	UNIVERSIDAD DON BOSCO FACULTAD DE INGENIERÍA ESCUELA DE COMPUTACIÓN
CICLO 02 2020	<p style="text-align: center;"><i>GUÍA DE LABORATORIO #12</i></p> <p> Nombre de la Practica: Introducción a jQuery Lugar de Ejecución: Centro de Cómputo Tiempo Estimado: 2 horas con 30 minutos MATERIA: Lenguajes Interpretados en el Cliente </p>

I. OBJETIVOS

Que el estudiante:

- Adquiera las nociones básicas para la creación de plugins para jQuery.
- Comience a utilizar plug-ins de jQuery para mejorar la experiencia del usuario.
- Transforme dinámicamente el documento, agregando o eliminando contenido con métodos y propiedades de jQuery.
- Modifique dinámicamente propiedades CSS mediante jQuery

II. INTRODUCCIÓN TEÓRICA

¿QUÉ ES JQUERY?

Es una “librería JavaScript muy rápida y muy ligera que simplifica el desarrollo de la parte de cliente de las aplicaciones web”. En otras palabras, jQuery incluye muchas utilidades para crear fácilmente las páginas web de las aplicaciones dinámicas complejas.

CONCEPTOS BÁSICOS

Para utilizar jQuery, solamente es necesario descargar la librería y enlazar en nuestras páginas el único archivo JavaScript que la forma:

```
<script type="text/javascript" src="jquery.js"></script>
```

La librería se puede descargar en dos versiones: la versión descomprimida ocupa 61 KB y es la mejor opción para desarrollar aplicaciones. La versión comprimida ocupa sólo 20 KB y es la opción ideal en el servidor de producción, para minimizar el tiempo de carga de la aplicación.

Una de las instrucciones más utilizadas en las aplicaciones JavaScript es la siguiente:

```
window.onload = function() { ... }
```

La instrucción anterior permite ejecutar el código JavaScript una vez que la página se ha cargado por completo. Se trata de una instrucción imprescindible en las aplicaciones web complejas, porque cualquier modificación en la página o la selección de cualquier elemento necesita que el árbol DOM de la página se haya construido por completo, y por tanto, que esté disponible el código HTML de la página entera. jQuery dispone de una instrucción muy similar equivalente a la anterior:

```
$(document).ready( function() { ... } );
```

La gran ventaja de la función de jQuery, es que la página se puede manipular en cuanto se ha cargado su código HTML (y por tanto, se ha construido el árbol DOM de la página) mientras que la función de JavaScript espera a que se carguen todos los elementos de la página, incluyendo todas las imágenes. De esta forma, las aplicaciones realizadas con jQuery pueden responder de forma mucho más rápida que las aplicaciones JavaScript tradicionales.

Por otra parte, la función `getElementById()` es posiblemente la más utilizada al desarrollar aplicaciones web, ya que permite seleccionar un elemento a partir de su atributo “id”. Además, existen otras dos funciones relacionadas: `getElementsByTagName()` y `getElementsByName()`. jQuery

incluye la función `$()` como una alternativa más rápida y completa para seleccionar cualquier elemento de la página:

CON JAVASCRIPT

```
// Seleccionar un elemento
document.getElementById("miParrafo");
// Seleccionar todos los enlaces
document.getElementsByTagName("a");
```

CON JQUERY

```
// Seleccionar un elemento
$("#miParrafo");
// Seleccionar todos los enlaces
$("a");
```

SELECTORES AVANZADOS

Uno de los componentes más impresionantes de jQuery es el de sus selectores, que permiten olvidarse de los aburridos `document.getElementById()` y del código JavaScript complejo. jQuery utiliza lo mejor de CSS 1, CSS 2, CSS 3 y XPath para seleccionar de forma sencilla cualquier elemento de la página sin necesidad de saturar el código XHTML con atributos "id". Ejemplos:

```
// Seleccionar todos los párrafos de la página que tengan al menos un
enlace
$("p[a]")
// Seleccionar todos los "div" que no estén ocultos
$("div:visible")
// Seleccionar todos los radio buttons que han sido seleccionados
$("input[@type=radio][@checked]")
// Seleccionar varios elementos mediante sus "id" y "class" (devuelve
un array)
$("p.importante, div#menu, span#introduccion p.especial a")
```

Además, jQuery soporta muchos de los selectores que define CSS 3, algo que los navegadores actuales sólo pueden soñar con tener algún día, como `:nth-child(n)`, `:empty`, `:not`, `:disabled`. Para los programadores más acostumbrados a trabajar con XML, jQuery también permite utilizar XPath para seleccionar elementos, como por ejemplo:

```
// Seleccionar todos los párrafos de la página
$("/html/body//p");
// Seleccionar todos los enlaces cuyo atributo "rel" valga "nofollow"
$("//a[@ref='nofollow']");
```

jQuery es tan avanzado que incluso añade sus propios selectores que no existen ni en CSS ni en XPath, pero que facilitan mucho el desarrollo de las aplicaciones:

```
// Selecciona todos los elementos pares de una lista (":odd" selecciona
los impares)
$("ul#menu li:even")
// Selecciona los 4 primeros párrafos de la página
$("p:lt(4) ")
// Selecciona todos los enlaces que contengan el texto "pincha aquí"
$("a:contains('pincha aquí')")
```

EVENTOS

Además del evento `ready()` visto anteriormente, jQuery dispone de varias funciones relacionadas con la gestión de los eventos. El método `bind()`, por ejemplo, permite asociar un número ilimitado de eventos a un mismo elemento:

```
// Al pinchar sobre un párrafo de la página, se muestra un alert() con su contenido
$("p").bind("click", function() {
    alert( $(this).text() );
});
```

El modelo de eventos de jQuery es muy completo, ya que cada evento dispone de dos posibilidades: si no se pasa ninguna función, se ejecuta el evento del elemento; si se pasa una función, se asocia esta función al evento del elemento:

```
// Ejecuta el evento "onclick" en todos los párrafos de la página
$("p").click();
// Asocia una función al evento "onclick" de todos los párrafos de la página
$("p").click(function() {
    alert( $(this).text() );
});
```

jQuery dispone de tantas funciones como eventos estándar de JavaScript. El nombre de cada función es el mismo que el del evento, sin el habitual prefijo “on”: `focus()`, `blur()`, `keyup()`, `mouseover()`, `mouseup()`, `resize()`, `submit()`, etc.

