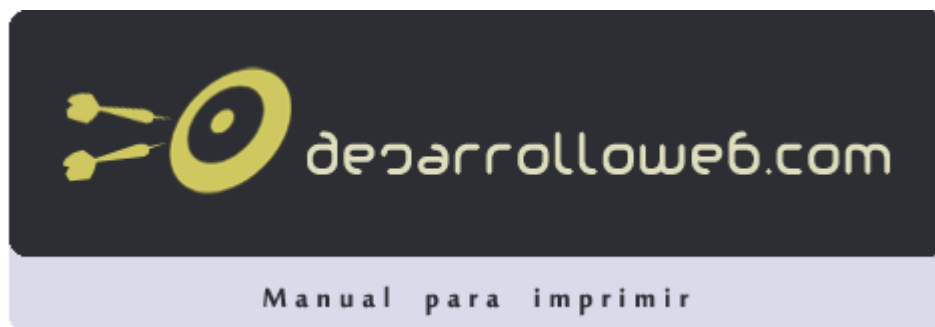


# Manual de jQuery Mobile

*Manual de jQuery Mobile, el framework para el desarrollo compatible de sitios web orientados a dispositivos móviles, basado en el popular jQuery.*



## Autores del manual

Este manual ha sido realizado por los siguientes colaboradores de DesarrolloWeb.com:

**Miguel Angel Alvarez**  
Director de DesarrolloWeb.com  
<http://www.desarrolloweb.com>  
(3 capítulos)

**Dairo Galeano**  
Desarrollador independiente  
(10 capítulos)

## Introducción a jQuery Mobile

*Características de jQuery Mobile, el framework Javascript para desarrollo de aplicaciones web para móviles, basado en jQuery.*

jQuery Mobile Framework es un producto que sin duda va a dar mucho que hablar en los próximos años. Estamos asistiendo a la revolución de los dispositivos móviles de acceso a Internet y este framework, basado en el popular jQuery, se va a convertir en el mejor aliado para el desarrollo de sitios web orientados para al segmento de consumidores web en movilidad.

En esta introducción a jQuery Mobile voy a ofrecer las claves o características de este framework, de modo que las personas puedan saber qué tipo de ayudas nos ofrece para el desarrollo y por qué va a significar una revolución a la hora de crear aplicaciones web compatibles con la mayoría de las plataformas móviles.

### jQuery se reinventa a si mismo

Quizás recordemos el lema o slogan de jQuery: "Write Less, Do More". Pues jQuery Mobile Framework es esa misma idea, pero elevado a la siguiente potencia.

Para explicar esto quiero antes preguntar a los lectores ¿Para qué sirve un framework? Supongo que la mayoría lo sabrá de sobra, pero comento básicamente estas dos situaciones:

1. No tener que lidiar con las particularidades de cada navegador. Desarrollar una vez con código jQuery y que se vea correctamente en todos los navegadores del mercado. Incluso, cuando saquen otro navegador, o versiones nuevas de los existentes, que no tengas que retocar tu código para adaptarlo también a ellos.
2. También sirve para escribir menos código fuente y hacer cosas más espectaculares.

Ahora ¿Por qué jQuery Mobile es otra vuelta de tuerca?

1. Porque **con los dispositivos móviles se han multiplicado el número de navegadores y de plataformas**. Tenemos muchos fabricantes, de tablets y smartphones y diversos dispositivos con características distintas, como tamaños de pantallas, sistemas operativos diferentes y diversos navegadores basados en cada uno de ellos. Vamos, que si antes con los navegadores para PCs había problemas de compatibilidad, cuando teníamos apenas 3 sistemas operativos y 3 navegadores populares, ahora con los móviles la cosa todavía se hace más complicada.
2. Porque el **desarrollo de sitios web con jQuery Mobile es todavía más automático** de lo que era trabajar con jQuery. Con mucho menos código haces muchas más cosas.

Como te puedes figurar, con los dispositivos móviles todavía se hace más necesario usar un framework, que te facilite desarrollar una vez y ejecutar de manera compatible en todos los dispositivos. Además de ello, con jQuery Mobile reducirás drásticamente el tiempo de desarrollo de tu sitio web para dispositivos de movilidad.



## Características de jQuery Mobile

Lo primero que debe quedarnos claro para describir jQuery Mobile es que no se trata de un nuevo framework creado desde cero. En absoluto. Podemos entender a jQuery Mobile como un plugin para jQuery puesto que es un producto que está **basado en el propio framework Javascript jQuery**. Por tanto, aquellas personas que ya conocen jQuery no van a tener que aprender nada nuevo, sino aplicar sus conocimientos y desarrollar fácilmente aplicaciones para móviles. Esto significa una gran ventaja, puesto que hay muchas personas que conocen jQuery y que van a poder pasarse sin prácticamente ningún problema al desarrollo para dispositivos.

Lo segundo que quiero señalar es que, si bien jQuery nos sirve para desarrollo de funcionalidades Javascript compatibles con todos los navegadores, jQuery Mobile va un poco más allá. No se trata simplemente de una capa para realizar código Javascript que funcione en todos los navegadores, sino un conjunto de **herramientas que simplificará el proceso de crear páginas para móviles**, desde la escritura del propio código HTML, la maquetación con CSS y la creación de efectos dinámicos con Javascript.

Un momento: ¿como nos va a ayudar a hacer HTML jQuery Mobile? No es que vayas a escribir código HTML desde jQuery, sino que, al incluir jQuery Mobile tu código HTML básico será optimizado para realizar diversos comportamientos dinámicos en los navegadores móviles, de manera automática. Además, muchas cosas del propio framework las vas a poder configurar directamente a través de atributos HTML.

Bueno ¿Y qué hay de CSS? Pues tampoco es que tengas que escribir tu código CSS desde jQuery Mobile, sino que este framework dispondrá diversas herramientas CSS, también de manera automática, que podrás utilizar en tus desarrollos. Por poner dos ejemplos, con este framework tus componentes de formularios se estilizarán de manera automática, sustituyendo los feos elementos nativos de los navegadores de dispositivos móviles. Además, dispondrás de un pequeño framework CSS para poder hacer cosas como la maquetación a partir de una rejilla.

Aparte de todo lo comentado hasta ahora, que opino es clave para entender bien qué es jQuery Mobile Framework, aquí van otra serie de características rápidas del producto:

- **Creado sobre jQuery con arquitectura de jQueryUI:** Los propios creadores de jQuery usaron su experiencia para desarrollar el framework para móviles y además implementaron la arquitectura diseñada para las librerías de interfaces de usuario jQueryUI. Por tanto se trata de un producto muy bien pensando, en base a la experiencia de años.
- **Está desarrollado para trabajar con HTML5:** de hecho, estamos obligados a hacer páginas HTML5 para aprovechar todas las características del framework.
- **Repleto de automatismos:** Si ya era fácil hacer Ajax en jQuery, todavía es más fácil en jQueryMobile. De hecho, si el framework capta que puede hacer una conexión Ajax en lugar de una convencional, lo hace automática por Ajax. Y eso es solo un ejemplo, también son automáticas las transiciones entre páginas, la personalización del aspecto de la página, etc.
- **Preparado para dispositivos táctiles:** Los dispositivos táctiles tienen cambios en la gestión de eventos y jQuery Mobile nos facilita la labor de adaptarnos a ellos.
- **Personalización de temas:** Igual que ocurría con las jQueryUI, el jQuery Mobile podemos elegir entre varios temas gráficos ya listos para aplicar al aspecto de nuestra página. Además, podemos crear nuestros propios temas personalizados.

## Compatible con el mayor número de plataformas

Los creadores del framework comentan entre sus características que se han esforzado para cubrir el mayor número de plataformas de dispositivos móviles posible. Dicen que el target que han buscado es mayor que el del resto de frameworks disponibles en el mercado.

**Nota:** A decir verdad, no he comprobado hasta que punto la compatibilidad de jQuery Mobile sea mayor o menor que la de otros productos como Sencha Touch. Pero si nos vamos a la página de Sencha veremos que ellos comentan que el framework está preparado para iOS, Android y BlackBerry, cuando jQuery Mobile alcanza muchas otras plataformas.

En la siguiente imagen podemos ver los logos de los sistemas operativos que jQuery Mobile soporta.



No obstante, cabe señalar existen diversos grados de compatibilidad para cada sistema, o mejor dicho, para cada navegador dentro de cada familia de dispositivos. En la documentación del framework, en la sección de Supported Platforms veremos que el grado de compatibilidad está dividido en tres niveles distintos, desde Grado-A (donde están la mayoría navegadores para iOS y Android, así como BlackBerry, Palm WebOS, los navegadores de ordenadores de escritorio, etc.) a Grado-B (donde encontramos a Symbian, Opera Mini 5.0 y 6.0 para iOS o Balckberry 5.0) o Grado-C (con el resto de los smartphones, entre los que se encuentra Windows Mobile o Blackberry 4).

En resumen, que según apuntan en la documentación, solo se ha dejado sin soporte deliberadamente en esta versión 1.0 del framework el sistema Samsung Bada (El sistema operativo propietario de Samsung para smartphones), aunque dicen que probablemente funcione relativamente bien incluso sin haberla probado, pues todavía no hay dispositivos o emuladores.

En los próximos artículos de DesarrolloWeb.com empezaremos a desarrollar con jQuery Mobile, comenzando por las [explicaciones sobre cómo hacer una primera página básica](#). De momento os dejo un par de enlaces que puedan complementar esta información:

- Como no, es imprescindible dar el enlace a este framework, para que podáis ir conociéndolo: <http://jquerymobile.com>
- Además, una referencia interesante a un sitio donde se puede ver una [galería de sitios creados con jQuery Mobile](#).

## Introducción a jQuery Mobile en vídeo

Hemos realizado una retransmisión en directo que puede servir para poder tener una visión general de jQuery Mobile y para dar los primeros pasos con este framework para el desarrollo de sitios para dispositivos móviles. En la introducción se explican cosas importantes como qué es jQuery Mobile, cómo insertarlo en una página web o qué relación tiene con el proyecto jQuery global. Además se comienza a codear con diversos ejemplos en los que se puede apreciar la estructura de la página básica, la estructura multipágina y la personalización de elementos del HTML con distintos roles.

Artículo por Miguel Angel Alvarez

## Configuraciones por defecto de jQuery Mobile

*Breve introducción a las configuraciones predeterminadas que vienen instaladas con jQuery Mobile y todo lo que ellas involucran.*

Una de las grandes ventajas que tiene jQuery Mobile framework, es que se auto-inicia. Es el mismo *framework* quien se encarga de crear cada uno de los *widgets* que nosotros le indiquemos a través del atributo "data-role". Esto es una gran ventaja ante cualquier framework que tenga como labor principal la de crear interfaces de usuario, pues todo se hace casi de manera transparente para el desarrollador.



Como te habrás percatado, aún a estas instancias del [Manual de jQuery Mobile](#) no se ha visto ni una sola línea de Javascript, pero lo mejor es que hemos logrado la constitución de importantes y relevantes herramientas para cualquier aplicación web móvil. Sin embargo, con lo anterior no pretendemos decir que jQuery Mobile es solo para crear widgets, sino que tiene muchos más soportes. Es más, podemos hacer algunos trabajos de AJAX con el framework, así como cualquier otro tipo de interacción usuario/sitio, de manera muy simple.

Hablando un poco mas de AJAX, creo que hemos comentado anteriormente que el propio framework también lo maneja de forma automática, haciendo incluso por nosotros tratamientos especiales frente a errores, por ejemplo. En este sentido, si llega a producirse algún problema con una transacción HTTP, o cualquier otro protocolo, jQuery Mobile lo detectará, enviando al usuario un "Toask" como aviso, con un mensaje del problema. También procesa los eventos relacionados con la carga de los datos que se pueden realizar a través de cualquier petición que se haga al servidor, mostrándonos de la misma forma un Toask de de carga.

**Nota:** Para aclarar dudas, quiero decir que un Toask es como una especie de "alert", que aparece encima de las aplicaciones, con la deferencia de que éste no usa botones, se oculta de forma automática, se usan en los dispositivos móviles y páginas web de forma momentánea.

## Mención de las opciones por defecto de jQuery Mobile

En la documentación oficial de jQuery Mobile encontramos un texto que, en síntesis, dicta lo siguiente:

*"Para comenzar, basta decir que cualquier modificación que se quiera hacer sobre las configuraciones de jQuery Mobile se deben hacer envueltas dentro del evento "mobileinit", puesto que no se trabaja con el tradicional evento "ready" que se aplica al elemento document en otros proyectos relacionados con jQuery. Esto se debe a que la auto-inicialización de jQuery Mobile se lleva a cabo antes de ejecutarse el ultimo evento mencionado".*

Pero en muchos de los casos es más productivo no incluir las configuraciones dentro del evento "mobileinit", aunque así lo diga la documentación de jQuery Mobile. Esto es debido a que, en ocasiones, hemos probado a modificar las opciones, pero en realidad no se obtienen resultados visibles de ello, algo que sí se logra haciendo las diferentes configuraciones fuera del evento "mobileinit" y dentro del "ready".

Cuando hablamos de estas configuraciones por defecto de jQuery Mobile, estamos hablando de una gran lista, pero también debo decir que en este artículo no las vamos a mencionar todas, pues algunas son solo para casos demasiado especiales y donde quizás pocas aplicaciones realmente lo exijan.

