

# Solución de la tarea para DWE07.

## ENUNCIADO

Queremos programar un lector de RSS en JavaScript, que utilice y que almacene nuestros recursos RSS en una base de datos.

Este lector RSS se apoya en una página PHP, que se encarga de conectar al recurso RSS y devolver los datos obtenidos, en diferentes formatos (texto, JSON, etc.), dependiendo de la petición solicitada a dicha página. En el fichero rss.php se facilita toda la información de las posibles opciones y parámetros que se pueden utilizar.

Se facilita la página HTML con la estructura del lector. Requisitos:

- ✓ El contenido HTML devuelto por los recursos RSS se almacenará en un array, para tenerlo siempre en memoria. Al cargar la aplicación se hará una pre-carga en memoria, de todos los recursos RSS disponibles en la base de datos.
- ✓ Al pulsar el botón "Añadir RSS" se solicitará que introduzcamos el título del nuevo recurso RSS, y a continuación la URL del recurso. Si pulsamos Aceptar, hará una petición AJAX (programada por el método clásico) al recurso rss.php (ver los parámetros en el código fuente). Cuando termine dicha petición, realizará la pre-carga de dicho recurso RSS en memoria.
- ✓ Cada vez que cambiemos de opción en el listado desplegable de RSS, realizará un difuminado para ocultar el recurso actual, accederá a la memoria y mostrará el HTML del nuevo recurso RSS, empleando un efecto de difuminado. Cuando pulsemos < **ANTERIOR** o **SIGUIENTE**> nos cambiará al correspondiente recurso RSS, usando los efectos de difuminado antes mencionados.
- ✓ Al pulsar el botón de "Eliminar RSS", borrará de la base de datos el recurso y nos situará en el primer RSS de la lista.
- ✓ Cada vez que se produzca algún tipo de actividad AJAX, mostrar el indicador AJAX facilitado en el contenedor "indicador".

Se valorará el uso de funciones para realizar las opciones solicitadas anteriormente.

Sólo será necesario programar usando el método clásico de AJAX, en la opción "Añadir RSS". En el resto de los casos, todas las peticiones AJAX que hagáis, animaciones, etc. los podréis hacer utilizando jQuery.

Se facilita la tabla de MySQL, y todos los ficheros necesarios para el proyecto. Programar todo el código de JavaScript dentro del fichero index.js.

Se os facilita un vídeo de demostración de funcionamiento de un lector RSS hecho con JavaScript, AJAX y jQuery, para que veáis cómo tendría que funcionar vuestra aplicación.

[Resumen textual alternativo](#)

**Captura de cómo podría quedar la aplicación:**



Captura de lo que puede ser una posible solución a la tarea.

**SOLUCIÓN:**

[Fichero .zip con el enunciado y recursos necesarios para realizar la tarea DWE07.](#) (0.21 MB)

**CÓDIGO FUENTE DE SOLUCIÓN A LA TAREA**

**FICHERO: DWE07\_TAREA.html**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>SOLUCION TAREA DWE07</title>
<link rel="stylesheet" type="text/css" href="estilos.css">
<script type="text/javascript" src="DWE07_SOLTAREA.js"></script>
<script type="text/javascript" src="funciones.js"></script>
<script type="text/javascript" src="index.js"></script>
</head>
<body>
<form name="formulario">
<div id="principal">

    <div id="cabecera">
        <div id="indicador"></div>
        <div id="titulo">Lector de Titulares RSS con AJAX y jQuery</div>
    </div>

    <div id="seccionFeeds">
        <div id="noticias">
            Cargando Contenido...
        </div>

    </div>

    <div id="controles">
        <input type="button" name="anterior" id="anterior" value="< ANTERIOR" />
        <input type="button" name="siguiente" id="siguiente" value="SIGUIENTE"> />
        <hr/>
        <select name="campoSelect" id="campoSelect"></select>
        <input type="button" name="crearRSS" id="crearRSS" value="Añadir RSS" />
        <input type="button" name="borrarRSS" id="borrarRSS" value="Eliminar RSS" />
    </div>