Además, jQuery añade un evento que no existe en JavaScript y que es realmente útil para muchas aplicaciones: `toggle()`. A este método se le pasan dos funciones, cuya ejecución se alterna en función de las veces que se pincha sobre el elemento.

La primera vez que se pincha sobre el elemento (y todas las veces impares), se ejecuta la primera función y la segunda vez que se pincha el elemento (y todas las veces pares) se ejecuta la segunda función:

```
$("#p").toggle(function(){
    alert("Me acabas de activar");
},function(){
    alert("Me acabas de desactivar");
});
```

INTERACCIONES AJAX

Ahora que la mayoría de aplicaciones web deben incorporar interacciones creadas con Ajax, jQuery permite incluirlas sin ningún tipo de dificultad. Los siguientes ejemplos muestran lo sencillo que es realizar algunas interacciones comunes en Ajax:

```
// Cargar el contenido de una página HTML en un elemento
$("#div#noticias").load("noticias.html");
// Descargar un script desde el servidor y ejecutarlo
$.getScript("/ruta/hasta/miScript.js");
// Petición GET al servidor con parámetros y función de respuesta
// La petición POST es idéntica, salvo que se cambia $.get por $.post
$.get("/ruta/hasta/el/scriptDelServidor.php",
    { idProducto: "AX00342", cantidad: "3" },
    function(data){
        alert("Se ha añadido al carrito: " + data);
    }
);
```

Además, jQuery incluye muchas otras funciones para construir peticiones Ajax complejas y para notificar al usuario el inicio/ejecución/finalización de las peticiones Ajax.

EFFECTOS VISUALES

Todas las aplicaciones web modernas incorporan pequeños efectos visuales y animaciones que, si se utilizan de forma adecuada, mejoran la interacción con la aplicación y su usabilidad. jQuery incluye de serie los efectos visuales más utilizados por los diseñadores, pudiendo controlar mediante sus opciones la duración de cada efecto:

```
$("#p").hide(); // Ocultar un elemento
$("#p").show(); // Mostrar un elemento
$("#p").toggle(); // Si estaba oculto, mostrarlo; si era visible, ocultarlo
$("#p").show("slow"); // Mostrar el elemento con una animación muy suave
$("#p").slideToggle("fast"); // Desplegar/ocultar el elemento con una animación muy rápida
$("#p").fadeOut(); // El elemento desaparece con un fundido muy suave
```

OTRAS UTILIDADES

jQuery incluye muchas otras funciones para facilitar el desarrollo de las aplicaciones web, entre otras:

```
// Obtiene el valor de la propiedad "color" de CSS del primer párrafo de la página
$("#p").css("color");
// Establece el valor de la propiedad "color" de CSS
$("#p").css("color", "red");
// Establece varias propiedades CSS de un elemento
$("#p").css({ color: "red", background: "blue", font-weight: "bold" });
// Oculta todos los elementos de un formulario
$(miFormulario.elements).hide();
// Devuelve "true" si el navegador es alguna versión de Internet Explorer
$.browser.msie
// Devuelve "true" si el navegador es alguna versión de Safari
$.browser.safari
// Obtiene o establece el valor del "innerHTML" del elemento
$("#contenidos").html();
$("#contenidos").html("<h1>Contenidos</h1><p>Cargando...</p>");
// Añade la clase CSS a todos los elementos o la quita si ya la tenían establecida
$("#p").toggleClass("seleccionado");
```

Con jQuery, se pueden hacer mucho más cosas y mucho mejores, escribiendo menos código JavaScript, además de tener la garantía absoluta de que tus aplicaciones funcionan igual de bien en cualquier navegador.

III. MATERIAL Y EQUIPO

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

No.	Requerimiento	Cantidad
1	Guía de práctica #12: Introducción a jQuery	1
2	Computadora con editor HTML instalado y navegadores	1
3	Memoria USB.	1

IV. PROCEDIMIENTO

Ejercicio #1: El siguiente ejemplo muestra cómo crear una galería de imágenes desarrollando completamente toda la funcionalidad, sin plugins o librerías externas de jQuery.

Guión 1: formula1.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Galería de autos de carrera</title>
  <meta charset="utf-8" />
  <link rel="stylesheet"
href="http://fonts.googleapis.com/css?family=Caudex:400,400italic" />
  <link rel="stylesheet" href="css/gallery.css" />
  <script
src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
  <script>
    if (typeof jQuery == 'undefined') {
      document.write(unescape("%3Cscript src='jquery/jquery.js'
type='text/javascript'%3E%3C/script%3E"));
    }
  </script>
  <script src="js/galeria.js"></script>
  <script>
    $(document).ready(function() {
      $("#galeria").galeria();
    });
  </script>
</head>
<body>
<section>
<article>
<div id="galeria">
  
  
  
  
  
  
  
  
</div>
</article>
</section>
</body>
</html>
```

Guión 3: galeria.js

```
(function($) {
  //Creación del método galeria
  $.fn.galeria = function() {
    //Se define una plantilla donde se crea la estructura
    //que deseamos tenga la galería. En el div id="panel"
    //se define un parámetro ({imagenes}) que será utilizado
    //para cargar las imágenes de la galería
    var plantilla = '<div id="thumbs"></div>' +
      '<div id="panel">{imagenes}</div>' +
      '<div id="titulo"></div>';