## Cambiar las transiciones por defecto de páginas y diálogos

Cuando nos movemos entre una página y otra en una aplicación jQuery Mobile, nos damos cuenta de que por defecto hace el efecto de transición tipo "pop", pero como la cosa va sobre gustos, en mi caso me gusta más la transición "flip", entonces sería demasiado pedir hacer un data-transition a cada uno de los enlaces. Para eso disponemos de una opción para cambiar las transiciones de todas las paginas al mismo tiempo, cosa que se consigue con el siguiente código:

```
$.mobile.defaultPageTransition='flip';
```

Con esto todas nuestras páginas harán la transición flip.

**Nota:** Recuerda que en el [artículo de la barras de navegación](#) hablamos sobre las transiciones y explicábamos que "flip" no funciona en todos los navegadores.

También existe una forma muy parecida para cambiar las transiciones de los cuadros de diálogo, así tendremos una mejor animación unificada no sólo para las páginas, sino también para los cuadros de diálogo.

```
$.mobile.defaultDialogTransition='slideup';
```

## Modificar y Mostrar los Toask relacionados con AJAX

Anteriormente, en este artículo mencionábamos algo sobre los Toask y los trabajos que realizaban en jQuery Mobile, pues ahora vamos a ver como modificamos el mensaje que traen por defecto.

Para el caso del mensaje de carga necesitaríamos un código como el siguiente.

```
$.mobile.loadingMessage = "Hemos modificado el mensaje del Loading";
```

Ahora si lo que queremos es cambiar el mensaje que viene dentro del Toask, que aparece cuando hay un error, necesitamos el siguiente código:

```
$.mobile.pageLoadErrorMessage = 'Nuevo mensaje';
```

También se puede controlar la aparición y desaparición de los Toask. Digamos que quiero que, al presionar un botón, aparezca el mensaje de carga. Entonces escribimos un código como este:

```
$.mobile.loadingMessage = "Hemos modificado el mensaje del Loading";
$.mobile.showPageLoadingMsg('');
```

Luego para ocultar el Toask lo hacemos con una sencilla línea de código como esta.

```
$.mobile.hidePageLoadingMsg();
```

Recuerda que si quieres tener una referencia más amplia de las configuraciones por defecto en jQuery Mobile Framework, puedes visitar su documentación, en la sección de configuraciones por defecto [jquerymobile.com/demos/1.0/docs/api/globalconfig.html](http://jquerymobile.com/demos/1.0/docs/api/globalconfig.html)

## Poniendo todo en un solo ejemplo

Ahora vamos a compartir un código de ejemplo en donde se ilustra todo lo explicado en este artículo:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Configuraciones por defecto</title>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
<script src="jquery-1.7.1.min.js"></script>
<script src="jquery.mobile-1.0.min.js"></script>
<script>

$(document).ready(function() {
$.mobile.pageLoadErrorMessage = 'Esta pagina no existe, el mensaje de error ha sido cambiado';
$.mobile.fixedToolbars.show(true);
$.mobile.defaultDialogTransition='slideup';
$.mobile.defaultPageTransition='flip';
$("#mostrar").click(function(e) {
$.mobile.loadingMessage = "Hemos modificado el mensaje del Loading";
$.mobile.showPageLoadingMsg('');
});

$("#ocultar").click(function(e) {
$.mobile.hidePageLoadingMsg();
});

});
```

```
});

</script>
</head>

<body>
<div data-role="page" id="home">
<div data-role="header" data-position="fixed">
<h1>Configuraciones</h1>
</div>
<div data-role="content">

<a href="#" id="mostrar" data-role="button">Mostrar Loading</a>
<a href="#" id="ocultar" data-role="button">Ocultar Loading</a>
<a href="#pag2" data-role="button" >Probar nueva transicion de paginas</a>
<a href="#dialogo" data-role="button" data-rel="dialog">Probar nueva transicion de Dialogos</a>
<a href="demo.html" data-role="button" >Aqui hay un error, la pagina del vinculo no existe</a>

</div>
<div data-role="footer">
<h3>Configuraciones</h3>
</div>

</div>

<div data-role="page" id="pag2">
<div data-role="header" data-position="fixed">
<h1>Configuraciones pagina 2</h1>
</div>
<div data-role="content">

<p>Este es una pagina y puedes ver que hace la transicion flip, sin poner el data-transition en el
anlace, todas las paginas tendran la misma transicion</p>

<a href="#home" data-role="button" data-icon="back">Regrsar</a>

</div>
<div data-role="footer">
<h3>Configuraciones pagina 2</h3>
</div>

</div>

<div data-role="page" id="dialogo">
<div data-role="header" data-position="fixed">
<h1>Configuraciones</h1>
</div>
<div data-role="content">
<p>Este es un dialog y puedes ver que hace la transicion slideup, sin poner el data-transition en el
anlace, todos los dialogos tendran la misma transicion</p>
<a href="#home" data-role="button" data-icon="back">Regrsar</a>

</div>
</div>
</body>
</html>
```

Sin son observadores, se darán cuenta de que usamos una configuración no explicada.

```
$.mobile.fixedToolbars.show(true);
```

Con ésta se logra que todas las barras (header/ footer) que tengan el atributo data-position="fixed", se fijen de la misma forma que lo hace la barra de Google, donde podemos hacer *scroll* en la pagina y la barra también bajará, posicionándose siempre en la parte superior de la ventana del documento.

Artículo por [Dairo Galeano](#)



## Página básica en jQuery Mobile Framework

*Cómo es la estructura de la página más básica que podríamos hacer con el framework jQuery Mobile.*

En el anterior artículo donde [presentamos jQuery Mobile](#) me cansé de decir lo sencillo que es desarrollar con este framework y cómo con poco código podemos hacer grandes cosas. Realmente en este artículo no vamos a explotar las enormes posibilidades que nos ofrecen estas librerías, puesto que vamos a hacer una página muy básica, pero al menos servirá para comenzar a apreciar algunas de sus características comentadas anteriormente.

A continuación vamos a realizar nuestra primera página básica, por un lado incluiremos los scripts para instalar el framework, y por otro veremos también cuál es la estructura de código HTML que debemos implementar para crear esta primera página. Todo esto nos servirá para apreciar como se puede hacer una página para móviles con estructura fundamental de cabecera, cuerpo y pie y cómo jQuery Mobile les coloca automáticamente unos estilos básicos.

### Código HTML de la página básica

En este primer ejercicio, en lugar de comentar paso a paso como construir la página, quiero mostrar directamente cuál sería el código HTML necesario para crearla. Esto nos dará una idea global sobre esta estructura de documento y luego comentaré las distintas partes del código.

```
<!DOCTYPE html>
<html>
<head>

  <title>Probando jQuery Mobile</title>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
  <script type="text/javascript" src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
  <script type="text/javascript" src="http://code.jquery.com/mobile/1.0/jquery.mobile-
1.0.min.js"></script>

</head>

<body>
  <div data-role="page">
    <div data-role="header">
      Página para móviles
    </div>
    <div data-role="content">
      Esta es la versión para móviles de mi sitio web
    </div>
    <div data-role="footer">
      Contacta y lo que sea...
    </div>
  </div>
</body>
</html>
```

A continuación podemos ver, parte por parte, comentado todo el anterior código HTML.

**Nota:** Como acabas de ver, la página básica de este framework para móviles incluye, no solo los scripts jQuery y el componente jQuery Mobile, con sus CSS, sino también un bloque de código HTML para crear las partes fundamentales de toda página jQuery Mobile. Esto es porque jQuery mobile, como verás a lo largo de los siguientes artículos, no es solo un framework para crear dinamismos del lado del cliente -como podríamos entender a jQuery-, sino todo un paquete de utilidades que van desde la maquetación de contenidos para móviles, mejoras en la presentación e interfaces de usuario, hasta una serie de métodos para realizar dinamismos e interacción con el usuario.



## Esto es HTML5!

Como se dijo en el artículo anterior, jQuery Mobile funciona con HTML5. Es decir, para poder disponer de todas las funcionalidades del framework debemos escribir código HTML5.

Por ello, podremos apreciar como se define el doctype de HTML5 en la primera línea de código.

```
<!DOCTYPE html>
```

**Nota:** Los entendidos de HTML5 verán también que hemos definido una etiqueta META para especificar el juego de caracteres UTF-8, necesario en esta versión del lenguaje HTML. Recordar que se debe guardar el archivo con ese formato, UTF-8, con vuestro editor de texto para que todo vaya correctamente.

## Etiqueta META Viewport

El segundo punto que nos salta a la vista es la etiqueta META "viewport", que sirve para decirle a jQuery que debe mostrar la página a escala 1.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Esta etiqueta se debe indicar porque algunos dispositivos realizan un escalado automático de la página para mostrarla a unas dimensiones que no tienen por qué ser las deseadas. En algunos casos se utiliza una anchura virtual de 900 píxeles aproximadamente, con lo que nuestra página podría verse escalada. Con esta etiqueta la anchura de la página será igual a la anchura de la pantalla del dispositivo.

Sin embargo, esta definición no entrará en conflicto con la posibilidad de hacer zoom a la página, lo que es interesante desde el punto de vista de la accesibilidad. El único problema de esta etiqueta es que en iOS no se actualiza automáticamente la anchura de la página al cambiar la orientación del dispositivo (de horizontal para vertical o viceversa), con lo que las dimensiones pueden no ajustarse a la pantalla cuando alteramos la posición del móvil. Según comentan en la documentación, queda pendiente corregir este asunto en versiones venideras del framework.

## Incluir los scripts del framework

A continuación vemos todas las etiquetas para instalar jQuery Mobile en nuestra página. Consiste en un link a una hoja de estilos CSS y la inclusión de los scripts Javascript de jQuery y jQuery Mobile.

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
<script type="text/javascript" src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
<script type="text/javascript" src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js"></script>
```

**Nota:** Como se puede ver, estamos incluyendo la versión 1.0 del framework, pero quizás en el momento que leas este artículo estén en una release más moderna, por lo que atento al copiar y pegar.

En la documentación del framework nos recomiendan hacer los enlaces con los scripts alojados en el CDN (Content Delivery Network) de jQuery, para un mejor desempeño. Pero si lo preferimos también podemos tenerlos alojados en nuestro propio servidor.

## Definir la estructura de página y los data-role

A continuación podemos fijarnos en el cuerpo de la página, dentro de la etiqueta BODY. Veremos que la página básica incluye diversas divisiones DIV con distintas utilidades, definidas por los atributos data-role. Estas forman la estructura de la página básica en jQuery Mobile.

Para comenzar veremos que existe una división que engloba toda la estructura de la página básica, con el data-role="page".

```
<div data-role="page">
```

Dentro de esa división tendremos otros DIV con los tres roles fundamentales de toda página jQuery Mobile: cabecera, cuerpo y pie.

La cabecera se indica con el data-role="header".

```
<div data-role="header">
```

El cuerpo de la página lo colocamos con un data-role="content".

```
<div data-role="content">
```

Finalmente, el pie de página se indica con un data-role="footer".

```
<div data-role="footer">
```

Con esto hemos terminado con la página básica. Podemos [ver el ejemplo en funcionamiento](#).

## Conclusión

Lo más seguro es que no te haya sorprendido mucho esta primera página jQuery Mobile. Realmente no tiene nada en especial y ciertamente, si la vemos en un navegador, no observaremos nada que resulte espectacular. No obstante, en el siguiente artículo que publicaremos en Desarrolloweb.com modificaremos esta página básica, para aprender cómo es la estructura de la página multipágina de jQuery Mobile, que ya nos permitirá mostrar algunas de las características interesantes de este framework.

No obstante, quiero que se aprecie que jQuery Mobile ya nos ha alterado un poco el aspecto de nuestra página básica a partir de unos estilos en la cabecera y pie, que son meramente automáticos. Nuevamente, os pido un poco de paciencia para conocer en detalle todas las bondades anunciadas de este framework en futuros artículos.

Artículo por [Miguel Angel Alvarez](#)

## Estructura multipágina de jQuery Mobile

*Cómo hacer una página multipágina, de modo que se puedan reunir varias páginas en un archivo HTML, en jQuery Mobile Framework.*

Continuamos dando nuestros primeros pasos para el desarrollo de páginas orientadas a dispositivos móviles con jQuery Mobile. En esta ocasión nos vamos a dedicar a la estructura de las páginas "multipágina", lo que nos dará ocasión de ver ya algún efectillo interesante en nuestro navegador.

En el anterior artículo del [Manual de jQuery Mobile](#) explicamos cuál es la [estructura de la página básica](#) y ahora veremos que la multipágina no es más que la unión de varias páginas básicas en un mismo archivo HTML.

Quizás alguna persona con conocimientos avanzados de jQuery pueda estar decepcionada, al ver como en este manual de jQuery Mobile no hemos utilizado todavía ningún código Javascript. Si es así lo siento, pero ya advierto que esta situación va a mantenerse al menos durante bastantes artículos más. Pues, si bien es cierto que en esta ocasión vamos a poder ver en marcha efectos dinámicos, no es necesario utilizar Javascript para nada, porque muchos de los comportamientos básicos de este framework se configuran directamente sobre el código HTML, por medio de diversos atributos con el prefijo "data-".

Para hacer la página básica ya vimos que las divisiones tenían un atributo "data-role" para indicar la estructura de la página, con su cabecera, cuerpo y pie. Ahora veremos otros atributos parecidos para indicar otras cosas sobre las páginas que estamos desarrollando.

## Multipágina es la unión de varias páginas básicas

En el mismo código HTML podemos incluir varios bloques de página básica y tendremos un archivo HTML con diversas páginas que se conoce como Multipágina. Pero tranquilos, esto no quiere decir que en jQuery Mobile tengamos que desarrollar siempre todas las páginas del sitio web en el mismo archivo HTML, podemos organizarlas si lo deseamos en archivos aparte, o hacer una mezcla entre páginas multipágina y páginas básicas en diferentes archivos HTML.

