos de Internet Explorer
    var activexmodes = ["Msxml2.XMLHTTP", "Microsoft.XMLHTTP"];
     //Test for support for ActiveXObject in IE first (as XMLHttpRequest in IE7 is
broken)
    if(window.ActiveXObject){
        for(var i=0; i<activexmodes.length; i++){
            try{
                return new ActiveXObject(activexmodes[i]);
            }
            catch(e){
                return false;
            }
        }
    }
    // Si se está usando Chrome, Mozilla, Safari, Opera, etc.
    else if (window.XMLHttpRequest){
        return new XMLHttpRequest();
    }
    else{
        return false;
    }
}

var request = new ajaxRequest();
request.onreadystatechange = function(){
    if(request.readyState==4){
        if(request.status==200 || window.location.href.indexOf("http")==-1){
        //Recibir resultado como un objeto de JavaScript usando la función eval()
        //var jsondata = eval("("+request.responseText+")");
        //Recibir resultado como un objeto de JavaScript usando el método parse()
            var jsondata = JSON.parse(request.responseText);
            var rssentries = jsondata.items;
            var output = "<ul>";
            for(var i=0; i<rssentries.length; i++){
                output += "<li>";
                    output += "<a href=\"" + rssentries[i].link + "\" title=\"" +
rssentries[i].title + "\">";
                output += "<img src=\"" + rssentries[i].logo + "\">";
                output += "<h3>" + rssentries[i].title + "</h3>";
                output += "<p>" + rssentries[i].description + "</p>";
                output += "</a>";
                output += "</li>";
            }
            output += "</ul>";
            document.getElementById("result").innerHTML = output
        }
        else{
            alert("Ha ocurrido un error mientras se realizaba la petición");
        }
    }
}

request.open("GET", "json/linksdata.json", true);
```

```
request.send(null);
```

Guión 4: linkbox.css

```
* {
    margin: 0;
    padding: 0;
}

html, body {
    height: 100%;
}

body {
    /* Old browsers */
    background: #d8e0de;
    /* FF3.6+ */
      background: -moz-linear-gradient(top, #d8e0de 0%, #aebfbc 22%, #99afab 43%,
#8ea6a2 63%, #829d98 84%, #4e5c5a 100%);
    /* Chrome,Safari4+ */
             background:   -webkit-gradient(linear,   left   top,   left   bottom,
color-stop(0%,#d8e0de),      color-stop(22%,#aebfbc),      color-stop(43%,#99afab),
color-stop(63%,#8ea6a2), color-stop(84%,#829d98), color-stop(100%,#4e5c5a));
    /* Chrome10+,Safari5.1+ */
        background:  -webkit-linear-gradient(top,  #d8e0de  0%,#aebfbc  22%,#99afab
43%,#8ea6a2 63%,#829d98 84%,#4e5c5a 100%);
    /* Opera 11.10+ */
     background: -o-linear-gradient(top, #d8e0de 0%,#aebfbc 22%,#99afab 43%,#8ea6a2
63%,#829d98 84%,#4e5c5a 100%);
    /* IE10+ */
     background: -ms-linear-gradient(top, #d8e0de 0%,#aebfbc 22%,#99afab 43%,#8ea6a2
63%,#829d98 84%,#4e5c5a 100%);
    /* W3C */
         background:  linear-gradient(to  bottom,  #d8e0de  0%,#aebfbc  22%,#99afab
43%,#8ea6a2 63%,#829d98 84%,#4e5c5a 100%);
    /* IE6-9 */
       filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#d8e0de',
endColorstr='#4e5c5a',GradientType=0 );
    font-size: 16px;
}

header h1 {
    color: white;
    font: bold 48px Verdana,Helvetica,"Liberation Sans",sans-serif;
    text-align: center;
    text-shadow: 1px 1px #fe4902,
                 2px 2px #fe4902,
                 3px 3px #fe4902;
    -webkit-transition: all 0.12s ease-out;
    -moz-transition: all 0.12s ease-out;
    -ms-transition: all 0.12s ease-out;
    -o-transition: all 0.12s ease-out;
}

header h1:hover {
    position: relative;
    top: -3px;
    left: -3px;
    text-shadow: 1px 1px #fe4902,
                 2px 2px #fe4902,
                 3px 3px #fe4902,
                 4px 4px #fe4902,
                 5px 5px #fe4902,
                 6px 6px #fe4902;
}