    //Reemplazando el parámetro {imagenes} por el contenido real
    plantilla = plantilla.replace("{imagenes}", $(this).html());
    $(this).html(plantilla);

    var html_ul = "<ul>";
    $("#panel img").each(function() {
      html_ul += '<li><a href="javascript:void(0);">';
```

```

        html_ul += '';
        html_ul += '</a>';
    });
    html_ul += "</ul>";

    $("#thumbs").html(html_ul);

    //Mostrar imágenes de tamaño completo en el panel principal
    //de la galería
    $("#panel img:eq(0)").show();

    $("#thumbs a").click(function(){
        show($(this).find("img").attr("src"));
    });

    //Hacer aparecer una imagen la primera vez que se carga la página
    $("#thumbs a:eq(0)").click();

    //Aparecer y desaparecer los títulos de las imágenes
    $("#panel").mouseover(function() {
        $("#titulo").fadeIn(750);
    });
    $("#panel").mouseout(function() {
        $("#titulo").fadeOut(750);
    });
}

//Función que mostrará las imágenes de la galería en el panel
function show(src){
    var image = $('#panel img[src="' + src + '"]');
    //Ocultar imágenes presentes en el panel antes de mostrar la actual
    $("#panel img").fadeOut();
    //Mostrar la imagen actual sobre la que se hizo clic
    image.show();
    //Hacer que se agregue un efecto de animación cuando aparezca
    //cada imagen de la galería.
    $("#panel").animate({
        'width': image.innerWidth(),
        'height': image.innerHeight()
    }, 1000, function(){
        image.fadeIn(1000);
        $("#titulo").html("<span>" + image.attr('title') + "</span>");
        $("#titulo").css({
            'background': 'rgba(50,50,50,0.8)',
            'color': 'white',
            'height': "25px",
            'position': 'relative',
            'top': '-25px',
            'width': image.innerWidth()
        });
    });
}
})(jQuery);

```

Resultado:



Ejercicio #2: El siguiente ejemplo muestra una galería creada utilizando el plug-in de jQuery conocido como GalleryView. En este ejemplo se muestra una configuración avanzada del control en el bloque script dentro del HTML. Como podrá notar se invocan varios plug-in de jQuery, necesarios para toda la funcionalidad mostrada.

Guión 1: galleryview.html

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8" />
<title>Radio Don Bosco</title>
<!-- First, add jQuery (and jQuery UI if using custom easing or animation -->
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
<script src="//ajax.googleapis.com/ajax/libs/jqueryui/1.8.18/jquery-
ui.min.js"></script>
<!-- Second, add the Timer and Easing plugins -->
<script src="js/jquery.timers-1.2.js"></script>
<script src="js/jquery.easing.1.3.js"></script>
<!-- Third, add the GalleryView Javascript and CSS files -->
<script src="js/jquery.galleryview-3.0-dev.js"></script>
<link rel="stylesheet" href="css/jquery.galleryview-3.0-dev.css" />
<!-- Lastly, call the galleryView() function on your unordered list(s) -->
<script>
    $(function(){
        $('#myGallery').galleryView({
            //INT - duration of panel/frame transition (in milliseconds)
            transition_speed: 2000,
            //INT - delay between panel/frame transitions (in milliseconds)
            transition_interval: 4000,
            //STRING - easing method to use for animations (jQuery provides 'swing'
            //or 'linear', more available with jQuery UI or Easing plugin)
            easing: 'swing',
            //BOOLEAN - flag to show or hide panel portion of gallery
            show_panels: true,
            //BOOLEAN - flag to show or hide panel navigation buttons
            show_panel_nav: false,
            //BOOLEAN - flag to show or hide panel overlays
            enable_overlays: true,
            //INT - width of gallery panel (in pixels)
            panel_width: 500,
            //INT - height of gallery panel (in pixels)
            panel_height: 313,
            //STRING - animation method for panel transitions (crossfade,
            //fade,slide,none)
            panel_animation: 'slide',
            //STRING - cropping option for panel images (crop = scale image and
            //fit to aspect ratio determined by panel_width and panel_height,
            //fit = scale image and preserve original aspect ratio)
            panel_scale: 'crop',
            //STRING - position of panel overlay (bottom, top)
            overlay_position: 'bottom',
            //BOOLEAN - flag to allow user to grab/drag oversized images
            //within gallery
            pan_images: true,
            pan_style: 'drag',
            //INT - determines smoothness of tracking pan animation
            //(higher number = smoother)
            pan_smoothness: 15,
            //INT - index of panel/frame to show first when gallery loads
            start_frame: 1,
            //BOOLEAN - flag to show or hide filmstrip portion of gallery
            show_filmstrip: true,
            //BOOLEAN - flag indicating whether to display navigation buttons
            show_filmstrip_nav: true,
            //BOOLEAN - flag indicating whether to display slideshow
            //play/pause button
        });
    });
</script>
```

```

        enable_slideshow: false,
        //BOOLEAN - flag to start slideshow on gallery load
        autoplay: false,
        //BOOLEAN - flag to show or hide frame captions
        show_captions: true,
        //INT - number of frames to show in filmstrip-only gallery
        filmstrip_size: 3,
        //STRING - type of filmstrip to use (scroll = display one line of
frames,
        //scroll filmstrip if necessary, showall = display multiple rows of
        //frames if necessary)
        filmstrip_style: 'scroll',
        //STRING - position of filmstrip within gallery (bottom, top, left,
right)
        filmstrip_position: 'bottom',
        //INT - width of filmstrip frames (in pixels)
        frame_width: 96,
        //INT - width of filmstrip frames (in pixels)
        frame_height: 50,
        //FLOAT - transparency of non-active frames (1.0 = opaque, 0.0 =
transparent)
        frame_opacity: 0.5,
        //STRING - cropping option for filmstrip images (same as above)
        frame_scale: 'crop',
        //INT - spacing between frames within filmstrip (in pixels)
        frame_gap: 5,
        //BOOLEAN - flag to show or hide infobar
        show_infobar: true,
        //FLOAT - transparency for info bar
        infobar_opacity: 1
    });
});
</script>
<style type="text/css">
    body {
        margin: 2em 11.2em;
        font-family: Arial,Helvetica,sans-serif;
    }
</style>
</head>
<body>
<section>
<article>
    <ul id="myGallery">
        <li>
            
        </li>
        <li>
            
        </li>
        <li>
            
        </li>
        <li>
            
        </li>
        <li>