La página básica vimos que incluía una serie de scripts Javascript y estilos CSS del framework, además de una declaración META para definir la escala o zoom con la que se debería presentar. Todo eso permanece igual, lo único que cambiará será el cuerpo de la página, donde colocaremos varios bloques data-role="page".

Veamos una unión de varios bloques de páginas básicas.

```
<!-- PORTADA -->
<div data-role="page" id="portada">
  <div data-role="header">
    <h1>Titular de página portada</h1>
  </div>
  <div data-role="content">
    <p>Fabrico guitarras de todos los colores</p>
    <p>Puedes ver <a href="#galeria">mis guitarras</a></p>
    <p>Puedes <a href="#comprar" data-rel="dialog">comprar una guitarra</a></p>
  </div>
  <div data-role="footer">
    Pie de página
  </div>
</div>
<!-- /PORTADA -->

<!-- GALERIA -->
<div data-role="page" id="galeria" data-title="Galería de guitarras">
  <div data-role="header">
    <h1>Galería de fotos de mis guitarras</h1>
  </div>
  <div data-role="content">
    <p>Puedes ver todas mis guitarras</p>
    <p><a href="#portada">volver</a></p>
  </div>
  <div data-role="footer">
    Pie de página
  </div>
</div>
<!-- /GALERIA -->

<!-- COMPRAR -->
<div data-role="page" id="comprar" data-title="Comprar guitarras">
  <div data-role="header">
    <h1>Comprar guitarras</h1>
  </div>
  <div data-role="content">
    <p>Servicio temporalmente indisponible</p>
    <p><a href="#portada">volver</a></p>
  </div>
  <div data-role="footer">
    Pie de página
  </div>
</div>
<!-- /COMPRAR -->
```

Este sitio estaría formado por tres páginas distintas: "portada", "galeria" y "comprar". A pesar que están escritas en el mismo archivo HTML, jQuery se encargará de mostrarlas como si fueran páginas independientes. La navegación entre ellas está realizada por enlaces, como veremos a continuación.

## Enlaces entre las páginas de la estructura multipágina

Cada una de las páginas básicas del esquema multipágina tienen un identificador, que se pone en el DIV principal de la página, el que tiene `data-role="page"`. Veamos a continuación el código HTML de la división general de la página de portada.

```
<div data-role="page" id="portada">
```

Como vemos, se le ha colocado `id="portada"`, con lo que ya tiene un nombre asignado. Ahora, si queremos hacer enlaces dirigidos hacia esta portada, los creamos simplemente como si fueran enlaces internos. Para ello colocamos el `href="#portada"`, es decir, tal como se hace con los enlaces dentro de la misma página que conocemos del HTML.

```
<a href="#portada">Ir a la portada</a>
```

Lo genial de estos enlaces es que, para pasar de una página a otra, jQuery Mobile produce una transición automática, que hace que una página actual desaparezca y entre la enlazada con un efecto suavizado. Más adelante explicaremos cómo configurar las transiciones entre páginas.

## Títulos de las páginas

Como todas las páginas de esta estructura multipágina están en el mismo archivo HTML, ocurre que todas comparten un mismo TITLE, que fue indicado en el HEAD de la página. Sin embargo los desarrolladores probablemente querrán que cada página tenga un título independiente. Para hacer posible ese caso de uso, jQuery Mobile dispone de un atributo `data-title` que se coloca en el DIV principal de cada una de las páginas.

En `data-title` podemos indicar el título que tiene cada página del esquema de multipágina y jQuery Mobile se encargará de actualizarlo automáticamente cuando se navegue a esa página. Podemos ver el título de la sub-página de galería:

```
<div data-role="page" id="galeria" data-title="Galería de guitarras">
```

Así vemos que, cuando el usuario pulse un enlace que le lleve a esta página, se cambiará de manera automática el titular de la ventana del navegador a lo indicado mediante `data-title="Galería de guitarras"`.

En resumen. La primera de las páginas podrá tener el título que indiquemos en el TITLE del documento HTML, pero si queremos asignar otros títulos diferentes a otras páginas de la estructura multipágina, deberemos hacerlo a través de los atributos `data-title`.

## Atributos data-X

Quiero hacer una mención aparte a estos atributos especiales de jQuery Mobile. Como has podido ver, al desarrollar para este framework se utilizan diversos atributos "data-X", como los mencionados `data-role` o `data-title`.

**Nota:** Otro ejemplo que hemos puesto en este artículo de la estructura multipágina es `data-rel="dialog"`, que asignado a un enlace, provoca que la página que se está enlazando se muestre como si fuera un cuadro de diálogo. Más adelante veremos más cosas acerca de los cuadros de diálogo, pero para que se identifique ese `data-rel="dialog"` pongo aquí el código del enlace donde se ha colocado ese atributo.

```
<a href="#comprar" data-rel="dialog">comprar una guitarra</a>
```

Todos estos atributos con prefijo `data-` son perfectamente válidos en la sintaxis HTML5, ya que las especificaciones del lenguaje permiten cualquier número de atributos que comiencen por la cadena "data-". En HTML5 se utilizan para embeber datos de las aplicaciones web en las propias etiquetas del HTML y jQuery Mobile los utiliza, como has podido comprobar, para configurar el funcionamiento de las páginas desarrolladas con estas librerías.

Más adelante veremos otros usos de esta técnica para implementar diversos componentes de interfaces de usuario, configurar las transiciones entre páginas, el comportamiento de los enlaces, etc.

## Conclusión

De momento dejamos por aquí estas explicaciones, ofreciendo el [enlace al ejemplo en marcha donde puedes ver la estructura multipágina](#).

Como podrás comprobar al ver el ejemplo en marcha, se han realizado dos tipos de comportamientos dinámicos. Primero las transiciones entre páginas, con un efecto de desplazamiento suavizado. Luego el cambio automático del TITLE del documento, al pasar de una página a otra. Es simplemente una pequeña muestra de las posibilidades del framework y de como, a partir de mero código HTML, se pueden configurar diversos dinamismos. Ni hace falta decir que, si no basásemos nuestras aplicaciones en jQuery Mobile, realizar estos comportamientos dinámicos exigiría usar bastante Javascript y lidiar con las particularidades de cada dispositivo y navegador.

Artículo por [Miguel Angel Alvarez](#)

## Crear Botones con jQuery Mobile

*Crear botones con jQuery Mobile es muy sencillo y no requiere mucho esfuerzo, sin embargo los resultados son excitantes.*

Como venimos adelantando a lo largo de los anteriores artículos del Manual de jQuery Mobile, este framework ofrece una amplia gama de widgets o componentes de interfaz gráfica de usuario (GUI). Dichos componentes van más allá de los convencionales ofrecidos en HTML 5 y aunque es verdad que han tenido grandes mejoras con respecto a las versiones anteriores del lenguaje de la web, son todavía sobrepasados en calidad y variedad por los que podemos crear a través de jQuery Mobile.

Es por ello que jQuery Mobile ofrece una alternativa muy divertida de enriquecer visualmente nuestras aplicaciones web para móviles, a través de toda una cantidad de formas y efectos que vienen prediseñados en el framework. Lo mejor es que podemos acceder a todo esto de manera muy fácil, lo que nos permitirá crear con extraordinaria sencillez GUIs visualmente más atractivas.

En este artículo, y en otros que publicaremos en breve, estudiaremos las formas en que se pueden crear, personalizar y trabajar con componentes de interfaz gráfica de usuario. De momento nos iniciaremos con la creación de botones, que son unos de los elementos más usados en cualquier aplicación, ya sea para la web o para escritorio. Veremos que jQuery Mobile nos ofrece una alternativa para crear botones de estética atractiva y fácil de implementar en nuestros proyectos, aplicaciones web o sitios para móviles.

### Un Botón simple de jQuery Mobile

jQuery Mobile ofrece una manera simple de crear nuestros botones, a partir de una serie de propiedades o atributos que vienen predeterminados en el framework. Lo más destacable es que, para lograr nuestro objetivo, solo necesitamos pocas líneas de código y además, todo en HTML.

**Nota:** La posibilidad de configurar elementos de interfaz gráfica a través de HTML es una de las características de jQuery Mobile y es una gran diferencia con respecto a las librerías jQuery UI, donde requeríamos de Javascript para crear y mostrar nuestros componentes de interfaz gráfica de usuario o widgets. Aunque vale la pena decir que dichos scripts no eran muy difíciles de concebir o crear, pues todo lo que tiene que ver con jQuery está diseñado para facilitarnos la vida.

Si recordamos un poco sobre lo que hemos aprendido en desarrolloweb.com sobre de jQuery UI para crear un botón lo, cual no era muy difícil, primero necesitábamos un código HTML como el siguiente:

```
<a href="#" id="boton" > Este es my buttton simple jQUI</a>
```

El cual es un simple enlace, el cual se referencia a través del id botón, pero que jQuery UI con ayuda de algo de javascript lo convertía en un bonito botón, dicho script sería algo como el siguiente:

```
$("#boton").button();
```

Ahora bien, en el caso de jQuery Mobile no se requiere crear scripts para implementar nuestros botones. Eso lo hacemos de forma directa en nuestro documento o código HTML. Se logra con código como el siguiente:

```
<a href="#" id="boton" data-role="button" >Este es my buttton simple con jQM</a>
```

Como podemos ver fue muy fácil crear nuestro botón. Solo tuvimos que asignarle el valor de button al atributo data-role de nuestro enlace, algo que podemos hacer sobre un botón creado a través de la etiqueta input lo cual sería algo como esto:

```
<input type="button" data-role="button" value="Boton input"/>
```

Es decir que fácilmente podemos hacerlo de forma directa sobre nuestros submit para enviar formularios a un servidor, lo que nos ayudaría a salir del clásico y convencional botón HTML, eso lo logramos con la siguiente línea de código:

```
<input type="submit" data-role="button" value="Boton input con type submit"/>
```

## Modificar las propiedades por defecto de los botones de jQuery Mobile

Una de las cosas que pasan con los botones es que, por defecto, ocupan todo el ancho de la pantalla, sin importar la resolución de la misma, pero en ciertos momentos no queremos hacer eso. Existe una manera muy simple de crear botones que se ajusten a la tamaño del texto que llevan dentro y al igual que en la creación del botón por defecto, solo necesitamos modificar algunos atributos en el mismo código HTML. Podemos conseguirlo con el siguiente código:

```
<a href="#" data-role="button" data-inline="true" >Tiene el mismo ancho del texto</a>
```

Como se puede observar solo tuvimos que modificar el atributo data-inline y asignarle el valor a true con el fin que el botón se limite al tamaño del texto.

Otras de las cosas que se pueden hacer con los botones es cambiar sus colores, con el fin de no quedarnos con los que vienen de manera predeterminada. Como todo en jQuery Mobile, se hace de manera muy simple, rápida y con un grado de eficiencia muy alto.

Es decir, que si lo que queremos es tener un boton con diferentes colores, tendríamos que escribir un atributo más a nuestros botones, dicho código puede ser el siguiente:

```
<a href="#" data-role="button" data-inline="true" data-theme="e" >Este boton es de otro color</a>
```

Lo que hicimos fue agregar el atributo data-theme al cual le damos el valor de e atribuyéndole un tema nuevo a nuestro botón. Dichos temas van de la a hasta la e. Como podemos hacernos a la idea, cada tema de tendrá unos colores diferentes. Por ejemplo, si probamos el siguiente código:

```
<input type="submit" data-role="button" value="submit con otro color" data-theme="b"/>
```

Obtenemos un botón con un muy atractivo color azul, el cual equivale al darle el valor de b al atributo data-theme. Todo esto nos comprueba que hacer interfaces graficas de usuario que visualmente sean atractivas, es una tarea que se puede hacer en jQuery Mobile con cierto grado de facilidad.