</div>
</form>
</body>
</html>

```

#### FICHERO: estilos.css

```

/* Estructura de la página */
body{
    background-color: #000000;
}

#principal{
    width: 1024px;
    padding: 3px;
    border: 4px solid white;
    margin: 0 auto;
}

```

```
}

#cabecera{
    height: 32px;
    line-height: 26px;
    background-color: #1677ED;
    font-size: 20px;
    padding: 2px;
}

#indicador{
    float: right;
    font-size: 15px;
    padding-right: 10px;
}

/* Noticias RSS */
#seccionFeeds{
    height: 600px;
    position: relative;
    background: #48B3F2;
    padding: 0 0 0 10px;
}

#noticias, #pie{
    width: 100%;
    height: 600px;
    position: relative;
    overflow: auto;
    left: 0px;
}

#noticias{
    top: 0px;
}

#pie{
    top: -600px;
}

/* Pie de la página */

#controles{
    padding-top: 4px;
    padding-bottom: 4px;
    background-color: #1677ED;
    text-align: center;
}

#controles input{
    width: 100px;
    color: #000000;
}
```

```
font-size: 10px;
background-color: #89D393;
}

#controles select{
    background-color: #89D393;
}
```

#### FICHERO: DWE07\_SOLTAREA.js

```
////////////////////////////////////////
// Cuando el documento esté preparado realizamos las siguientes tareas.
////////////////////////////////////////

// Variables GLOBALES.
// Esta variable contendrá todo el contenido HTML de los RSS para la precarga.
var misRSS=new Array();

// Esta variable contiene el RSS que se está mostrando actualmente.
var rssActivo=-1;

// Creamos un objeto XHR para hacer las peticiones AJAX a nivel global.
var miXHR = new objetoXHR();

// Cuando el documento esté preparado.
$(document).ready(function()
{
    // Cubrimos el campo select con el listado de RSS.
    cubrirSelectRSS();

    // Precargamos todos los RSS y los metemos en un array.
    precargarRSS("todo");

    // Mostrar el RSS seleccionado por defecto en el campo select.
    mostrarSeleccionRSS();

    // Programamos el evento de cambio en el campo Select de RSS.
    $("#campoSelect").change(function()
    {
        mostrarSeleccionRSS();
    });

    // Programamos el evento de pulsar el boton de Eliminar RSS.
    $("#borrarRSS").click(function()
    {
        if ($("#campoSelect").val() != "-1")
        {
            if (confirm("Esta seguro/a de eliminar este recurso RSS"))

```

```

        {
            $.get("rss.php",{ accion: "borrar", id: $("#campoSelect").val() }
            {
                if (datos=="OK")
                    cubrirSelectRSS();
                    mostrarSeleccionRSS();
            });
        }

    }

});

// Programamos el boton <ANTERIOR
$("#anterior").click(function()
{
    if ($("#campoSelect option:selected").index() != 0) // Si la opción seleccion
    {
        document.getElementById("campoSelect").selectedIndex--;
        mostrarSeleccionRSS();
    }
});

// Programamos el boton SIGUIENTE>
$("#siguiente").click(function()
{
    // Si el índice de la opción seleccionada no es la última...
    if ($("#campoSelect option:selected").index() < ($("#campoSelect option").si
    {
        document.getElementById("campoSelect").selectedIndex++;
        mostrarSeleccionRSS();
    }
});

// Asignamos eventos al botón de Añadir RSS
crearEvento(document.getElementById("crearRSS"),"click",grabarRSS);

});

function cubrirSelectRSS()
{
    // Leemos el listado de RSS y actualizamos el campo select
    $.getJSON("rss.php?accion=recursosRSS",function(datos)
    {
        var texto="";
        $.each(datos, function(contador,objeto)
        {
            texto+="

```

```
});  
}  
  
function precargarRSS(numero)  
{  
    // Inicializamos el número total de RSS.  
    // Esta variable la necesitamos para desactivar el indicador AJAX  
    // cuando termine de precargar todos los RSS.  
    var total=0;  
    var contador=0;  
  
    activarIndicadorAjax();  
  