```



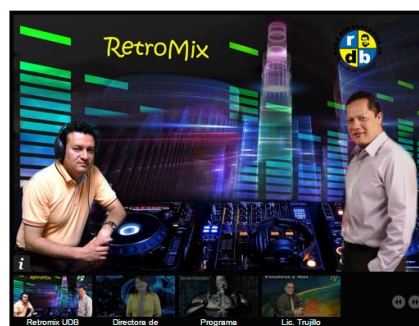
```

        
    </li>
    <li>
        
    </li>
    <li>
        
    </li>
    <li>
        
    </li>
    <li>
        
    </li>
    <li>
        
    </li>
    <li>
        
    </li>
</ul>
</article>
</section>
</body>
</html>

```

IMPORTANTE: Todos los scripts JS y las hojas de estilo relacionadas con este ejercicio irán incluidas en los recursos para esta guía de práctica que deberá descargar desde el sitio web de la Universidad.

Resultado:



Ejemplo #3: El siguiente ejemplo ilustra cómo utilizar un plug-in jQuery para incorporar un reproductor de música en el sitio web que funciona como cualquier reproductor de música de escritorio. En este caso, se está utilizando el plug-in de jQuery speakker que permite crear una lista de reproducción personalizada, mostrar una imagen de la canción que se esté reproduciendo, subir o bajar volumen, avanzar a la siguiente canción o retroceder a la anterior, entre otras funciones.

Guión 1: player.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Speakker | Reproductor de audio</title>
  <meta charset="utf-8" />
  <meta name="description" content="Audioplayer from a new generation" />
  <link rel="Shortcut Icon" type="image/ico" href="img/speakker.ico" />
  <link rel="stylesheet" media="screen" href="css/player.css" />
  <!-- Los scripts para utilizar speakker -->
  <script src="js/jquery.min.js"></script>
  <script src="js/speakker-big-
1.2.24r229.min.js"></script>
  <script>
    $(document).ready(function() {
      projekktor('.projekktor');
    });
  </script>
</head>
<body>
<div id="wrap">
<header class="inside-plus">
  <div id="logo">
    
    <h1>Reproductor Speakker</h1>
  </div>
</header>
<div class="inside">
  <p>
    Esta es una pequeña demostración de un reproductor de audio en web
    llamado Speakker implementado con la librería jQuery.
    Este reproductor usa una lista de reproducción con JSON que puede
    verificar en el archivo <a href="playlist.json">playlist.json</a>
    para que puede tener una idea de cómo personalizar otra lista más
    acorde a su gusto musical.
    Puede encontrar más información sobre cómo hacer su propia lista
    <a href="http://www.projekktor.com/docs/playlists">aquí</a>.
    Si desea chequear un poco la documentación de este reproductor
    ve a <a href="http://www.projekktor.com/docs/api">este sitio</a>.
  </p>
  <p>
    Speakker es un producto de libre distribución. De modo que si te
    gusta lo que ves puedes hacer una:
    <a href="http://www.projekktor.com/donate.php">donación</a>.
  </p>
  <div class="separator">&nbsp;</div>
  <!-- this is where Speakker will sit on. Note that all classes are
obligatory. -->
  <!-- "dark" can be replaced by "light" to change the color scheme -->
  <audio class="projekktor speakker dark">
    <source src="playlist.json" type="application/json" />
  </audio>
  <div id="about">
    <p class="left">
      Speakker es un producto de last.fm-resque audio player construido
      sobre la base del famoso reproductor de vídeo
      <a href="http://www.projekktor.com" title="projekktor
website">projekktor</a>.
    </p>
```

```

        <p class="separator">
        &copy; 2011-2012 <a href="http://www.porkhead.org">Porkhead.org</a>
        &amp; <a href="http://www.spinningairwhale.com">Spinning Airwhale</a>
    </p>
</div>
</div>
<script type="text/javascript" src="js/noise.js"></script>
</body>
</html>

```

Guión 2: playlist.json

```

{"playlist":
  [
    {
      "0": {"src":"junk/Akon - Angel.mp3", "type":"audio/mp3"},
      "config": {
        "title": "Akon - Angel",
        "poster": "junk/Akon - Angel.jpg"
      }
    },
    {
      "0": {"src":"junk/Alexandra Stan - Saxobeat.mp3", "type":"audio/mp3"},
      "config": {
        "title": "Alexandra Stan - Saxobeat",
        "poster": "junk/Alexandra Stan - Saxobeat.jpg"
      }
    },
    {
      "0": {"src":"junk/Katy Perry - Firework.mp3", "type":"audio/mp3"},
      "config": {
        "title": "Katy Perry - Firework",
        "poster": "junk/Katy Perry - Firework.jpg"
      }
    },
    {
      "0": {"src":"junk/Maroon 5 Feat. Christina Aguilera - Move like
jagger.mp3", "type":"audio/mp3"},
      "config": {
        "title": "Maroon 5 Feat. Christina Aguilera - Move like jagger",
        "poster": "junk/Maroon 5 - Moves like jagger.jpg"
      }
    },
    {
      "0": {"src":"junk/Rihanna - We found love.mp3", "type":"audio/mp3"},
      "config": {
        "title": "Rihanna - We found love",
        "poster": "junk/Rihanna - We found love.jpg"
      }
    }
  ]
}

```

Guión 3: noise.js