Para mayor claridad le dejo un código de una página web la cual contiene todo lo explicado en este artículo, con el cual espero que aclaren algunas dudas que hayan quedado en el aire:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">
<title>Botones con jQuery Mobile</title>
<link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
<script src="jquery-1.7.1.min.js"></script>
<script src="jquery.mobile-1.0.min.js"></script>
</head>
<body>
<div data-role="page">
<div data-role="header"><h1>Botones</h1></div>
<div data-role="content">
<a href="#" id="boton" data-role="button" >Este es my buttton</a>
<input type="button" data-role="button" value="Boton input"/>
<input type="submit" data-role="button" value="Boton input con type submit"/>
<a href="#" data-role="button" data-inline="true" >Tiene el mismo ancho del texto</a>
```

```
<a href="#" data-role="button" data-inline="true" data-theme="e" >Este boton es de otro color</a>
<input type="submit" data-role="button" value="submit con otro color" data-theme="b"/>
</div>
<div data-role="footer"><h3>Botones</h3>
</div>
</body>
</html>
```

También podemos ver este código funcionando en un [ejemplo ya construido](#).

Espero que se diviertan con jQuery Mobile!! De momento lo dejamos hasta una próxima ocasión, donde hablaremos de cómo hacer listas más atractivas con este framework. No olviden dejar comentarios y apuntes sobre este artículo.

Artículo por [Dairo Galeano](#)

## Con jQuery Mobile las listas son mejores

*Creación de listas y los usos que le podemos dar en jQuery Mobile dentro de nuestras aplicaciones web móviles.*

A lo largo de diversos artículos del [Manual de jQuery Mobile](#) venimos expresando las grandes ventajas que tiene usar este framework para crear aplicaciones, orientadas a dispositivos móviles, con una poderosa interfaz grafica de usuario (GUI). En el artículo anterior pudimos ver un ejemplo de ello: [Crear botones con jQuery Mobile](#). La forma de usar los widgets que vienen predeterminados en el framework es una tarea muy sencilla, pero obtenemos resultados impactantes, con los cuales podremos sorprender a nuestros amigos.

Uno de los componentes de interfaz gráfica de usuario más relevante, eficiente y útil en jQuery Mobile son las listas, ya que nos pueden servir como menús de contextos, o cualquier otro uso que le podamos dar en nuestras aplicaciones web móviles. El límite está en nuestra imaginación, pues todo en este framework se amolda de acuerdo con nuestras necesidades, o más bien, las características específicas del dispositivo móvil que lo cargue.

Llegará el momento en el que tendremos que mostrar un catalogo de artículos, o servicios de una empresa, o cualquier cosa similar. Esta es una tarea que podemos cumplir con los componentes o widgets listas, que creamos con jQuery Mobile.

Siguiendo el ejemplo del catálogo de productos, podemos tener cada uno de los artículos organizados en nuestras listas y al hacer clic, estaríamos viajando de forma directa a una página, que nos mostrara toda la información del elemento seleccionado.

### Ejemplo ilustrativo con listas

Para este artículo donde estamos trabajando con listas vamos a crear un ejemplo bastante sencillo pero muy ilustrativo de las cosas que podemos hacer con dichos elementos. Con esto obtendremos un horizonte de ideas e ilustraciones de cosas útiles por hacer con las listas, que nos brinda jQuery Mobile

Nuestro ejemplo tendrá cuatro páginas web, incluidas dentro del mismo documento HTML. En la primera crearemos de inmediato una lista, que nos servirá como ejemplo clásico e ilustrativo de este artículo. En la lista habrán tres elementos con textos diferentes, al hacer clic sobre uno estaremos viendo una pequeña información sobre aquello que se haya seleccionado.

La información de cada uno de los textos presentes en los elementos de la lista se albergará en las tres páginas restantes. Aunque en nuestro ejemplo no serán productos, sino medios de transporte.

Inicialmente comenzaremos creando una lista sencilla con todas las configuraciones predeterminadas por jQuery Mobile, ya sobre la marcha iremos configurándola para conseguir mejores efectos visuales. Para obtener una lista sencilla basta con un código como este:

```
<ul data-role="listview" >
```



```
<li><a href="#coche" >Coche</a></li>
<li><a href="#avion">Avión</a></li>
<li><a href="#aeroplano">Aeroplano</a></li>
</ul>
```

Hasta aquí hemos conseguido crear una lista sencilla, donde sus elementos servirán como enlace para comunicarnos con otras tres páginas web. Cada página esta relacionada de forma directa con los elementos de la lista, es decir, que al dar clic sobre el elementos con texto de "Coche", el navegador nos llevará a una página que contiene información acerca de los coches.

## Enriquecer visualmente las listas

Al igual que en el caso de los botones, podemos hacer ciertas modificaciones e ir agregando atributos a las listas de forma directa sobre las etiquetas HTML. Esta posibilidad de configurar los propios widgets en el código HTML facilitará mucho la vida, a la hora de crear nuestras aplicaciones con jQuery Mobile.

### Divisiones:

Lo primero que vamos a hacer con las listas es crear unos divisores, que son otros elementos de la lista, pero que actúan de forma diferente. Éstos elementos son solo para orientar al usuario en la lista. En otras palabras, ellos actúan como una especie de objetos inhabilitados a los eventos que se puedan generar por el usuario, y por tanto, no reaccionaran al clic al doble o al teclado etc.

En las listas se pueden agregar o indexar divisores en cualquier lugar, algo que es muy frecuente, pues aunque no respondan a ningún evento, ellos cumplen una tarea importante. Se trata de una forma visual de mantener informado al usuario que está usando nuestra aplicación web.

A continuación vamos a crear divisores en nuestra lista, estos nos servirán para indicar los diferentes tipos de medios de transporte, que en nuestro ejemplo son: Transporte terrestre y transporte aéreo, para que nuestro usuario sepa por donde va.

**Nota:** Para servir de ejemplo de uso de los divisores podemos pensar en otra cosa. Imagina que tu lista tuviera productos comerciales. Entonces podrías poner divisores con textos como: artículos electrónicos y artículos de alimentación.

La tarea con los divisores es muy sencilla y se consigue con un código como el siguiente.

```
<ul data-role="listview" >
  <li data-role="list-divider">Medios de transporte terrestres</li>
  <li><a href="#coche" >Coche</a></li>
  <li data-role="list-divider">Medios de transporte Aéreos</li>
  <li><a href="#avion">Avión</a></li>
  <li><a href="#aeroplano">Aeroplano</a></li>
</ul>
```

Como pueden darse cuenta, es muy sencillo. Solo hay que agregar dos elementos `<li>` en la posición donde deseamos tener nuestros divisores, pero a estos elementos debemos asignarle el atributo `data-role` al cual le damos el valor de `list-divider`. Con ese único detalle es suficiente para obtener un divisor en nuestras listas.

**Nota:** En el código anterior los divisores se han indentado a la derecha, pero eso simplemente es para que visualmente se puedan identificar mejor. Sobra decir que en el código HTML estos sangrados no se tienen en cuenta por los navegadores, aunque sí nos sirven a los desarrolladores a leer mejor los códigos que creamos.

### Sangrado:

Otro asunto que atañe a las listas es que en su configuración por defecto no conservan un margen en la página. Si lo deseamos, podemos agregar ese margen por medio de un nuevo atributo a nuestra lista, que se coloca de forma directa en la etiqueta `<ul>` del HTML. Ese atributo es el `data-inset`, el cual recibe un valor booleano, es decir, que si queremos producir el margen solo que darle valor de `true` al atributo. Con esto obtendremos una lista que conservará ciertos márgenes con respecto a la página, además de crearle un borde redondeado. Todo eso se logra con un código como este:

```
<ul data-role="listview" data-inset="true">
```

### Temas gráficos de personalización:

Los componentes de interfaz gráfica de usuario en jQuery Mobile comparten la gran mayoría de atributos, es decir, que a nuestra lista podemos modificar su data-theme con el fin de obtener una lista con color diferente. Esto se consigue igual que se explicó en los botones, con un código HTML como el siguiente:

```
<ul data-role="listview" data-inset="true" data-theme="b">
```

También podemos hacer modificaciones en sus divisores agregando un nuevo atributo en la etiqueta <ul>. Es útil hacer esto para que nuestros divisores se diferencien de los demás elementos, quedándonos un código como este:

```
<ul data-role="listview" data-inset="true" data-theme="b" data-dividertheme="a">
```

Como se puede observar, solo agregamos un nuevo atributo data-dividertheme con el que se le da un tema a los divisores de la lista, este es el "a", con el conseguimos un color negro. El efecto es que los divisores aparecerán con el color diferente que asigna jQuery Mobile por defecto.

## Código completo de uso de listas en jQuery Mobile

Pues bien, espero que hayan comprendido la importancia de las listas. Ahora, para mayor claridad de todo lo explicado vamos a compartir el código completo con todo lo que hemos tratado en este artículo. También usaremos aspectos vistos en la sección de [Crear botones con jQuery Mobile](#), al igual que otros temas que se han tratado previamente en este manual.

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">
<title>Listas con jQuery Mobile</title>
<link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
<scriptsrc="jquery-1.7.1.min.js"></script>
<scriptsrc="jquery.mobile-1.0.min.js"></script>
</head>
<body>
<div data-role="page" id="inicio">
<div data-role="header"><h1>Listas</h1></div>
<div data-role="content">
<ul data-role="listview" >
<li><a href="#coche" >Coche</a></li>
<li><a href="#avion">Avión</a></li>
<li><a href="#aeroplano">Aeroplano</a></li>
</ul>
<ul data-role="listview" data-inset="true" data-theme="b" data-dividertheme="a">
<li data-role="list-divider">Medios de transporte terrestres</li>
<li><a href="#coche" >Coche</a></li>
<li data-role="list-divider">Medios de transporte Aereos</li>
<li><a href="#avion">Avión</a></li>
<li><a href="#aeroplano">Aeroplano</a></li>
</ul>
</div>
<div data-role="footer"><h3>Listas</h3>
</div>
</div>
<div data-role="page" id="coche">
<div data-role="header"><h1>Coches</h1></div>
<div data-role="content">
Los coches son un medio de transporte terrestre<br>
<a href="#inicio" data-role="button" data-theme="b" data-inline="true">Inicio</a>
</div>
<div data-role="footer">
<h3>Los coches</h3>
</div>
</div>
<div data-role="page" id="avion">
<div data-role="header"><h1>Aviones</h1></div>
```

```
<div data-role="content">
  Los aviones son un medio de transporte aereo<br>
  <a href="#inicio" data-role="button" data-theme="e" data-inline="true">Inicio</a>
</div>
<div data-role="footer">
  <h3>Aviones</h3>
</div>
</div>

<div data-role="page" id="aeroplano">
  <div data-role="header"><h1>Aeroplanos</h1></div>
  <div data-role="content">
    Los aeroplanos son un medio de transporte aereo, menor que los aviones<br>
    <a href="#inicio" data-role="button" data-theme="d" data-inline="true">Inicio</a>
  </div>
  <div data-role="footer">
    <h3>Aeroplanos</h3>
  </div>
</div>
</body>
</html>
```

Espero puedan comprender todo el código anterior. Recomiendo leer otros artículos previos del manual si es que tienes alguna duda sobre asuntos como el [sistema multipaginas de jQuery Mobile](#).

Antes de acabar, quiero que se aprecie también que las tres páginas que contienen la información de los medios de transporte (coche, avión, aeroplano), tienen una forma de retorno al inicio. Son enlaces convertidos en botones, si no sabes cómo [crear botones con jQuery Mobile](#) revisa el artículo publicado anteriormente.

También puede [ver el ejemplo en funcionamientos en una página aparte](#).

Con esto les dejo hasta la próxima entrega. Como siempre, es todo un placer poder compartir información con ustedes. Nos vemos pronto y no olvides dejar tus comentarios o dudas.

Artículo por [Dairo Galeano](#)

## Los cuadros de diálogo son diferentes en jQuery Mobile

*Cómo crear cuadros de diálogo con jQuery Mobile. Diversos usos y configuraciones que convierten a los cuadros de diálogo una poderosa herramienta.*

Uno de los componentes de interfaz gráfica de usuario más utilizado por los desarrolladores son, sin lugar a dudas, los cuadros de diálogo. Incluso sin importar el entorno en que se ejecute la aplicación, ya sea web, de escritorio o para dispositivos móviles, e independientemente del objetivo para el que está destinada.

Su importancia se debe a su capacidad de informar al usuario y de servir a la vez para tomar sencillas decisiones que afectan al flujo de la aplicación. Otro detalle importante es que la información de los cuadros de diálogo se trata de manera diferente a cualquier otro tipo de componente de interfaz gráfica de usuario, pues siempre obligan a los usuarios a dar, al menos, una ojeada rápida a la información. Típicos cuadros de diálogo sirven para arrojar pistas de cómo trabajar con las aplicaciones, otros son advertencias y el más común es el de alerta.

**Nota:** Las alertas son muy usadas en los entornos web. Tanto es así que por lo general, cuando uno lee un manual de Javascript, lo primero que se enseña es a poner el tradicional "hola mundo" en un mensaje de alerta de nuestro navegador. Para comprobarlo basta hacer memoria y recordar los primeros artículos que se vieron en DesarrolloWeb.com en los [manuales de Javascript](#) o en el [manual de jQuery](#).

Las alertas, o cualquier cuadro de dialogo que creamos en jQuery Mobile, se construye con una sencillez sorprendente. Además son altamente versátiles, ya que nos permiten hacer muchas cosas, y cuando digo que muchas es porque los diálogos parecen no tener límites, viajando mas allá de una simple alert, confirmDialog, o

inputDialog.

La razón por la que un cuadro de dialogo en jQuery Mobile parece no tener límites es porque realmente son como una página web completa, ya sea de las múltiples en [nuestro documento HTML](#) o separada en un archivo aparte. Gracias al hecho de ser una página independiente, podremos hacer multitud de cosas como agregarle imágenes, crear formularios, anexasle botones, agregar videos, o todo lo que soporte una página web corriente. Con esto se puede entender el sin fin de aplicaciones y posibilidades que los cuadros de diálogo pueden tener en jQuery Mobile. Pero claro, tampoco podemos abusar de ellos, recordemos que aun siguen siendo cuadros de diálogo.

## Convertir una página web en cuadro de dialogo

Para que una página se comporte como un cuadro de dialogo no debemos hacer nada sobre ella, sino que debemos construirla tal cual como se ha venido haciendo en todo el tutorial. Lo que hace que se convierta en un diálogo, es un atributo que se agrega en el enlace que nos lleva a la página. De modo que, la labor de presentar esa página como un diálogo, la realiza el propio framework y con una implementación muy simple. El vínculo también puede ser un botón, si lo preferimos antes que un enlace corriente. En cualquier caso nos llevará hacia la página que se comporta como caja de diálogo, ya esté dentro del mismo documento HTML o fuera de él.

El atributo en concreto que debemos colocar en los enlaces dirigidos a páginas que queremos que se comporten como cuadros de diálogo es `data-rel="dialog"`. Luego veremos ejemplos completos, pero antes quiero aclarar un asunto que fue motivo de duda entre un grupo de personas a las que explicaba el funcionamiento de los diálogos en jQuery Mobile: ¿Qué hace diferente a un cuadro de diálogo de otro tipo de página, si solo es una página normal con un enlace al que hemos colocado un atributo especial?