#result {
    margin: 20px auto;
```

```
    width: 400px;
}

#result ul {
    list-style-type: none;
    width: 400px;
}

#result ul li a {
       text-decoration: none;
}

#result ul li h3 {
    color: #ff670f;
    font: bold 1.4em/1.5 Tahoma,Helvetica,"Liberation Sans",sans-serif;
}

#result ul li img {
       border: 2px solid #d3bfbc;
       -moz-box-shadow: 4px 4px 5px #b68d4c;
       -webkit-box-shadow: 4px 4px 5px #b68d4c;
       box-shadow: 4px 4px 5px #b68d4c;
       -moz-border-radius: 25px;
       -webkit-border-radius: 25px;
       border-radius: 25px;
    float: left;
    margin: 0 15px 0 0;
}

#result ul p {
    font: 200 0.9em/1.5 Georgia,"Bitstream Serif",serif;
    color: #1f3b08;
}

#result ul li {
    /* Old browsers */
    background: #e8ccbe;
    /* FF3.6+ */
    background: -moz-linear-gradient(top, #e8ccbe 2%, #d3bfbc 99%);
    /* Chrome,Safari4+ */
             background:   -webkit-gradient(linear,   left   top,   left   bottom,
color-stop(2%,#e8ccbe), color-stop(99%,#d3bfbc));
    /* Chrome10+,Safari5.1+ */
    background: -webkit-linear-gradient(top, #e8ccbe 2%,#d3bfbc 99%);
    /* Opera 11.10+ */
    background: -o-linear-gradient(top, #e8ccbe 2%,#d3bfbc 99%);
    /* IE10+ */
    background: -ms-linear-gradient(top, #e8ccbe 2%,#d3bfbc 99%);
    /* W3C */
    background: linear-gradient(to bottom, #e8ccbe 2%,#d3bfbc 99%);
    /* IE6-9 */
        filter:  progid:DXImageTransform.Microsoft.gradient(startColorstr='#e8ccbe',
endColorstr='#d3bfbc',GradientType=0);
    height: auto;
    overflow: auto;
    padding: 10px;
}

#result ul li:hover{
    /* Old browsers */
    background: #f3e2c7;
    /* FF3.6+ */
      background: -moz-linear-gradient(top, #f3e2c7 0%, #c19e67 48%, #b68d4c 52%,
#e9d4b3 100%);
    /* Chrome,Safari4+ */
             background:   -webkit-gradient(linear,   left   top,   left   bottom,
color-stop(0%,#f3e2c7),      color-stop(48%,#c19e67),      color-stop(52%,#b68d4c),
color-stop(100%,#e9d4b3));
```

```
    /* Chrome10+,Safari5.1+ */
        background:  -webkit-linear-gradient(top,  #f3e2c7  0%,#c19e67  48%,#b68d4c
52%,#e9d4b3 100%);
    /* Opera 11.10+ */
     background: -o-linear-gradient(top, #f3e2c7 0%,#c19e67 48%,#b68d4c 52%,#e9d4b3
100%);
    /* IE10+ */
     background: -ms-linear-gradient(top, #f3e2c7 0%,#c19e67 48%,#b68d4c 52%,#e9d4b3
100%);
    /* W3C */
        background:  linear-gradient(to  bottom,  #f3e2c7  0%,#c19e67  48%,#b68d4c
52%,#e9d4b3 100%);
    /* IE6-9 */;
       filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f3e2c7',
endColorstr='#e9d4b3',GradientType=0 );
    cursor: pointer;
}
```

Resultado:

Ejercicio #5: El siguiente ejemplo muestra cómo crear un formulario de autocompletado que muestre sugerencias que coincidan con los caracteres que se van ingresando en una caja de texto. Primero se deben cargar los datos presionando un botón cargar datos para que luego, mientras se vaya digitando en la caja de texto la aplicación genere las autosugerencias coincidentes. Este ejemplo utiliza la propiedad responseXML

Guión 1: dataset_filter.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Search </title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/dataset_filter.css">
    <script src="js/dataset_filter.js"></script>
    <script src="js/util_ajax.js"></script>
</head>
<body>
<header>
    <h1>Catálogo Librería</h1>
</header>
<section>
<article>
<div>
    <input type="button" value="load" onclick="get_booklist();"><br><br>
    <input type="text" name="search" onkeyup="searchkeyup(event);"><br>
                    <div     style="width:600px;height:30px;font-family:helvetica;"
id="id_title"></div>
```

```
        <ul  style="border:1px  solid  #99f;width:600px;height:330px;"  class="bacon"
id="id_booklist">
    </ul>
    <br>
</article>
</div>
</section>
</body>
</html>
```