```

//Gimme some noise...
function generateNoise(opacity) {
  if (!!!document.createElement('canvas').getContext) {
    return false;
  }

  var canvas = document.createElement("canvas"),
  ctx = canvas.getContext('2d'),
  x, y,
  r, g, b,

```

```

opacity = opacity || .2;

canvas.width = 100;
canvas.height = 100;

ctx = canvas.getContext("2d");

for ( x = 0; x < canvas.width; x++ ) {
    for ( y = 0; y < canvas.height; y++ ) {
        r = Math.floor(Math.random() * 80);
        g = Math.floor(Math.random() * 80);
        b = Math.floor(Math.random() * 80);

        ctx.fillStyle = "rgba(" + r + "," + g + "," + b + "," + opacity + ")";
        ctx.fillRect(x, y, 1, 1);
    }
}

document.body.style.backgroundImage = "url(" + canvas.toDataURL("image/png") +
)";
}

//crypt spam
function UnCryptMailto(s) {
    var n=0;
    var r="";
    for(var i=0; i < s.length; i++) {
        n = s.charCodeAt(i);
        if(n>=8364) {n = 128;}
        r += String.fromCharCode(n-(1));
    }
    return r;
}

function linkTo_UnCryptMailto(s) {
    location.href=UnCryptMailto(s);
}

```

IMPORTANTE: La hoja de estilos y los plug-in necesarios para el funcionamiento del script se proporcionarán en los recursos de la guía de práctica.

Resultado:



Ejercicio #4: El siguiente ejemplo muestra cómo crear un acordeón con imágenes como contenido. Se utilizan 4 tipos de acordeones: horizontal que requiere clic para mostrar contenidos de cada ficha, horizontal autoprogramable para que vayan abriéndose cada una de las pestañas, horizontal combinado con un ítem vertical y vartical que también funciona al hacer clic sobre la pestaña.

Guión 1: desarrolloweb.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>Desarrollo web</title>
  <link rel="stylesheet" href="css/accordion.css" />
  <script
src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
  <script src="js/jquery.msAccordion.js"></script>
</head>
<body>
<header>
  <h1>Horizontal Accordion</h1>
</header>
<section>
<article>
<div id="accordionGiftLelo">
  <div class="set">
    <div class="title">
      
    </div>
    <div class="content">
      <a href="http://www.giftlelo.com/29-embroidered-devil-eyes-4819.html">
        
      </a>
    </div>
  </div>
  <div class="set">
    <div class="title">
      
    </div>
    <div class="content">
      <a href="http://www.giftlelo.com/28-spring-embroidery-4797.html">
```

```

        
    </a>
</div>
</div>
<div class="set">
    <div class="title">
        
    </div>
    <div class="content">
        <a href="http://www.giftlelo.com/26-charming-fairy-4737.html">
            
        </a>
    </div>
</div>
<div class="set">
    <div class="title">
        
    </div>
    <div class="content">
        <a href="http://www.giftlelo.com/25-geometrical-embroidery.html">
            
        </a><br />
    </div>
</div>
<div class="set">
    <div class="title">
        
    </div>
    <div class="content">
        <a href="http://www.giftlelo.com/20-elephant-with-luck.html">
            
        </a>
    </div>
</div>
<div class="set">
    <div class="title"></div>
    <div class="content"><a href="http://www.giftlelo.com/19-lady-in-red.html"></a></div>
</div>
<div class="set">
    <div class="title"></div>
    <div class="content"><a
href="http://www.giftlelo.com/17-fireworks-embroidery-art.html"></a></div>
</div>
</div>
</article>
</section>

<header>
    <h1>Horizontal Accordion - Automatic Delay 4 sec</h1>
</header>
<section>
<article>
<div class="accordionWrapper" id="accordion1">
<div class="set set1">
    <div class="title"></div>

```

```

        <div class="content"></div>
</div>
<div class="set set2">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set3">
    <div class="title"></div>
    <div class="content">
    </div>
</div>
<div class="set set4">
    <div class="title"></div>
    <div class="content"><br />
</div>
</div>
<div class="set set5">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set6">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set7">
    <div class="title"></div>
    <div class="content"></div>
</div>
</div>
</article>
</section>

<header>
    <h1>Nested Accordion - Click on set 2</h1>
</header>
<section>
<article>
<div class="accordionWrapper" id="accordionNested">
<div class="set set1">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set2">
    <div class="title"></div>
    <div class="content">
        <div id="accordionNestedChild">
            <div class="set set1">
                <div class="title"></div>

```

```

        <div class="content"><a
href="http://www.giftlelo.com/29-embroidered-devil-eyes-4819.html"></a></div>
    </div>
    <div class="set set2">
        <div class="title"></div>
        <div class="content"><a
href="http://www.giftlelo.com/28-spring-embroidery-4797.html"></a></div>
    </div>
    <div class="set set3">
        <div class="title"></div>
        <div class="content"><a
href="http://www.giftlelo.com/26-charming-fairy-4737.html"></a>
</div>
    </div>
</div>
</div>
</div>
<div class="set set3">
    <div class="title"></div>
    <div class="content">
    </div>
</div>
<div class="set set4">
    <div class="title"></div>
    <div class="content"><br />
</div>
</div>
<div class="set set5">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set6">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set7">
    <div class="title"></div>
    <div class="content"></div>
</div>
</div>
</article>
</section>
<header>
    <h1>Vertical Accordion</h1>
</header>
<section>
<article>
<div id="accordion3" class="accordionWrapper" style="width:486px;">
<div class="set set1">

```



```

        <div class="title"></div>
        <div class="content"></div>
</div>
<div class="set set2">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set3">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set4">
    <div class="title"></div>
    <div class="content"><br />
</div>
</div>
<div class="set set5">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set6">
    <div class="title"></div>
    <div class="content"></div>
</div>
<div class="set set7">
    <div class="title"></div>
    <div class="content"></div>
</div>
</div>
</article>
</section>
<script>
$(document).ready(function() {
    $("#accordionGiftLelo").msAccordion({defaultid:1});
    $("#accordion1").msAccordion({defaultid:6, autodelay:4});
    $("#accordionNested").msAccordion({defaultid:2});
    $("#accordionNestedChild").msAccordion({defaultid:2, vertical:true});
    $("#accordion3").msAccordion({vertical:true});
})
</script>
</body>
</html>