    // Averiguamos cuantos RSS hay en la tabla.  
    $.get("rss.php?accion=numRSS", function(datos)  
    {  
        if (numero==1)  
            total=1;  
        else  
            total=datos;  
    });  
  
    // Leemos el listado de RSS y actualizamos el campo select  
    $.getJSON("rss.php?accion=recursosRSS",function(datos)  
    {  
        // Para cada registro del listado de RSS hace lo siguiente:  
        $.each(datos, function(indice1,objeto)  
        {  
  
            if (numero=="todo" || numero==indice1)  
            {  
                // Carga el contenido JSON de ese recurso RSS en Internet.  
                $.getJSON("rss.php",{accion: "cargar", id: objeto.id}, function(d  
                {  
                    var texto="";  
                    $.each(datosrss, function(indice2,objeto)  
                    {  
                        texto+="

### "+objeto.titulo+"</h3>"+objeto.descripcion+ }); // Metemos todo el contenido del RSS en misRSS, en la posici // al id del RSS en la base de datos. misRSS[indice1]=texto; contador++; if (contador==total) desactivarIndicadorAjax(); }); } }); }); });


```

```

}

////////////////////////////////////////
// Función mostrarSeleccionRSS: muestra en el contenedor #noticias el RSS
// que está seleccionado en el campo de selección.
// Lo hace empleando efectos de animación fadeIn y fadeOut.
////////////////////////////////////////
function mostrarSeleccionRSS()
{
    activarIndicadorAjax();

    $("#noticias").fadeOut(1000,function()
    {
        var texto="<h1>Fuente RSS: "+$("#campoSelect option:selected").text()+"</h1>";
        $("#titulo").html("Lector de Titulares RSS con AJAX y jQuery >>> Fuente RSS");
        $(this).html(texto).delay(1500).fadeIn(1000,function(){desactivarIndicadorA
        });
    });
}

////////////////////////////////////////
// Función grabarRSS: será llamada a cada cambio de estado de la petición AJAX
// cuando la respuesta del servidor es 200(fichero encontrado) y el estado de
// la solicitud es 4, accedemos a la propiedad responseText
////////////////////////////////////////
function grabarRSS()
{
    tituloRSS=prompt("Introduzca Titulo del nuevo recurso RSS","");
    urlRSS=prompt("Introduzca URL del nuevo recurso RSS","http://");
    if (tituloRSS !="" && tituloRSS !=null && urlRSS!="http://" && urlRSS != null)
    {
        cargarAsync("rss.php?accion=nueva&titulo="+tituloRSS+"&url="+urlRSS,fu
        {
            if (this.readyState==4 && this.status == 200)
            {
                precargarRSS(this.responseText);
                cubrirSelectRSS();
            }
        });
    }
}
}

```

#### FICHERO: funciones.js

```

////////////////////////////////////////
// Función cross-browser para crear objeto XMLHttpRequest
////////////////////////////////////////
function objetoXHR()

```

```

{
    if (window.XMLHttpRequest)
    {
        // El navegador implementa la interfaz XHR de forma nativa
        return new XMLHttpRequest();
    }
    else if (window.ActiveXObject)
    {
        var versionesIE = new Array('Msxml2.XMLHTTP.5.0', 'Msxml2.XMLHTTP.4.0',
            'Msxml2.XMLHTTP.3.0', 'Msxml2.XMLHTTP', 'Microsoft.XMLHTTP');

        for (var i = 0; i < versionesIE.length; i++)
        {
            try
            { /* Se intenta crear el objeto en Internet Explorer comenzando
                en la versión más moderna del objeto hasta la primera versión.
                En el momento que se consiga crear el objeto, saldrá del bucle
                devolviendo el nuevo objeto creado. */
                return new ActiveXObject(versionesIE[i]);
            }
            catch (errorControlado) {} //Capturamos el error,
        }
    }

    /* Si llegamos aquí es porque el navegador no posee ninguna forma de crear el ob
    Emitimos un mensaje de error usando el objeto Error.