Sí tiene diferencias y notables, comenzando por que las páginas que son cuadros de diálogos no se guardan en el historial del navegador, es decir, que no podremos regresar a ellas con el botón atrás y por tanto solo tendremos acceso si presionamos el enlace que nos lleva hasta la pagina. Otra diferencia es que el framework se encarga de poner en el header de la pagina un icono de cierre al dialogo. Otro detalle visual atractivo que se produce es que la interfaz gráfica del cuadro de diálogo aparece flotando, permaneciendo el documento anterior con el fondo oscuro.

Pero bien, ahora veamos un completo ejemplo de la creación de un primer cuadro de diálogo, a través de un código cuya sencillez se apreciará. Tendremos una página normal y en ella un botón, el cual abrirá el cuadro de dialogo, que será como una especie de alerta. Dicho botón se consigue con un HTML como el siguiente.

```
<a href="#dialogo" data-role="button" data-inline="true" data-rel="dialog" data-theme="b">Abrir dialogo</a>
```

**Nota:** Vuelvo a insistir en que debemos fijarnos en el atributo `data-rel`, al cual se le asigna el valor de `"dialog"`. Todo lo demás que necesitas saber para entender ese código ya ha sido explicado en artículos anteriores, como [Crear botones con jQuery Mobile](#). Asimismo, también debes haber aprendido el manejo del [sistema multi-paginas](#), que ofrece el framework.

Con un botón como el que creamos en el código anterior, ya estaríamos haciendo de la pagina con id `"dialogo"` un cuadro de diálogo. Sin embargo, también debemos crear un mecanismo de regreso ya que, a diferencia de las cajas de alerta convencionales que se crean con Javascript, éstos no permanecen en la página que los invoca. Debes tener en cuenta que cuando se va a un cuadro de diálogo se viaja a través del enlace a una nueva página, a pesar que jQuery Mobile produzca una visualización diferente.

Para regresar nuevamente a la pagina, podremos agregar el siguiente código en la pagina con id de `"dialogo"`.

```
<a href="#aja" data-role="button" data-rel="back">Aceptar</a>
```

Como se ve, en este caso también es un botón, con la diferencia que esta vez su atributo `data-rel` tiene el valor de `"back"`. Con esto no tenemos la necesidad de asignar la dirección al enlace hacia la página inicial. Incluso si le damos una dirección diferente a la página de inicio, jQuery Mobile nos devolverá a la pagina que invocó el dialogo.

## Asignar temas gráficos a las cajas de diálogo

Otra de las cosas que podemos hacer con los cuadros de diálogo es combinar sus temas, con lo que obtendremos variedad de tipos de alerta. Por ejemplo, podríamos poner el botón de un color, el encabezado de otro y el contenido de otro. Solo tendríamos que asignarle temas diferentes a cada uno de los objetos o elementos mencionados.

Para ver todo esto vamos a crear un segundo cuadro de dialogo, que será llamado a través de otro botón que construiremos con un HTML como este.

```
<a href="#dialogo2" data-role="button" data-inline="true" data-rel="dialog" data-theme="c">Abrir dialogo configurado</a>
```

Ya solo bastaría crear otra página web e identificarla como id="dialogo2", en esta podríamos por ejemplo hacer lo siguiente.

```
<div data-role="page" id="dialogo2">
<div data-role="header" data-theme="b"><h1>dialogo con colores configurados</h1></div>
  <div data-role="content" data-theme="a">
Este es un cuadro de dialogo configurado que esta dentro del mismo HTML del enlace
  <a href="#" data-role="button" data-theme="e" data-rel="back">Aceptar</a>
</div>
```

Como puedes observar solo tenemos que asignarle el atributo data-theme, con diferentes valores, sobre la etiqueta HTML que crea el objeto. De esa forma podemos obtener paginas y en este caso de cuadros de diálogo más variados en color.

## Ejemplo completo de creación de cuadros de diálogo en jQuery Mobile

Pues bien solo nos resta atar cabos sueltos en el tema de los diálogos y para eso les dejo el código del ejemplo que hemos visto en este artículo.

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">
<title>Cuadros de dialogos con jQuery Mobile</title>
<link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
<script src="jquery-1.7.1.min.js"></script>
<script src="jquery.mobile-1.0.min.js"></script>
</head>
<body>
<div data-role="page" id="inicio">
<div data-role="header"><h1>Trabajando con dialogos</h1></div>
  <div data-role="content">
Presiona el boton para abrir un cuadro de dialogo
  <a href="#dialogo" data-role="button" data-inline="true" data-rel="dialog" data-theme="b">Abrir dialogo</a><br>
Presiona el boton para abrir un cuadro de dialogo con colores configurados
  <a href="#dialogo2" data-role="button" data-inline="true" data-rel="dialog" data-theme="c">Abrir dialogo configurado</a>
  </div>
<div data-role="footer"><h3>Cuadros de dialogos</h3></div>
</div>

<div data-role="page" id="dialogo">
<div data-role="header" ><h1>dialogo de ejemplo</h1></div>
  <div data-role="content">
Este es un cuadro de dialogo sencillo que esta dentro del mismo HTML del enlace
  <a href="#" data-role="button" data-rel="back">Aceptar</a>
  </div>
</div>

  <div data-role="page" id="dialogo2">
<div data-role="header" data-theme="b"><h1>dialogo con colores configurados</h1></div>
  <div data-role="content" data-theme="a">
Este es un cuadro de dialogo configurado que esta dentro del mismo HTML del enlace
  <a href="#" data-role="button" data-theme="e" data-rel="back">Aceptar</a>
```

```
</div>
</body>
</html>
```

Sera hasta una próxima, esperamos veros pronto y recuerda que puedes [ver el ejemplo en funcionamiento en una página aparte](#).

Artículo por **Dairo Galeano**

## Iconos con jQuery Mobile

*Para facilitar la creación de interfaces gráficas, jQuery Mobile incluye una gran gama de iconos prediseñados, que podemos incluir en las páginas muy fácilmente.*

A lo largo del [Manual de jQuery Mobile](#) hemos visto que las capacidades de este framework son realmente sorprendentes. En artículos anteriores se han dado a conocer algunas de las bondades de estas librerías para el desarrollo de sitios web para dispositivos móviles. Hemos relatado el uso de diversos componentes de interfaz gráfica de usuario, pero también existen otros modos interesantes de enriquecer visualmente nuestros widgets.

En este artículo exploraremos una de las formas que ofrece jQuery Mobile para mejorar el diseño de las aplicaciones, de una manera sencilla, a través de iconos que vienen prediseñados dentro del mismo framework. El uso de estos elementos es algo que nos facilita mucho la vida a la hora de desarrollar sitios web, pues podemos hacer interfaces visualmente más atractivas.

### Agregar iconos a los componentes de interfaz grafica de usuario

Como hemos tratado de demostrar en artículos anteriores, jQuery Mobile ofrece muchas utilidades, y lo hace de una manera muy fácil. Es el caso de los iconos. Para hacer uso de ellos no es necesario de grandes configuraciones o líneas de código Javascript solo un poco mas de HTML.

Pues bien arranquemos con los iconos. Imaginen que tienen, un botón que nos envía a la página inicial de nuestra aplicación web. Este botón se podría hacer más atractivo y visual si ponemos un icono que represente el fin o el uso que tiene. ¿Cuál sería ese icono? ¿Dónde conseguimos ese icono?

Las dos preguntas o dudas las resuelve jQuery Mobile de manera sencilla. Primero el icono sería una casa que representa el home. Por supuesto, este icono viene incorporado dentro del framework. Respondiendo a la segunda pregunta, para hacer que nuestro botón tenga el icono de la casa, solo se necesita el siguiente código:

```
<a href="#" data-role="button" data-icon="home" > Inicio</a>
```

Solo hay que agregar un nuevo atributo data-icon y asignarle el valor de "home". Con esto conseguimos que nuestro botón tenga un pequeño icono de una casa.

Pero eso no es todo, jQuery Mobile dispone de iconos en varios colores y utilizará aquel que mejor se vea con el tema gráfico que estés utilizando en tu aplicación. Supongamos que tu botón tiene el tema "a", con el que conseguimos que elementos como botones sean de color negro. En ese caso el icono será de color blanco, para que resalte. Ahora si tienes el tema "c", que es un color con tono muy bajo, el framework le asignará a nuestro botón un icono más oscuro, siempre acorde al color del tema.

jQuery Mobile cuenta con una larga lista de iconos. Todos se indican a través de distintos valores del atributo data-icon. Algunos ejemplos son los siguientes:

- **Flecha hacia la izquierda:** data-icon="arrow-l"
- **Flecha hacia la derecha:** data-icon="arrow-r"
- **Flecha hacia arriba:** data-icon="arrow-u"
- **Flecha hacia abajo:** data-icon="arrow-d"
- **Eliminar:** data-icon="delete"
- **Mas:** data-icon="plus"
- **Menos:** data-icon="minus"

**Nota:** Esta es solo una lista parcial, por mencionar algunos ejemplos. Si quieres saber la lista completa te invito a verla en la [documentación oficial de jQuery Mobile](#).

## Posición de los iconos

Por defecto jQuery Mobile, mostrara los iconos en la parte izquierda del botón, pero esa configuración también se puede alterar a través de un atributo en el código HTML. Podemos observar el siguiente tag:

```
<a href="#" data-role="button" data-icon="plus" data-iconpos="bottom"> Inicio </a>
```

Como se puede ver, ahora usamos otro icono, con nombre "plus", además en el código HTML agregamos el atributo data-iconpos="bottom", para poner el icono bajo el texto. Este atributo puede recibir los siguientes valores para posicionar el icono: "top", "left", "right" y "bottom". Sobra la explicación de los resultados que obtendremos con cada uno de ellos.

Además de los valores mencionados anteriormente, el atributo data-icon-pos puede recibir un quinto valor, el cual es: "notext", con el que eliminamos el texto de nuestro botón y solo aparecerá el icono que le hayamos asignado.

Ahora crearemos un botón que solo tendrá nuestro icono, es decir, no aparecerá el texto. Lo agregaremos dentro del header, con el fin de obtener un botón parecido al de la documentación de jQuery Mobile, algo que conseguimos con el siguiente código:

```
<a href="#" data-role="button" data-icon="home" data-iconpos="notext" class="ui-btn-right">Inicio</a>
```

Como pueden ver, no solo lo dejamos sin texto, también le asignamos la clase CSS "ui-btn-right" con el fin de que aparezca a la derecha del header. Todo esto viene prediseñado en el framework, solo debemos usarlo, a través del HTML.

## Dónde situar los archivos de los iconos jQuery Mobile

Los iconos son archivos gráficos y por tanto, para que se vean en tu sitio web, tendrán que estar en algún lugar accesible por el navegador. En principio esa localización puede variar, atendiendo a dos posibles usos del framework.

1. Si estas trabajando con jQuery Mobile dentro de tu mismo servidor o de forma local (es decir, entraste a su web <http://jquerymobile.com>, bajaste el zip del framework para incluirlo en tus páginas), verás que dentro de la descarga hay una carpeta llamada "images", la cual debes incluir dentro del directorio de tu servidor donde has colocado el estilo CSS de jQuery Mobile. (El archivo de hojas de estilo de jQuery Mobile se llama algo como jquery.mobile-1.0.min.css).
2. Si estas trabajado con el CDN de jQuery Mobile, es decir, estás incluyendo tanto los estilos como los scripts Javascript a partir de la ruta absoluta del propio dominio del framework, no necesitas incluir las imágenes en ningún lugar. Es decir, los iconos se accederán directamente desde el servidor de jQuery Mobile y por tanto no necesitas copiarlos en tu espacio de hosting.



## Poniendo todo junto

Para facilitar un poco las cosas voy a dejar un código HTML, con una pagina web de ejemplo donde se usa todo lo explicado en este artículo. Debo mencionar que esta web, no hace nada especial solo demuestra el funcionamiento de los iconos y todo lo expuesto en el presente artículo.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8"/>
<meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">
<title>Iconos con jQuery Mobile</title>
<link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
<script src="jquery-1.7.1.min.js"></script>
<script src="jquery.mobile-1.0.min.js"></script>
</head>
<body>
<div data-role="page">
<div data-role="header" data-theme="b">
<a href="#" data-role="button" data-icon="home" data-iconpos="notext" class="ui-btn-right">Inicio</a>
<h1>Uso de iconos con jQuery Mobile</h1>
</div>
<div data-role="content" data-theme="c">

<a href="#" data-role="button" data-icon="home" >Inicio</a>
<a href="#" data-role="button" data-icon="plus" data-iconpos="bottom">Inicio</a>

</div>
<div data-role="footer" data-theme="b">
<h3>Uso de iconos con jQuery Mobile</h3>
</div>
</div>
</body>
</html>
```

El [ejemplo en marcha se puede ver en una página aparte](#).

Para terminar te invito a modificar el código y obtener diferentes posiciones de los iconos, además de probar otras imágenes de las que ofrece el framework.

Artículo por **Dairo Galeano**

## Las barras de navegación y transiciones en jQuery Mobile

*Artículo dedicado a las barras de navegación, que conseguimos con el role navbar, y las transiciones entre páginas en jQuery Mobile.*

Una de las cosas que particularmente más me han gustado de jQuery Mobile son las barras de navegación, por ser extremadamente prácticas. Sirven precisamente para lo que su nombre indica, navegar, pero lo hacen de una manera magistral. Poseen un aspecto muy vistoso, además de ciertas propiedades que podremos ajustar con el fin de hacerlas más potentes.

Para explicar las barras de navegación vamos a realizar algo parecido al ejemplo desarrollado en el [artículo donde hablamos sobre las listas](#), donde teníamos cuatro páginas dentro del mismo documento HTML. La primera es una página inicial y las otras tres sirven para mostrar la información adicional a la página.