```

Guión 2: jquery.msAccordion.js

```

//menu Accordion
//author: Marghoob Suleman
//Date: 05th Aug, 2009
//Version: 1.0
//web: www.giftlelo.com | www.marghoobsuleman.com
;(function($){
    $.fn.msAccordion = function(options) {
        options = $.extend({
            currentDiv:'1',
            previousDiv:'',
            vertical: false,
            defaultid:0,
            currentcounter:0,
            intervalid:0,
            autodelay:0,

```

```

        event:"click",
        alldivs_array:new Array()
    }, options);
    $(this).addClass("accordionWrapper");
    $(this).css({overflow:"hidden"});
    //alert(this);
    var elementid = $(this).attr("id");
    var allDivs = this.children();
    if(options.autodelay>0) {
    $("#"+ elementid +" > div").bind("mouseenter", function(){
        pause();
    });
        $("#"+ elementid +" > div").bind("mouseleave", function(){
            startPlay();
        });
    }
    //set ids
    allDivs.each(function(current) {
var iCurrent = current;
var sTitleID = elementid+"_msTitle_"+(iCurrent);
var sContentID = sTitleID+"_msContent_"+(iCurrent);
var currentDiv = allDivs[iCurrent];
var totalChild = currentDiv.childNodes.length;
var titleDiv = $(currentDiv).find("div.title");
titleDiv.attr("id", sTitleID);
var contentDiv = $(currentDiv).find("div.content");
contentDiv.attr("id", sContentID);
options.alldivs_array.push(sTitleID);
//$("#"+sTitleID).click(function(){openMe(sTitleID);});
        $("#"+sTitleID).bind(options.event,
function(){pause();openMe(sTitleID);});
    });
    //make vertical
    if(options.vertical) {
        makeVertical();
    };
    //open default
    openMe(elementid+"_msTitle_"+options.defaultid);
    if(options.autodelay>0) {startPlay();};
    //alert(allDivs.length);
    function openMe(id) {
        var sTitleID = id;
        var iCurrent =
sTitleID.split("_")[sTitleID.split("_").length-1];
        options.currentcounter = iCurrent;
        var sContentID = id+"_msContent_"+iCurrent;
        if($("#"+sContentID).css("display")==="none") {
            if(options.previousDiv!="") {
                closeMe(options.previousDiv);
            };
            if(options.vertical) {
                $("#"+sContentID).slideDown("slow");
            } else {
                $("#"+sContentID).show("slow");
            }
            options.currentDiv = sContentID;
            options.previousDiv = options.currentDiv;
        };
    };
    function closeMe(div) {
        if(options.vertical) {
            $("#"+div).slideUp("slow");
        } else {
            $("#"+div).hide("slow");
        };
    };
};

```

```

function makeVertical() {
    $("#"+elementid +" > div").css({display:"block", float:"none",
clear:"both"});
    $("#"+elementid +" > div > div.title").css({display:"block",
float:"none", clear:"both"});
    $("#"+elementid +" > div > div.content").css({clear:"both"});
};
function startPlay() {
    options.intervalid = window.setInterval(play,
options.autodelay*1000);
};
function play() {
    var sTitleId = options.alldivs_array[options.currentcounter];
    openMe(sTitleId);
    options.currentcounter++;
    if(options.currentcounter==options.alldivs_array.length)
options.currentcounter = 0;
};
function pause() {
    window.clearInterval(options.intervalid);
};
}
})(jQuery);

```

Resultado:



IMPORTANTE: Los scripts jQuery necesarios para el funcionamiento del ejemplo, estarán incluidos en los recursos de la guía de práctica.

Ejemplo #5: El siguiente ejemplo ilustra cómo utiliza el plugin fullPage de jQuery para crear un sitio web con navegación en una sola página, cargando todo el contenido y usando la funcionalidad del plugin para mostrar el contenido seleccionado en el menú con un efecto de desplazamiento.

Guión 1: home.html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <title>Frameworks de JavaScript</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="css/jquery.fullPage.css">
    <link rel="stylesheet" href="css/examples.css">
    <link rel="stylesheet" href="css/menu.css">
    <link rel="stylesheet" href="css/sections.css">
    <!--[if IE]>
        <script type="text/javascript">
            var console = { log: function() {} };
        </script>
    <![endif]-->
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
    <script
src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.9.1/jquery-ui.min.js"></script
>

```

```

<script src="jquery/jquery.fullPage.js"></script>
<script src="jquery/example.js"></script>
<script>
    $(document).ready(function() {
        $('#fullpage').fullpage({
            sectionsColor: ['#1bbc9b', '#4BBFC3', '#7BAABE',
'#648CD2', '#5673A3', '#ccddff'],
            anchors: ['firstPage', 'secondPage', '3rdPage',
'4thpage', '5thpage', 'lastPage'],
            menu: '#menu'
        });
    });
</script>
</head>
<body>
<nav id="cssmenu">
<ul id="menu">
    <li data-menuanchor="firstPage">
        <a href="#firstPage">Backbone.js</a>
    </li>
    <li data-menuanchor="secondPage">
        <a href="#secondPage">Angular.js</a>
    </li>
    <li data-menuanchor="3rdPage">
        <a href="#3rdPage">Prototype</a>
    </li>
    <li data-menuanchor="4thpage">
        <a href="#4thpage">Ext JS</a>
    </li>
    <li data-menuanchor="5thpage">
        <a href="#5thpage">Ember</a>
    </li>
</ul>
</nav>
<div id="fullpage">
    <div class="section " id="section0">
        <h1>Backbone.js</h1>
        <p>
            Backbone.js es el framework preferido por muchos. En todos lados
escucharás
            que están hablando de Backbone, muchas empresas importantes lo están
utilizando.
            Esté framework lo conocí por primera vez hace unos 2 años cuándo
trabajaba
            en Softtek y ahora de nueva cuenta lo estoy retomando en la empresa
actual
            que trabajo.
        </p>
        <p>
            <strong>Ventajas:</strong> La comunidad es fuerte y está creciendo
bastante.
            Underscore.js (que se utiliza en gran medida) es también un gran
framework.
        </p>
        <p>
            Desventajas: Carece de abstracciones fuertes y deja mucho que desear.
El framework
            entero es sorprendentemente ligero y da lugar a que sea repetitivo.
Cuanto más
            grande es una aplicación, más se hace evidente.
        </p>
        