    Más información sobre gestión de errores en:
    http://www.javascriptkit.com/javatutors/trycatch2.shtml */

    throw new Error("No se pudo crear el objeto XMLHttpRequest");
}

////////////////////////////////////
// Función cross-browser para añadir Eventos
////////////////////////////////////
var crearEvento = function()
{
    function w3c_crearEvento(elemento, evento, mifuncion)
    {
        elemento.addEventListener(evento, mifuncion, false);
    }

    function ie_crearEvento(elemento, evento, mifuncion)
    {
        var fx = function()
        {
            mifuncion.call(elemento);
        };

        // Enlazamos el evento con attachEvent. Cuando usamos attachEvent

```



```

// dejamos de tener acceso al objeto this en mifuncion. Para solucionar eso
// usaremos el método call() del objeto Function, que nos permitirá
// asignar el puntero this para su uso dentro de la función. El primer
// parámetro que pongamos en call será la referencia que se usará como
// objeto this dentro de nuestra función. De esta manera solucionamos el pr
// de acceder a this usando attachEvent en Internet Explorer.

elemento.attachEvent('on' + evento, fx);
}

if (typeof window.addEventListener !== 'undefined')
{
    return w3c_crearEvento;
}

else if (typeof window.attachEvent !== 'undefined')
{
    return ie_crearEvento;
}

})(); // <= Esta es la parte más crítica - tiene que terminar en ()

////////////////////////////////////
// Función cross-browser para modificar el contenido
// de un DIV
////////////////////////////////////
function textoDIV(nodo, texto)
{
    //var nodo = document.getElementById(idObjeto);
    while (nodo.firstChild)
        nodo.removeChild(nodo.firstChild); // Eliminamos todos los hijos de ese obj
    // Cuando ya no tenga hijos, agregamos un hijo con el texto que recibe la funció
    nodo.appendChild(document.createTextNode(texto));
}

////////////////////////////////////
// Función activarIndicadorAjax: activa el gif animado
// indicando que se está ejecutando una petición AJAX.
// Es necesario tener un contenedor en el documento con el id="indicador".
////////////////////////////////////
function activarIndicadorAjax()
{
    // Activamos el indicador Ajax antes de realizar la petición.
    document.getElementById("indicador").innerHTML="<img src='ajax-loader.gif' /
}

////////////////////////////////////
// Función desactivarIndicadorAjax: desactiva el gif animado
// que indicaba que se estaba ejecutando una petición AJAX.
// Es necesario tener un contenedor en el documento con el id="indicador".
////////////////////////////////////
function desactivarIndicadorAjax()

```

```

{
    // Activamos el indicador Ajax antes de realizar la petición.
    document.getElementById("indicador").innerHTML="";
}

////////////////////////////////////////
// Función cargarAsync: carga el contenido de la url
// usando una petición AJAX a la url indicada y llamará a
// la funcionRetorno cuando termine la petición AJAX.
////////////////////////////////////////
function cargarAsync(url,functionRetorno)
{
    if (miXHR)
    {
        activarIndicadorAjax();

        //Si existe el objeto  miXHR
        miXHR.open('GET', url, true); //Abrimos la url, true=ASINCRONA

        // En cada cambio de estado(readyState) se llamará a la función indicada en
        miXHR.onreadystatechange = eval(functionRetorno);

        // Hacemos la petición al servidor. Como parámetro: null ya que los datos v
        miXHR.send(null);
    }
}

```

#### FICHERO: rss.php

```

<?php
/*
    Esta página realiza las siguientes acciones según los parámetros recibidos:

    Parámetros:    accion=nueva&url=xxxxx
    Acción:        Insertará la url xxxxx en la tabla MySQL de RSS.
    Salida:        Imprimirá el ID del último recurso añadido.
    Formato:       Texto