En este caso vamos a hacer el ejemplo de una página que hable sobre jQuery, otra página que hable de jQuery UI y una última sobre jQuery Mobile. La página inicial tendrá solo una introducción y la barra de navegación para ir a las otras. Además, para completar el artículo, veremos también algunas cosas interesantes de la configuración de jQuery al respecto de las transiciones.

## Agregando las barras de navegación

Las barras de navegación son algo parecidas a las pestañas, por eso lo ideal es ponerlas en el encabezado (elemento con `data-role="header"`), o en el pie de la página (elemento con `data-role="footer"`). Esto no es estrictamente necesario, pero hace que nuestra aplicación esté mejor organizada. En el caso del ejemplo de este artículo, pondremos la barra de navegación en el pie de la página, algo que se logra con el siguiente código:

```
<div data-role="navbar">
  <ul>
    <li><a href="#jq" >jQuery</a></li>
    <li><a href="#jqui" >jQuery UI</a></li>
    <li><a href="#jqm" >jQuery Mobile</a></li>
  </ul>
</div>
```

Como debemos saber, si hemos seguido la marcha del [Manual de jQuery Mobile](#), a todo lo que tiene que ver con los widgets del framework, debemos agregarle el atributo `data-role`. En este caso recibe el valor de `"navbar"`. Además, como se aprecia en el código anterior, tenemos que crear una estructura en HTML, con un elemento DIV, dentro del cual debe estar contenida una lista (Elemento UL). Pero atención porque esta lista no organizará sus elementos uno debajo del otro, sino uno a continuación del otro, es decir, crea una barra horizontal.

En este momento, tal vez algunos lectores se preocupen por el espacio, en caso de que, al expandirse de forma horizontal, no quepan todos los elementos en las a menudo estrechas pantallas de los dispositivos. El problema lo resuelve jQuery Mobile de forma muy eficiente, ya que cada uno de los elementos tiene el mismo ancho, es decir que si tenemos dos elementos, cada uno tendrá, la mitad del ancho de la barra, y si son tres entonces cada uno será de un ancho igual a un tercio de la barra. Aun hay más, pues si la pantalla es muy pequeña, o son muchos elementos, entonces mostraría la barra en varias líneas, pasando de una fila de elementos a una rejilla o tabla.

## Agregando temas

A lo largo del [Manual de jQuery Mobile](#), hemos visto que para agregar temas a cualquiera de los widgets, presentes en el framework, solo debemos colocar el atributo `data-theme`. Sin embargo, en las barras de navegación no funciona así. Es decir, si en nuestro elemento DIV hacemos esto, no se provocará ningún cambio. En cambio, si queremos que todos los elementos de la barra de navegación se vean diferentes al que jQuery Mobile le asigna por defecto, debemos agregar el `data-theme` en el elemento que contiene a nuestra barra, que en este caso es el pie de la página.

Si hacemos lo que se explicaba anteriormente terminaríamos por construir un pie de página como el siguiente.

```
<div data-role="footer" data-position="fixed" data-theme="b">
  <div data-role="navbar">
    <ul>
      <li><a href="#jq" data-icon="star">jQuery</a></li>
      <li><a href="#jqui" data-icon="plus"> jQuery UI </a></li>
      <li><a href="#jqm" data-icon="grid"> jQuery Mobile </a></li>
    </ul>
  </div> <!-- /navbar -->
</div> <!-- /footer -->
```

Como puedes ver además de agregar el tema `"b"` al pie de la página, con el que obtenemos el color azul, también le puse un icono a cada uno de los elementos que componen la barra de navegación. Si no sabes cómo trabajar con iconos, revisa el artículo anterior del manual de jQuery Mobile donde nos dedicamos a explicar el tema de los iconos y sus configuraciones.

Pero cabe resaltar que a cada uno de los elementos, puedes asignarle un tema diferente. Tal vez quieras que alguno resalte, como por ejemplo el que tiene el texto jQuery Mobile, algo que conseguimos con un código como el siguiente.

```
<li><a href="#jqm" data-icon="grid" data-theme="e" data-transition="flip"> jQuery Mobile </a></li>
```

Ahora hemos agregado un `data-theme` a un solo elemento, pero quizás les llame la atención, el atributo `data-transition`. Con esa configuración logramos que al pasar de una página a otra se modifique la transición normal que hace jQuery Mobile.

## Transiciones entre páginas con jQuery Mobile

jQuery Mobile cuando se pasa de una página a otra, tanto si son archivos diferentes como si son páginas en el mismo documento HTML con la estructura multipágina, se produce una transición automática. Este asunto ya se había comentado anteriormente en otros artículos de DesarrolloWeb.com, pero para los que no lo sepan todavía, se trata de un efecto especial bastante atractivo, que mediante una animación suavizada, nos intercambia la página que estamos visualizando. La transición por defecto se puede cambiar por medio de atributos HTML de una manera muy sencilla.

El atributo data-transition, con el cual modificamos las transiciones entre las páginas, o cuadros de diálogo, puede recibir los siguientes valores:

- Slide: la que se asigna por defecto. data-transition="slide"
- Slide Up: La página se perderá por la parte superior. data-transition="slideup"
- Slide Down: La página se perderá por la parte inferior. data-transition="slidedown"
- Pop: La página aparece de improvisto. data-transition="pop"
- Fade: Hace un efecto de fundido. data-transition="fade"
- Flip: La página hace un bonito efecto 3D. data-transition="flip"

**Nota:** La transición flip no funciona muy bien en navegadores Firefox e Internet Explorer. Si quieres ver el efecto debes probarlo en Google Chrome o Safari, también es diferente el efecto que se genera en los dispositivos Android 2.2, donde hemos probado el efecto y no es el resultado que se obtienen en dispositivos con iOS 4.x. En iOS 5.X funciona perfecto también.

## Ejemplo de navegación en una página web

Para mayor comprensión dejaremos el código del ejemplo mencionado en este artículo. Por ahorrar un poco de espacio solo he colocado el código de las dos primeras páginas. Tendremos una barra de navegación en la página inicial. Debo mencionar que la página no cumple grandes tareas funcionales, pero ilustra el uso de las barras de navegación, además de las transiciones de páginas. <!DOCTYPE html>

```
<html>
<head>
  <meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">
  <title> Barras de Navegacion</title>
  <link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
  <script src="jquery-1.7.1.min.js"></script>
  <script src="jquery.mobile-1.0.min.js"></script>
</head>
<body>
  <div data-role="page" id="inicio">
    <div data-role="header" data-position="fixed">
      <h1>jQuery</h1>
    </div><!-- /header -->
    <div data-role="content">
      <p>Navega a través de esta aplicación de DesarrolloWeb.com para conocer algo sobre
jQuery</p>
    </div><!-- /content -->

    <div data-role="footer" data-position="fixed" data-theme="b">
      <div data-role="navbar">
        <ul>
          <li><a href="#jq" data-icon="star" data-transition="pop">jQuery</a></li>
          <li><a href="#jqui" data-icon="plus" data-transition="slideup"> jQuery UI </a></li>
          <li><a href="#jqm" data-icon="grid" data-theme="e" data-transition="flip"> jQuery
Mobile </a></li>
        </ul>
      </div><!-- /navbar -->
    </div><!-- /footer -->
  </div><!-- /page -->

  <div data-role="page" id="jqm">
    <div data-role="header" data-position="fixed">
      <h1>jQuery Mobile</h1><a href="#inicio" data-role="button" data-icon="home" data-
iconpos="notext" class="ui-btn-right">Inicio</a>
```

```
</div><!-- /haeder -->
<div data-role="content">
  <p>Lectura inicial de la web oficial de jQuery Mobile:</br> <span class="ui-btn-active">
jQuery Mobile: framework para móviles.</span>
  </p>
  
</div><!-- /content -->

  <div data-role="footer" data-position="fixed" data-theme="b">
    <h3>jQuery Mobile</h3>
  </div><!-- /footer -->
</div><!-- /page -->
</body>
</html>
```

Con todo esto hemos aprendido dos tópicos de jQuery Mobile. Por un lado las mencionadas barras de navegación, pero también hemos dado un primer repaso a las transiciones entre páginas.

Nos vemos en una próxima, recuerda dejar tus comentarios y apuntes. De tarea te queda hacer las dos páginas restantes, o bien puedes [verlas ya realizadas en el ejemplo completo](#) en una página aparte.

Artículo por [Dairo Galeano](#)

## Elemento Slider de jQuery Mobile

*Con jQuery Mobile obtenemos elegantes sliders e interruptores, con los que podemos enriquecer visualmente nuestras aplicaciones móviles.*

Hasta ahora en el [Manual de jQuery Mobile](#) hemos aprendido a implementar variados componentes de interfaz de usuario, que se pueden crear a través del atributo data-role. Como se habrá podido comprobar, a partir de diversos atributos con el prefijo "data-" se pueden optimizar elementos HTML sencillos, y mediante jQuery Mobile convertirlos en elegantes widgets. Lo mejor es que todo se realiza de una forma automatizada y sin necesidad de programación.

Como todos sabemos, también HTML posee de diversos componentes de interfaz gráfica de usuario, que por lo general vienen incluidos dentro de los formularios. Además, ahora, con la versión cinco del lenguaje de la web, disponemos de algunos elementos de formulario adicionales, que en general extienden los INPUT de toda la vida. Se trata de una nueva gama de types diferentes, es decir, diversos tipos de campos INPUT, y dentro de éstos, aparecen los sliders.

Los sliders son una interfaz gráfica de usuario que nos permiten por medio de un elemento deslizable, ajustar un valor entre un rango. Ahora bien, quizás muchos de vosotros os preguntaréis ¿Si podemos crear sliders con HTML5, para que necesitamos a jQuery Mobile?

Si bien con HTML 5 se pueden crear sliders, estos cuentan con la forma o apariencia convencional, así que si lo que queremos es romper un poco el esquema debemos recurrir a jQuery Mobile, para la creación de sliders más sofisticados, con nuevas funcionalidades y capacidades de personalización. Por si esto te parece poco, quizás acabe de convencerte esta posibilidad, de expandir los elementos slider, si sabes que realmente no necesitas configurar practicamente nada en jQuery Mobile, ya que la filosofía del framework es realizar más cosas con menos esfuerzo.

### Modos de creación de los sliders con jQuery Mobile

Para la creación de sliders con jQuery Mobile, disponemos de opciones, con una estructura muy sencilla, además de otras opciones de configuración visual.

Arranquemos con la creación más sencilla, en donde solo debemos crear elementos INPUT, ya que jQuery Mobile se encargará de aplicarles el tema gráfico correspondiente, dependiendo del valor que posea su atributo type. Así, para crear un slider solo debemos escribir el siguiente código.

```
<label for="slider">Slider Simple: </label>
```

```
<input type="range" name="slider" id="slider" max="100" min="1" value="25" />
```

Como puedes ver, es solo un elemento INPUT (ya que el label realmente no forma parte del slider, simplemente coloca un texto al lado). Será jQuery Mobile el que se tome el trabajo de agregar el CSS correspondiente para que funcione de manera eficiente.

**Nota:** Para quienes no están relacionados con este nuevo INPUT de HTML5, cabe comentar que las nuevas especificaciones del lenguaje de la web contienen 13 nuevos tipos de campos INPUT.

Cada uno de estos nuevos INPUT son importantes sobre todo, para dispositivos táctiles, ya que dependiendo del tipo de input aparecerá un teclado virtual acorde para rellenar el campo. Podremos encontrar, entre otros, campos específicos para:

- Input type="tel": Formato para números de teléfonos.
- Input type="search": Formato para campos de búsqueda.
- Input type="url": Formato para escribir direcciones web.
- Input type="number": Formato para escribir valores numéricos.
- Input type="email": Formato para direcciones de email.
- Input type="range": Con este trabajamos en el presente artículo.
- Input type="datetime": Útil para fechas y horas.

Para vuestra información, estamos preparando documentación sobre este asunto que pronto aparecerá en DesarrolloWeb.com.

La etiqueta HTML, como has visto, no tiene código Javascript por ninguna parte. Es decir, jQuery Mobile lo implementa todo sin que debas preocuparte de nada. Sin embargo, debes tener en cuenta que el slider se cree a partir de un elemento INPUT, pues, de no hacer esto, el funcionamiento del slider no estaría garantizado.

Las otras opciones que tenemos a disposición de los sliders están relacionadas con los aspectos visuales del widget. Como ya podemos saber, disponemos del atributo data-theme para modificar el color de los componentes. Adicionalmente, los slider cuentan también con data-track-theme.

Así que si lo que queremos es un slider con variedad de colores tanto en el control que se desliza como en la ruta por donde pasa el control, necesitaríamos un código como el siguiente:

```
<label for="slider1">Slider Simple con temas: </label>
<input type="range" name="slider1" id="slider1" max="100" min="1" value="40" data-theme="b" data-track-theme="a"/>
```

Como se puede notar ahora hemos cambiado los atributos antes mencionados, lo cual no requiere mayor explicación si has seguido el [Manual de jQuery Mobile](#) hasta este punto.

## Hacer botones de opciones binarias o interruptores

Existen unos componentes de interfaz gráfica que son parecidos a los slider, pero que solo tienen dos estados. Se denominan interruptores y quizás podamos haberlos visto en funcionamiento en aplicaciones para dispositivos móviles. Con un interruptor tenemos dos opciones: los podemos activar o desactivar. Con ello podemos configurar alguna función en el dispositivo o en la aplicación web que estemos desarrollando.