    </div>
    <div class="section" id="section1">
        <div class="slide">

```

```

        <div class="intro">
        <h1>Angular.js</h1>
        <p>
            Angular.js es muy buen framework. Desarrollado por Googlers,
tiene algunas
            opciones de diseño muy interesantes.
        </p>
        <p>
            Ventajas: Muy bien pensado, con respecto a las plantillas y
diseño del
            controlador. Cuenta con un sistema de inyección de dependencia
(soy un
            gran fan de la IOC). Tiene gran soporte de UI-Binding y sin
duda su sintaxis
            es muy fácil de entender.
        </p>
        <p>
            Desventajas: El código base parece ser bastante extenso y no
muy modular.
            Las vistas no son lo suficientemente modulares.
        </p>
        
        </div>

    </div>
    <div class="slide">
        <div class="intro">
            
            <h1>¿Qué es Angular.JS?</h1>
            <p>
                Una tecnología nueva en la capa de cliente que nos permite
realizar
                cosas chachis en nuestras páginas sin necesidad de montar
jaleos de
                código. Separa muy bien la responsabilidad de cada tecnología
en su
                ámbito: CSS, HTML y Javascript, y las comunica cuando lo
considera
                necesario.
            </p>
        </div>

    </div>
    <div class="slide">
        <div class="intro">
            
            <h1>¿Por qué Angular.JS?</h1>
            <p>
                Característica principal. No manosees el árbol DOM para acceder
al valor
                de un elemento.
            </p>
            <pre>
            <code class="objectivec">
                <span class="comment">// Esto noooooo por favooooor!</span>
                var titulo = $(<span class="string">"#elem"</span>)
                <span class="variable">.up</span>
                (<span class="string">"li"</span>)
                <span class="variable">.first</span>
                (<span class="string">"p"</span>)
                <span class="variable">.down</span>
                (<span class="string">"span.olala"</span>)
                <span class="variable">.val</span>();
                <span class="comment">// Esto se ve mejor :)</span>
                var titulo = $scope<span class="variable">.titulo</span>;
            </code>
            </pre>

```

```

        </div>
    </div>
<div class="slide">
    <div class="intro">
        
        <h1>Vocabulario</h1>
        <p>
            <strong>Scope</strong>. Es el responsable de detectar los
cambios en
            el modelo y proporciona el contexto a las plantillas.
        </p>
        <pre>
<code class="objectivec">
    &lt;!doctype html&gt;
    &lt;html ng-app&gt;
    &lt;head&gt; ... &lt;/head&gt;
    &lt;body&gt;
    &lt;&lt;div ng-controller="GreetCtrl"&gt;
        &lt;p&gt;Hello {{name}}!&lt;/p&gt;
    &lt;/div&gt;
    &lt;div ng-controller="ListCtrl"&gt;
        &lt;ol&gt;
            &lt;li ng-repeat="name in names"&gt;{{name}}&lt;/li&gt;
        &lt;/ol&gt;
    &lt;/div&gt;
    &lt;/body&gt;
    &lt;/html&gt;
    </code>
    </div>
</div>
<div class="section" id="section2">
    <div class="intro">
        <h1>Prototype</h1>
        <p>
            Prototype es un framework que facilita el desarrollo de
aplicaciones web
            con JavaScript y AJAX. Su autor original es Sam Stephenson, aunque
las
            últimas versiones incorporan código e ideas de muchos otros
programadores.
            A pesar de que incluye decenas de utilidades, la librería es
compacta y
            está programada de forma muy eficiente.
        </p>
        
    </div>
</div>
<div class="section" id="section3">
    <div class="intro">
        <h1>Ext JS</h1>
        <p>
            Ext JS (pronunciado como "ekst"1 ) es una biblioteca de
JavaScript para
            el desarrollo de aplicaciones web interactivas usando
tecnologías como
            AJAX, DHTML y DOM. Fue desarrollada por Sencha.
        </p>
        <p>
            Originalmente construida como una extensión de la
biblioteca YUI por
            Jack Slocum, en la actualidad puede usarse como extensión
para las biblioteca
            jQuery y Prototype. Desde la versión 1.1 puede ejecutarse
como una aplicación
            independiente.
        </p>
    </div>
</div>

```

```

        </p>
      </div>
      
    </div>
    <div class="section" id="section4">
      <div class="intro">
        <h1>Ember</h1>
        <p>
          Nos parece que Ember.js es un framework súper interesante al que no
          conviene
          perder de vista. Se basa en una filosofía de Convention over
          Configuration
          (que traduciremos por algo equivalente a "seguir las convenciones
          en vez de
          configurar"), lo que permite al framework asumir de forma
          automática muchas
          tareas que en otros frameworks nos tocaría programar
          explícitamente.
        </p>
        
      </div>
    </div>
  </div>
</body>
</html>

```

Guión 2: example.js

```

$(document).ready(function() {
  $("#demosMenu").change(function() {
    window.location.href = $(this).find("option:selected").attr("id") +
    '.html';
  });
});

```

Ejercicio #6: El siguiente ejemplo muestra el uso de otro plugin de jQuery (easyAccordion) que permite mostrar contenidos en forma de acordeón, pero con contenido diverso; es decir, texto con imágenes en este caso.

Guión 1: serviciosrdb.html