Guión 2: util_ajax.js

```
function xml_text_node(elem,nam){
    try{
        return elem.getElementsByTagName(nam)[0].firstChild.nodeValue;
    }
    catch(e){
        return "";
    }
}

var AJAX = {
    XMLHttpFactories : [
        function () {return new XMLHttpRequest()},
        function () {return new ActiveXObject("Msxml2.XMLHTTP")},
        function () {return new ActiveXObject("Msxml3.XMLHTTP")},
        function () {return new ActiveXObject("Microsoft.XMLHTTP")}
    ],
    createXMLHTTPObject : function (){
        xmlhttp = null;
        for (var i=0;i<AJAX.XMLHttpFactories.length;i++) {
            try {
                xmlhttp = AJAX.XMLHttpFactories[i]();
            }
            catch (e) {
                continue;
            }
            break;
        }
        return xmlhttp;
    },
    objectToURL : function (url,query_str_obj){
           url_str=url + '?';
           for(property in query_str_obj)
                                                  url_str+=   property   +   "="   +
encodeURIComponent(query_str_obj[property]) + "&";
           return url_str;
    },
    execute : function (url, functionCallBack){
        request = new Object();
        request.ajaxRequest = AJAX.createXMLHTTPObject();
        request.sendRequest = function(){
            //Never happen, cause we create a new request each time
            if(request.ajaxRequest.readyState == 1){
                alert('error: still processing previous request');
                return;
            }

            request.ajaxRequest.open('get', url);
            request.ajaxRequest.onreadystatechange = function(){
                                       if(request.ajaxRequest.readyState   ==   4   &&
request.ajaxRequest.status == 200){
                    //alert(request.ajaxobj.responseText);
                    functionCallBack(request.ajaxRequest);
                }
            }
            request.ajaxRequest.send(null);
        }
```

```
        request.sendRequest();
        return request;
    }
};
```