Normalmente un slider lo crearemos a partir de una etiqueta INPUT, tal como se explicó hace unos instantes. No obstante, cuando se trata de crear interruptores, se utiliza un elemento SELECT, aunque el valor de data-roler continua

siendo "slider". Veamos un código de ejemplo con el cual podemos crear uno de estos interruptores.

```
<select name="slider" id="flip1" data-role="slider">
  <option value="off">Off</option>
  <option value="on">On</option>
</select>
```

Como se puede ver, es un elemento SELECT con el atributo data-role, al cual asignamos el valor "slider". Además, cabe fijarse también en los dos elementos OPTION, que expresan los distintos estados del interruptor. Para indicar a jQuery Mobile cuando está encendido y cuando está apagado debemos configurar los value de los OPTION. Puedes ver en el código de ejemplo los valores en el atributo value, "off" en el primero y "on" en el segundo.

**Nota:** Si haces más OPTION de los debidos, jQuery Mobile creará el slider de todas formas, pero no será nada agradable el resultado, obtendrás las tres opciones, pero enredadas. De igual forma, si no le das por ejemplo el valor al atributo value de la opción de encendido, el interruptor no funcionará.

A la hora de crear un interruptor también lo puedes personalizar, con los temas que disponemos en el framework. Para esta tarea disponemos del atributo data-theme, dejando a jQuery Mobile framework la tarea de hacer la combinación de temas entre las dos opciones.

## Conclusión

Con el fin de obtener componentes más vistosos y uniformes, será el propio framework quien se encargue que la configuración visual de cada uno de los elementos INPUT que se implementen en modo de slider.

Por tanto, nuestra tarea de codificación se queda en más bien poco. No obstante, como de costumbre, dejamos el código de un ejemplo completo, con todo lo explicado en este artículo.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">
<title>Sliders con jQuery Mobile</title>
<link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
<script type="text/javascript" src="jquery-1.7.1.min.js"></script>
</head>
<body>
<div data-role="page">
  <div data-role="header">
    <h1>Sliders</h1>
  </div>
  <div data-role="content">
    <div data-role="content">
      <form action="#" method="get">
        <div data-role="fieldcontain">
          <label for="slider">Slider Simple: </label>
          <input type="range" name="slider" id="slider" max="100" min="1" value="25" />
          <br>
          <label for="slider1">Slider Simple con temas: </label>
          <input type="range" name="slider1" id="slider1" max="100" min="1" value="40" data-theme="b" data-track-theme="a"/>
          <br>
        </div>
      </form>
    </div>
    <div data-role="fieldcontain">
      <label for="flip1">Seleccionar una de dos opciones</label>
      <select name="slider" id="flip1" data-role="slider">
        <option value="off">Off</option>
        <option value="on">On</option>
      </select>
    </div>
  </div>
  <div data-role="footer">
    <h3>Sliders</h3>
  </div>
</div>
```

```
</div>
</div>
</body>
</html>
```

Recuerda que también puedes [ver el ejemplo en una página aparte](#), para comprobar su funcionamiento. Hasta una próxima entrega.

Artículo por **Dairo Galeano**

## Elementos Collapsible de jQuery Mobile

*Uno de los widgets extraordinariamente atractivos que podemos encontrar en jQuery Mobile son los collapsible. Te explicamos cómo funcionan.*

Los elementos Collapsible de jQuery Mobile, son widgets bastante útiles, aunque algunos no sepan cómo usarlos. Es por ello que voy a comentar por brindar una breve explicación sobre este componente de interfaz gráfica de usuario.

Un collapsible, es un elemento que posee una barra con un título, que va a responder a eventos del usuario. Así, cuando se haga clic o tab -en caso de los eventos touch-, este abrirá o cerrará un panel con cualquier contenido que se haya puesto allí.

En el caso del Collapsible que se genera con jQuery Mobile, además de crear la barra, pone el icono de mas "+" cuando el panel está cerrado (es decir, data-icon="plus"), y el menos (data-icon="minus") cuando el contenido esta abierto. Todo esto lo hace jQuery Mobile de forma automática, limpia y rápida.

Debes tener en cuenta que la palabra collapsible en español significa desplegable, pero para no llevarnos a engaño con los clásicos elementos de formulario desplegables, en este artículo usaremos el término en inglés.

### Creación simple de elementos Collapsible

Los elementos Collapsible, requieren cierta configuración, o más bien ciertos requisitos de elementos HTML. Cuando se crea el Collapsible, con un esquema de etiquetas HTML bien construido, nos aseguramos que jQuery Mobile haga su tarea de forma eficiente.

Para comenzar, necesitamos un elemento DIV, que debe tener, como muchos otros widgets vistos en el manual de jQuery Mobile, el atributo data-role con el valor "collapsible".

**Nota:** Además existen otros requerimientos, que explicaremos después de ver el siguiente código

A continuación podemos ver cómo se crea un elemento collapsible.

```
<div data-role="collapsible">
  <h3>Elemento Collapsible simple</h3>
  <p>Este es el contenido del collapsible el cual podemos ocultar</p>
</div>
```

Como puedes observar en el código anterior, dentro del elemento DIV debemos hacer un encabezado, que será el título del collapsible. A continuación hacemos un párrafo dentro del DIV, el cual es el contenido del elemento, donde podemos insertar cualquier etiqueta HTML como contenido.

Por defecto, el collapsible aparece recogido, pero si se requiere podemos configurar el widget para que aparezca abierto al cargar la página. Para ello solo hay que agregar el atributo data-collapsed, que recibe un valor booleano. La forma de usarlo es muy simple y se puede apreciar en el siguiente código.

```
<div data-role="collapsible" data-collapsed="false">
  <h3>Elemento Collapsible abierto</h3>
  <p>Este es el contenido del collapsible el cual podemos ocultar</p>
```



```
<a href="#" data-role="button" data-inline="true">hola</a>
</div>
```

Con esto estamos creando un collapsible que al cargar la página está abierto. Además quise agregar un botón, algo que ya hemos aprendido en artículos pasados del manual de jQuery Mobile, para ilustrar que podemos meter cualquier elemento dentro de este widget. Por rizar el rizo, podríamos insertar hasta otro collapsible dentro del primero.

## Temas gráficos en collapsible

A lo largo del [manual de jQuery Mobile](#), se ha podido apreciar el uso de los temas prediseñados en el framework, que permiten aportar mayor variedad estética a los componentes de interfaz gráfica. En los temas podemos colocar diversos colores y efectos. Revisa el manual para encontrar más información sobre los temas, porque de momento en este artículo solo veremos lo relevante al widget.

Adicionalmente al atributo data-theme, los collapsible, tienen otro atributo para trabajar su configuración visual. Se trata de data-content-theme, con el que podemos asignar un tema diferente al contenido del elemento. Es decir, mientras el encabezado o la barra donde está el título, tendrá asignado el tema que modificamos con data-theme, podremos asignar por separado un tema al cuerpo con data-content-theme. Este interesante detalle lo podemos ver con el siguiente código.

```
<div data-role="collapsible" data-theme="a" data-content-theme="e">
  <h3>Encabezado con tema A</h3>
  <p>ENota:1 contenido tiene el tema e</p>
  <div data-role="collapsible" data-content-theme="a">
    <h3>Collapsible dentro de otro</h3>
    <p>En este caso vemos que este esta dentro de otro</p>
  </div>
</div>
```

Como se puede apreciar, también está inmerso un collapsible, cada uno puede tener sus temas diferentes, aunque también se puede trabajar con el tema que tiene el collapsible padre.

## Crear un acordeón con collapsible

Otro de los atractivos que tiene el elemento collapsible, es el hecho de poder convertirlo en un acordeón. Se trata de un componente donde podemos encontrar varios paneles unidos y relacionados entre sí, de modo que al abrirse uno, todos los demás se cerrarán. En un acordeón se mantendrá visible solo uno de los elementos que lo compone.

Este widget, o componente de interfaz gráfica de usuario, se encuentra disponible en jQueryUI, pero cabe aclarar que la forma de creación de jQuery Mobile es diferente. En este caso se simplifica todo, además de auto-inicializar, cada uno de los widgets que se han creado con data-role.

Siguiendo el camino de jQuery Mobile framework, es importante decir que para la creación de acordeones disponemos de un valor ?collapsible-set? que se le puede dar a un DIV con el atributo data-role. Esto se ve reflejado en el siguiente código.

```
<div data-role="collapsible-set" >
</div>
```

Dentro de este elemento DIV agregamos todos los collapsibles que deseemos, haciendo que cada uno se comporte como un elemento del acordeón. Como puedes imaginar, a los acordeones también se les puede aplicar temas gráficos con data-theme o data-content-theme, para que cada uno de sus elementos tenga el mismo color. Aunque también puedes asignar un tema diferente a cada uno de los elementos.

Para mayor claridad de todo lo explicado en este artículo veamos el código completo de una pequeña aplicación web móvil que ilustra el uso de collapsible.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">
    <title>Collapsible con jQuery Mobile</title>
```

```
<link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
<script src="jquery-1.7.1.min.js"></script>
<script src="jquery.mobile-1.0.min.js"></script>
</head>
<body>
  <div data-role="page">
    <div data-role="header" data-theme="b">
      <h1>Elementos Collapsible</h1>
    </div>
    <div data-role="content">
      <div data-role="collapsible">
        <h3>Elemento Collapsible simple</h3>
        <p>Este es el contenido del collapsible el cual podemos ocultar</p>
      </div>

      <div data-role="collapsible" data-collapsed="false">
        <h3>Elemento Collapsible abierto</h3>
        <p>Este es el contenido del collapsible el cual podemos ocultar</p>
        <a href="#" data-role="button" data-inline="true">hola</a>
      </div>

      <div data-role="collapsible" data-theme="a" data-content-theme="e">
        <h3>Encabezado con tema A</h3>
        <p>El contenido tiene el tema e</p>
        <div data-role="collapsible" data-content-theme="a">
          <h3>Collapsible dentro de otro</h3>
          <p>En este caso vemos que este esta dentro de otro</p>
        </div>
      </div>

      <div data-role="collapsible-set" data-theme="e" data-content-theme="a">
        <div data-role="collapsible">
          <h3>Primer elemento del acordeon</h3>
          <p>Ahora estamos viendo el lemento tres del acordeon</p>
        </div>

        <div data-role="collapsible" >
          <h3>Segundo elemento del acordeon</h3>
          <p>Ahora estamos viendo el lemento tres del acordeon</p>
        </div>

        <div data-role="collapsible" data-theme="a" data-content-theme="b">
          <h3>Tercer elemento del acordeon con tema diferente</h3>
          <p>Ahora estamos viendo el lemento tres del acordeon tambien tiene contenido
con otro tema</p>
        </div>
      </div>
    </div>
    <div data-role="footer">
      <h3> Elementos Collapsible</h3>
    </div>
  </div>
</body>
</html>
```

Recuerda que también puedes [ver un ejemplo en funcionamiento en una página aparte.](#)

Artículo por **Dairo Galeano**

## Hacer columnas y grillas con jQuery Mobile

*Creación de documentos web con grillas o contenidos dispuestos en columnas mediante jQuery Mobile.*

Todas las aplicaciones de jQuery Mobile están diseñadas para trabajar en resoluciones de pantallas variadas. Por esta razón, cuando creamos una lista, esta ocupa toda la pantalla, a menos que nosotros la cambiemos. Sin embargo, en algunos momentos, tal vez deseemos dividir los contenidos en columnas.

Como bien sabemos, con CSS podemos implementar esa división en columnas, pero como la labor de jQuery Mobile es ahorrarnos trabajo, las diseñaremos de una manera más sencilla gracias a él. Así que, si lo que tenemos pensado es hacer una aplicación web para dispositivos móviles en donde su contenido esté dividido en columnas, jQuery Mobile te ayudará de manera muy eficiente.

En este artículo del [Manual de jQuery Mobile](#) exploraremos la creación de columnas, mediante la grilla que nos ofrece el *framework* para maquetar contenidos.

### Creación de columnas en una aplicación jQuery Mobile

Imagina que un día tienes la necesidad de hacer unas columnas en tu aplicación web móvil, como en los periódicos. Si deseas recurrir a la ayuda que te brinda jQuery Mobile, podrás dividir esos contenidos en dos, tres y hasta ¡cinco columnas! Obviamente, en algunos tipos de aplicaciones y dispositivos, la usabilidad no saldrá beneficiada al utilizar hasta 5 columnas, pero al menos lo permite el framework.

Otro aspecto interesante es que no tienes que dividir en columnas toda la página web, sino simplemente algunas secciones de ella. Pero dejémonos de hablar y veamos ya mismo el código para crear dos columnas en nuestra página web móvil.

```
<div class="ui-grid-a">
<div class="ui-block-a"><strong>Esta es la columna numero 1</strong> El texto que se escriba aquí se amoldará a la mitad de pagina.</div>
<div class="ui-block-b"><strong>Esta es la columna numero 2</strong> El texto que se escriba aquí se amoldará a la otra mitad de pagina.</div>
</div><!-- /grid-a -->
```

Lo primero que hacemos es crear un elemento DIV, que tendrá como clase CSS, "ui-grid-a". Con tal clase le indicamos al framework que justo allí inicia nuestra grilla, que en este caso es de dos columnas. Como dije anteriormente no solo pueden tener dos columnas, pues podremos crear grilla que tengan hasta cinco. Para especificar las columnas que deseamos implementar están las siguientes clases.

- Para grillas con dos columnas le asignamos la clase: ui-grid-a.
- Para grillas con tres columnas le asignamos la clase: ui-grid-b.
- Para grillas con cuatro columnas le asignamos la clase: ui-grid-c.
- Para grillas con cinco columnas le asignamos la clase: ui-grid-d.