```

<!DOCTYPE html>
<html lang="es">
<head>
  <title>Servicios de Radio Don Bosco &copy;</title>
  <!-- Meta -->
  <meta charset="utf-8" />
  <!-- Scripts -->
  <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"></script>
  <script src="scripts/jquery.easyAccordion.js"></script>
  <script src="scripts/utility.js"></script>
  <!-- CSS Styles -->
  <link rel="stylesheet" media="screen"
href="css/accordion.css">
</head>
<body>
<div class="sample">
  <h1>Servicios de Radio Don Bosco</h1>
  <div id="accordion-1">
    <dl>
      <dt>Cu&ntilde;as Publicitarias</dt>
      <dd>
        <h2>Cu&ntilde;as Publicitarias</h2>
        <p>
          

```

```

        Creamos cu&ntilde;as publicitarias adaptadas a las necesidades
concretas
        de nuestros clientes. Apostamos por cu&ntilde;as publicitarias
vanguardistas,
        producciones de gran calidad, donde el mensaje, la voz del
locutor, la
        m&uacute;sica sean capaces de transmitir m&aacute;s emociones y
despertar
        el inter&eacute;s del consumidor.
        </p>
    </dd>
    <dt>Locuciones para PBX y centrales telef&oacute;nicas</dt>
    <dd>
        <h2>Locuciones para PBX y centrales telef&oacute;nicas</h2>
        <p>
            
            Realizamos todo tipo de locuciones profesionales para sus
proyectos de v&iacute;deo
            y comunicaci&oacute;n audiovisual. Nos adaptamos a sus guiones,
duraci&oacute;n de secuencias
            y entregamos el material en uno o varios archivos de audio para
facilitar
            la labor de montaje y mezcla en la edici&oacute;n.
        </p>
    </dd>
    <dt>Locuciones en OFF para v&iacute;deos</dt>
    <dd>
        <h2>Locuciones en OFF para v&iacute;deos</h2>
        <p>
            
            Realizamos todo tipo de locuciones profesionales para sus
proyectos de
            v&iacute;deo y comunicaci&oacute;n audiovisual. Nos adaptamos a
sus guiones,
            duraci&oacute;n de secuencias y entregamos el material en uno o
varios archivos
            de audio para facilitar la labor de montaje y mezcla en la
edici&oacute;n.
        </p>
    </dd>
    <dt>Locuciones en ingl&eacute;s</dt>
    <dd>
        <p>
            Le ofrecemos servicios de locuci&oacute;n en espa&ntilde;ol e
ingl&eacute;s.
            Grabamos sus locuciones en los formatos que solicite; wave,
mp3, entre otros.
        </p>
    </dd>
    <dt>Sincron&iacute;a de audio con v&iacute;deo</dt>
    <dd>
        <h2>Sincron&iacute;a de audio con v&iacute;deo</h2>
        <p>
            
            Si el cliente lo solicita, nos encargamos de sincronizar el
audio con las
            im&aacute;genes de v&iacute;deo. Nos ajustamos de forma exacta
y precisa
            al gui&oacute;n que el cliente nos facilite. De esta forma, el
realizador
            s&oacute;lo tendr&aacute; que integrar el audio, ya mezclado
(con m&uacute;sica,
            voz efectos) y ajustado con la imagen.
        </p>
    </dd>
    <dt>Producciones radiof&oacute;nicas</dt>

```



```

        <dd>
            <h2>Producciones radiof&ocute;nicas</h2>
            <p>
                
                Contamos con un staff de locutores y productores, tanto
t&eacute;cnicos
                como literarios, lo cual nos permite ofrecer a nuestros
clientes productos
                de alta calidad que cubren sus expectativas. Nos caracteriza el
profesionalismo
                y la prontitud en la entrega de los productos. Pregunte por
nuestros paquetes
                y tarifas especiales.
            </p>
        </dd>
    </dl>
</div>
</div>
</body>
</html>

```

Guión 2: utility.js

```

$(document).ready(function () {
    $('#accordion-1').easyAccordion({
        autoStart: false,
    });

    $('#accordion-2').easyAccordion({
        autoStart: false
    });

    $('#accordion-3').easyAccordion({
        autoStart: true,
        slideInterval: 5000,
        slideNum: false
    });

    $('#accordion-4').easyAccordion({
        autoStart: false,
        slideInterval: 5000
    });
});

```

IMPORTANTE: El resto de scripts js y hoja de estilos necesarios para el funcionamiento del ejemplo, serán proporcionados en los recursos de esta guía de práctica.

Resultado:



V. ANÁLISIS DE RESULTADOS

1. En el ejemplo del menú de navegación (ejemplo 3 del procedimiento de esta guía de práctica) la navegación por cada ejemplo se realiza con un enlace a páginas distintas. Modifique este comportamiento para que la navegación se pueda realizar con AJAX. Queda a su criterio si utiliza funciones o clases JS de la guía de práctica anterior o si utiliza alguna librería o funcionalidad de jQuery incorporada para lograr este comportamiento.
2. Examine el sitio web de ejemplo: <http://www.jque.re/plugins/image-galleries-decks/ascensor/#/Home> que utiliza una librería de jQuery llamada ascensor.js y cree una página web utilizando este enfoque en donde tenga una cuadrícula de páginas como la siguiente:



Las casillas con colores representan las páginas. Seleccione ud. un tema para desarrollar en dicho sitio web tomando como ejemplo el ejercicio 6 del procedimiento de esta guía de práctica.

VI. BIBLIOGRAFÍA

- Luc Van Lancker. jQuery El Framework JavaScript de la web 2.0. 2da Edición. Eni Ediciones. 1ra Edición edición Digital. Barcelona, España.
- David Sawyer McFarland. Programación JavaScript y jQuery. Anaya Multimedia.