Guión 3: dataset_filter.js

```
var g_booklist = new Array();
var g_searchstr = '';
var g_lastcount = 0;

function get_booklist(){
    AJAX.execute('xml/amazon.xml', ajax_response);
}

function getChildrenByTagName(element,name){
    var list = new Array();
    name = name.toUpperCase();
    for(var i=0; i<element.childNodes.length; i++)
        if (element.childNodes[i].nodeName.toUpperCase()==name)
            list.push( element.childNodes[i] );
    return list;
}

function getTextByName(element, name){
    name = name.toUpperCase();
    for(var i=0; i<element.childNodes.length; i++)
        if (element.childNodes[i].nodeName.toUpperCase()==name)
            return element.childNodes[i].firstChild.nodeValue;
    return '';
}

function ajax_response(ajaxobj){
    var xml = ajaxobj.responseXML;
          var  doc  =  (xml.firstChild.nextSibling)?  xml.firstChild.nextSibling  :
xml.firstChild;
    var booksxml = getChildrenByTagName(doc, 'book');

    if (booksxml!= null && booksxml.length>0)
    {
        g_booklist=new Array();
        for(var i=0; i<booksxml.length; i++)
        {
            t = getTextByName(booksxml[i],'title');
            a = getTextByName(booksxml[i],'author');
            u = getTextByName(booksxml[i],'url');
            g_booklist.push( createBook(a,t,u) );
        }
        g_booklist.sort(sortByTitle);

        g_searchstr='';
        for (var i=0; i<g_booklist.length; i++)
                                        g_searchstr+=   '   '+g_booklist[i].author+'|
'+g_booklist[i].title+'[|['+i+']|]';
        g_searchstr = g_searchstr.toLowerCase();

        showlist(g_booklist);
    }

}

function showlist(booklist){
    var dom_tracklist = document.getElementById('id_booklist');
    removeChildren( dom_tracklist );

    var count = (booklist.length > 15) ? 15 : booklist.length;
    for(var i=0; i<count; i++)
    {
```

```
        dom_tracklist.appendChild( createBookElement(booklist[i]) );
    }
    g_lastcount=count;

    var dom_tracklist = document.getElementById('id_title');
    removeChildren( dom_tracklist );
     dom_tracklist.appendChild( document.createTextNode("showing " + count + " of "
+ booklist.length));
}

function createBookElement(bookobj){
    var tagText = bookobj.title + ' - ' + bookobj.author;
    var element;
    try {
        element = document.createElement("<li>");//IE
    } catch (e) {
        element = document.createElement( "li" );
    }
    element.appendChild(createLinkElement(bookobj.url, tagText));
    return element;
}

function createLinkElement(url,text){
    var element;
    try {
        element = document.createElement("<a href='" + url + "'>");//IE
    } catch (e) {
        element = document.createElement( "a" );
        element.setAttribute('href',url);
    }
    element.appendChild( document.createTextNode( text ) );
    return element;
}

function searchkeyup(e){
    e = fixevent(e);
    targ=findtarget(e);
    var searchtext = ' '+targ.value.toLowerCase();
    if (searchtext.length<=2){
        if (g_lastcount!= g_booklist.length){
            showlist(g_booklist);
        }
        return;
    }

    var booklist=new Array();
    var results =g_searchstr.split(searchtext);
    for(var i=1; i<results.length; i++){
        var idx1 = results[i].indexOf('[|[');
        var idx2 = results[i].indexOf(']|]');
        if (idx1!=-1 && idx2!=-1 ){
           var idx = results[i].substring(idx1+3, idx2);
           if (isInt(idx)){
               booklist.push( g_booklist[idx] );
           }
        }
    }
    showlist(booklist);
}


function isInt (str){
    var i = parseInt (str);
    if (isNaN (i))
        return false;
    if (i.toString() != str)
        return false;
    return true;
```

```
}

function fixevent(e) {
     return (!e) ? window.event : e;
}

function findtarget(e) {
     if (e.target) targ = e.target;
     else if (e.srcElement) targ = e.srcElement;
     // defeat Safari bug
     if(targ.nodeType == 3)
     targ = targ.parentNode;
     return targ;
}

function removeChildren(list){
    if (list==null) return;
    var child = list.firstChild;
    while(child!=null){
        list.removeChild(child);
        child = list.firstChild;
    }
}

function sortByTitle(a,b){
        if(a.title > b.title) return 1;
        if(a.title < b.title) return -1;
        return 0;
}

function sortByAuthor(a,b){
        if(a.author > b.author) return 1;
        if(a.author < b.author) return -1;
        return 0;
}

function createBook(author,title,url){
    var a = new Object();
    a.author = author;
    a.title = title;
    a.url = url;
    return a;
}

function print_r(obj){
    var str2='';
    for(s in obj){
        str2 += "Array[" + s + "]" + obj[s] + "\n";
    }
    return str2;
}
```

Guión 4: dataset_filter.css

```
ul.bacon {
       list-style-type: none;
       margin: 0px 0px 0px 0px;
       padding: 0px 0px 0px 0px;
       font-size: 11px;
       font-family: Arial, sans-serif;
}

ul.bacon li {
       margin: 0px 0px 0px 0px;
       padding: 2px 2px 2px 2px;
       border: 1px solid #aaa;
       background: #ddd;
       width:594px;
```

```
        height:16px;
        color:#000;
}
```

Resultado:

V. ANÁLISIS DE RESULTADOS

1. Combinando la forma de funcionamiento de los ejercicios 2 y 3 del procedimiento de la guía de práctica, realice un ejemplo en donde aparezcan los afiches de las películas y al dar clic sobre uno de ellos se carguen los datos de un archivo JSON donde estará la información de esas películas. El JSON debe contener el título de la películas, los protagonistas (en una sola cadena), una sinopsis breve (al menos un párrafo de dos líneas) y el director. Debe realizar la funcionalidad con peticiones asíncronas al servidor; es decir, usar AJAX.

2. En el último ejemplo que funciona con datos provenientes de un archivo XML, agregue la funcionalidad para que al hacer clic sobre el enlace de las autosugerencias encontradas se llene el cuadro de texto con el título del libro sobre el que se hizo clic usando JavaScript no invasivo (unobstrisive).

VI. BIBLIOGRAFÍA

- Paul Deitel/Harvey Deitel/Abbey Deitel. Internet & World Wide Web. 1ra Edición. Editorial Pearson. 2014. México.