Una vez establecida la grilla, debemos proceder con la creación de los bloques, uno por columna. En el código anterior son dos bloques, cuya creación se logra también con la asignación de clases CSS. Como se habrá podido observar, asignamos al primer elemento DIV la clase "ui-block-a", mientras el segundo tiene "ui-block-b". Al igual que con las grillas, los bloques van aumentando de acuerdo con el número que hayamos establecido en la grilla. Solo debemos cambiar la última letra.

- Para crear el primer bloque usamos la clase ui-bloc-a.
- Para crear el segundo bloque usamos la clase ui-bloc-b.
- Para crear el tercer bloque usamos la clase ui-bloc-c.
- Para crear el cuarto bloque usamos la clase ui-bloc-d.
- Para crear el quinto bloque usamos la clase ui-bloc-e.

Cada uno de los bloques, tiene el mismo ancho, algo parecido a lo que pasa con las barras de navegación explicadas en un artículo anterior del manual de jQuery Mobile. Así que si tenemos cinco columnas, cada uno de los bloques, tendrá el 20% de ancho del documento web o elemento contenedor de la grilla.

## Hacer Grillas con temas

Hasta este punto del manual de jQuery Mobile, ya debes saber que para agregar temas, a cualquier widget del framework, debemos agregar el atributo data-theme. Sin embargo, esto no es así con las grillas, básicamente porque éstas no son *widgets*.

Debemos hacer un código como el siguiente, que se debe escribir incluso antes de crear los bloques.

```
<div class="ui-grid-a">
<div class="ui-bar ui-bar-e">
Con ello logramos que todos los bloques de nuestra grilla tengan colores uniformes. Sin embargo ¿Qué pasa en el momento en que deseemos tener bloques con colores diferentes? pues bien, en este caso debemos hacer lo mismo, un nuevo elemento DIV, con las clases, ui-bar, para indicar que es una barra, y ui-bar-b, la cual asigna color al bloque o la grilla. En este caso el código que obtendremos será el siguiente:
<div class="ui-grid-a">
  <div class="ui-block-a" >
    <div class="ui-bar ui-bar-e">
      <strong>Esta es la columna numero 1</strong>Con color amarillo    </div>
    </div>
    <div class="ui-block-b">
      <div class="ui-bar ui-bar-b">
        <strong>Esta es la columna numero 2</strong> Con color azul      </div>
      </div>
    </div>
  </div><!-- /grid-a -->
```

Ahora obtenemos dos columnas, la primera tiene un color amarillo, asignado por el tema "e", mientras que la segunda tiene es azul, al tener asignado el tema "b".

**Nota:** Las clases para darle color a las columnas y grillas, también se pueden poner en elemento div, que tienen las clases "ui-grid-a" o "ui-block-a", pero la documentación de jQuery Mobile no lo recomienda.

## Crear múltiples Filas

Con lo que hemos hecho hasta ahora solo hemos creado columnas. Para que estas columnas se conviertan en grillas, debemos tener más de una fila, de lo contrario solo tendremos columnas. Para crear el aspecto de una grilla hacemos el siguiente código:

```
<div class="ui-grid-a">
  <div class="ui-block-a" >
    <div class="ui-bar ui-bar-e">
      <strong>Esta es la columna numero 1</strong> El texto que se escriba aqui se almodará a la mitad
de pagina.
    </div>
  </div>
  <div class="ui-block-b">
    <div class="ui-bar ui-bar-a">
      <strong>Esta es la columna numero 2</strong> El texto que se escriba aqui se almodará a la otra
mitad de pagina.
    </div>
  </div>
  <div class="ui-block-a" >
    <div class="ui-bar ui-bar-c">
      <strong>Esta es la columna numero 1</strong> El texto que se escriba aqui se almodará a la mitad
de pagina.
    </div>
  </div>
  <div class="ui-block-b">
    <div class="ui-bar ui-bar-b">
      <strong>Esta es la columna numero 2</strong> El texto que se escriba aqui se almodará a la otra
mitad de pagina.
    </div>
  </div>
</div><!-- /grid-a con dos filas-->
```

Sólo debemos crear dos bloques mas, después de los iniciales, es decir, que si nuestra grilla es de tres columnas, debemos crear seis bloques para obtener dos filas.

Para mayor claridad deo un código de ejemplo con todo lo visto en el presente artículo.

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">
  <title>Columnas con JQM</title>
  <link rel="stylesheet" href="jquery.mobile-1.0.min.css" />
  <script src="jquery-1.7.1.min.js"></script>
  <script src="jquery.mobile-1.0.min.js"></script>
</head>
<body>
<div data-role="page" id="inicio">
<div data-role="header" data-position="fixed">
  <h1>Columnas</h1>
</div><!-- /header -->
<div data-role="content" >
  <span class="ui-btn-active">Dos columnas simples</span>
  <div class="ui-grid-a">
    <div class="ui-block-a" >
      <div class="ui-bar ui-bar-e">
        <strong>Esta es la columna numero 1</strong> El texto que se escriba aqui se almodará a la
mitad de pagina.
      </div>
    </div>
    <div class="ui-block-b">
      <div class="ui-bar ui-bar-b">
        <strong>Esta es la columna numero 2</strong> El texto que se escriba aqui se almodará a la
otra mitad de pagina.
      </div>
    </div>
  </div><!-- /grid-a -->
  <br>
  <span class="ui-btn-active">Dos columnas simples con dos filas</span>

  <div class="ui-grid-a">
    <div class="ui-block-a" >
      <div class="ui-bar ui-bar-e">
        <strong>Esta es la columna numero 1</strong> El texto que se escriba aqui se almodará a la
mitad de pagina.
      </div>
    </div>
    <div class="ui-block-b">
      <div class="ui-bar ui-bar-a">
        <strong>Esta es la columna numero 2</strong> El texto que se escriba aqui se almodará a la
otra mitad de pagina.
      </div>
    </div>
    <div class="ui-block-a" >
      <div class="ui-bar ui-bar-c">
        <strong>Esta es la columna numero 1</strong> El texto que se escriba aqui se almodará a la
mitad de pagina.
      </div>
    </div>
    <div class="ui-block-b">
      <div class="ui-bar ui-bar-b">
        <strong>Esta es la columna numero 2</strong> El texto que se escriba aqui se almodará a la
otra mitad de pagina.
      </div>
    </div>
  </div><!-- /grid-a con dos filas-->
```

```
</div><!-- /content -->

<div data-role="footer" data-position="fixed">
  <h3>Grillas</h3>
</div><!-- /footer -->
</div><!-- /page -->

</body>
</html>
```

También puedes ver el ejemplo en marcha en una página aparte, donde podrás apreciar las tres columnas también, además de las tres filas, código que no dejo completo, para no redundar mucho, así que si lo quieres puedes [ver el código fuente del ejemplo](#).

Artículo por [Dairo Galeano](#)

## Crear temas personalizados para tus aplicaciones jQuery Mobile

*El equipo de jQuery Mobile, pone a disposición una herramienta con la cual, podemos crear temas personalizados para las aplicaciones basadas en el framework.*

Quizás algunos de vosotros ya esteis pensando que todas las aplicaciones jQuery Mobile disponen de muy pocos temas de personalización visual para los *widgets*. Esto lo podría pensar alguien con experiencia en otras herramientas del equipo de jQuery, que recordará la gran variedad de temas que nos brindaba jQuery UI.

Sin embargo, a lo largo del [Manual de jQuery Mobile](#), hemos repetido en numerosas ocasiones que este *framework* para desarrollo de aplicaciones para móviles sólo contaba con cinco temas predefinidos.

La ventaja con jQuery Mobile es que los cinco temas vienen inmersos dentro de un mismo archivo CSS, una misma hoja de estilos que incluimos una única vez al insertar el framework en el HEAD de la página. Sin embargo, se recordará que en jQuery UI cada tema era un archivo CSS diferente.

No obstante, podemos afirmar que cinco *look&feels* en el archivo CSS básico de jQuery Mobile no cubren toda la necesidad que desarrolladores y diseñadores mas exigentes puedan tener. Tal vez, a vosotros no os guste tanto el color azul como a mi, prefirais el verde o el rojo, pero ninguno de estos dos colores están presentes en los temas que vienen prediseñados en el framework.

Para quienes estén pensando que para crear temas personalizados para nuestras aplicaciones jQuery Mobile, necesitamos grandes conocimientos en CSS3, quiero decirles que se equivocan. No se requiere hacer nada de código y todo es mas fácil de lo que parece.

Además, como veremos a continuación, a golpe de ratón obtenemos unos excelentes resultados, a los que nos tiene muy acostumbrados el framework. Es decir, siempre se trata de crear grandes cosas con mucha facilidad, o más bien, con extraordinaria simplicidad.

### Creando el nuevo tema en jQuery Mobile

Como decía anteriormente, no necesitamos grandes conocimientos en CSS3 para crear nuestros propios temas para aplicaciones jQuery Mobile. Para eso, en la misma página oficial del framework, disponemos de una herramienta online, llamada ThemeRoller. Con esta herramienta tenemos a la mano una forma fácil de crear nuestros propios temas.

Se encuentra disponible en la siguiente dirección web: [jquerymobile.com/themeroller](http://jquerymobile.com/themeroller)

Inicialmente ThemeRoller nos da la bienvenida a través de un mensaje de dialogo. Ponen a nuestra disposición una barra lateral, donde podemos ver cada una de las propiedades globales del nuevo tema, además de ver las modificaciones que podemos hacer a temas por separados, dependiendo de su valor en letras.

Para arrancar, tenemos tres temas que van de la A a la C, pero podemos ir agregando más según nuestras necesidades. Así que si quieres crear una aplicación muy variada en colores, puedes hacer temas hasta llegar a letra Z.

En las opciones que tenemos para las configuraciones globales podemos hacer cambios como:

- Cambiar el tipo de la fuente
- Cambiar el radio de los bordes
- Modificar el color de los iconos y muchas otras cosas...

Ya solo es cuestión de qué necesidades tengas o la forma en que quieras que se vean tus aplicaciones. Después, puedes ir modificando uno a uno los temas, dentro de los cuales ya encontramos configuraciones más particulares.

En el centro de la página contamos con un visualizador de cada uno de los temas, donde podemos ver en tiempo real los cambios que estamos creando en nuestro tema actual. Ésto nos ayuda a orientarnos sobre los resultados finales que obtendremos, mostrándonos diferentes widgets que maneja el framework, como listas, botones, barras entre otros.

## Creando nuestro tema personalizado

Como comentaba anteriormente, en el framework no vienen incluidos widgets de color rojo ni verdes, así que vamos a darles esos colores con el fin de obtener una variedad de temas más amplios. Para tal motivo vamos a trabajar partiendo como base de los temas que vienen incluidos dentro de jQuery Mobile.

Si ya ha visitado la aplicación ThemeRoller, te darás cuenta de que inicialmente no nos ofrece los temas incorporados en el framework, así que lo primero que hay que hacer es incorporar dichos temas. Logramos eso dando clic en la opción "import", donde obtendremos un diálogo en el cual podemos escoger la opción "import default theme". De forma automática aparece en el *textarea* todo el código CSS que incorpora jQuery Mobile, por último, al dar clic en "import" del dialogo, visualizaremos cada uno de los temas presentes en el framework.

En mi caso particular, lo que hice fue modificar los temas D y C que vienen en el framework, cada tema de forma individual tiene las siguientes secciones para ser modificadas:

- Header/ Footer Bar
- Content/ Body
- Button: Normal
- Button: Hover
- Button: Pressed

Como veis, las secciones son muy ilustrativas y no se requieren grandes configuraciones, pues todo está servido para hacerlo de manera gráfica, manejando los colores a través de paletas donde podemos seleccionarlos y muchas otras cosas que convierten a ThemeRoller en una herramienta muy útil.

Para solucionar el problema antes planteado de la ausencia de colores como el rojo o el verde, debemos simplemente crear un tema personalizado donde incluimos esos colores, además de agregar un sexto tema.

## Agregando el nuevo tema a mis aplicaciones

Una vez hemos hecho las modificaciones y temas que necesitamos para nuestras nuevas aplicaciones, procedemos a descargar el tema que estamos creando con ThemeRoller, haciendo clic Download Theme. En el diálogo te dice la forma en que debes incluir tu estilo creado, además de darle un nombre.

Puedes copiar el código que aparece en el diálogo de descarga, para incluir tu tema. Claro, que también debes cambiar el nombre que aparece allí (my-custom-theme.css), por el nombre que tú le has asignado.

Al descomprimir el archivo .zip del tema que creamos con ThemeRoller, podemos ver que viene incluido un ejemplo, donde ya se está trabajando con nuestro tema. Podemos observar que parte de la cabecera de ese documento HTML está compuesto de la siguiente forma.



```
<link rel="stylesheet" href="themes/tema-personalizado.min.css" />
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile.structure-1.1.0.min.css" />
<script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
<script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
```

En este caso, estamos usando para el tema modificado el nombre "tema-personalizado", en su versión comprimida, pues en la carpeta de themes dentro del .zip encontramos dos archivos CSS para el mismo tema.

Otro caso particular es que no se usa el archivo "jquery.mobile-1.0.min.css", sino uno con un nombre "jquery.mobile.structure-1.0.min.css". Este archivo es muy importante, pues al no incluirlo no funcionará el ejemplo. Ese archivo también se descarga junto con el .zip que obtenemos al descargar jQuery Mobile, así que puedes también ponerlo a trabajar de forma local, sin necesidad de usar el CDN.

Para terminar nos gustaría decir que un buen diseño con ThemeRoller depende del empeño, por eso te invitamos a trabajar con él, para obtener aplicaciones jQuery Mobile más personalizadas.

Artículo por *Dairo Galeano*