    Parámetros:    accion=borrar&id=x
    Acción:        Borrará la url con id x de la tabla de RSS.
    Salida:        Imprimirá mensaje de OK.
    Formato:       Texto

    Parámetros:    accion=leer&url=id
    Acción:        Se conecta a la URL indicada por el ID y descarga el RSS de Int
    Salida:        El fichero XML del RSS en formato JSON.
    Formato:       JSON

    Parámetros:    accion=recursosRSS
    Acción:        Devuelve todos los datos de la tabla rss.

```

Salida: Un array con los campos id, titulo, url en formato JSON.  
Formato: JSON

Parámetros: accion=numRSS  
Acción: Devuelve el número total de RSS que tenemos en la tabla.  
Salida: Un número indicando los RSS que hay en la tabla.  
Formato: texto

\*/

```
// Cabecera para indicar que vamos a enviar datos JSON y que no haga caché de los dat
header('Cache-Control: no-cache, must-revalidate');
header('Expires: Mon, 26 Jul 1997 05:00:00 GMT');
```

```
/* Utilizar el fichero dbcreacion.sql incluido en la carpeta para crear la base
Si fuera necesario modifica los datos de la configuración y adapталos a tu entorno de
```

```
// Configuración BASE DE DATOS MYSQL
```

```
$servidor = "localhost";
$basedatos = "ajax";
$usuario = "ajax";
$password = "dwec";
```

```
// Creamos la conexión al servidor.
```

```
$conexion=mysql_connect($servidor, $usuario, $password) or die(mysql_error());
mysql_query("SET NAMES 'utf8'", $conexion);
```

```
// Seleccionar la base de datos en esa conexión.
```

```
mysql_select_db($basedatos, $conexion) or die(mysql_error());
```

```
switch ($_GET['accion'])
```

```
{
```

```
case 'nueva':
```

```
    $sql= sprintf("insert into rss(titulo,url) values('%s','%s')",mysql_real_es
    mysql_query($sql,$conexion) or die(mysql_error());
    echo mysql_insert_id();
```

```
break;
```

```
case 'borrar':
```

```
    $sql= sprintf("delete from rss where id=%s",mysql_real_escape_string($_GET[
    mysql_query($sql,$conexion) or die(mysql_error());
    echo "OK";
```

```
break;
```

```
case 'cargar':
```

```
    // Consulta SQL para obtener los datos de los centros.
```

```
    $sql=sprintf("select * from rss where id=%s",mysql_real_escape_string($_GET
    $resultados=mysql_query($sql,$conexion) or die(mysql_error());
    $registros=mysql_fetch_assoc($resultados);
```

```
$doc = new DOMDocument();
$doc->load($registros['url']);
$arrFeeds = array();
foreach ($doc->getElementsByTagName('item') as $node)
{
    $itemRSS = array (
        'titulo' => $node->getElementsByTagName('title')->item(0)->nodeValue,
        'descripcion' => $node->getElementsByTagName('description')->item(0)->nodeValue,
        'url' => $node->getElementsByTagName('link')->item(0)->nodeValue,
        'fecha' => $node->getElementsByTagName('pubDate')->item(0)->nodeValue
    );

    array_push($arrFeeds, $itemRSS);
}

header('Content-Type: application/json');
echo json_encode($arrFeeds);
break;

case 'recursosRSS':
    $sql= sprintf("select * from rss order by id");
    $resultados=mysql_query($sql,$conexion) or die(mysql_error());
    while ($fila = mysql_fetch_array($resultados, MYSQL_ASSOC))
    {
        // Almacenamos en un array cada una de las filas que vamos leyendo del
        // Cada fila del array coincidirá con el id del RSS.
        $datos[$fila['id']]=$fila;
    }
    header('Content-Type: application/json');
    echo json_encode($datos);
break;

case 'numRSS':      // Devuelve el número total de RSS que tenemos en la base de
    $sql= sprintf("select * from rss order by id");
    $resultados=mysql_query($sql,$conexion) or die(mysql_error());
    echo mysql_num_rows($resultados);
break;
}

mysql_close($conexion);
?>
```