

# JQuery Mobile

# Introducción

- jQuery Mobile es un framework de interfaz de usuario basada en HTML5 diseñado para hacer que los sitios web se puedan acceder desde todo tipo de dispositivos (smartphone, tablet y escritorio).
- La pagina de referencia
  - <http://jquerymobile.com/>
- Se puede acceder a una explicación de los componente de JQuery Mobile en el siguiente enlace.
  - <http://demos.jquerymobile.com/1.4.5/>
- Donde se pueden encontrar demostraciones de los distintos componentes.

# Introducción

- JQuery Mobile, se basa en JQuery, actualmente la versión 1.4.5 de JQuery Mobile, es compatible con la
- La compatibilidad con los navegadores se puede comprobar en
  - <http://jquerymobile.com/browser-support/>

# Introducción

- A modo resumen, la versión 1.4
  - IOS 4
  - Android 2.1 – 4.4
  - Windows phone 7
  - Blackberry 6
  - Firefox Mobile 18
  - Chrome for Android 18
  - Opera Mobile 11
  - Chrome Desktop 16 - 24
  - Safari Desktop 5 - 6
  - Firefox Desktop 10 - 18
  - Internet Explorer 8 - 10
  - Opera Desktop 10 - 12

# Roles

- Parte principal del API.
- Definen que rol juega el elemento en la pagina Web.
- Se aplican con el atributo **data-role**

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

- Pueden ser entre otros
  - page
  - header
  - content
  - footer
  - button

# Paginas

- El atributo “**data-role**” con valor “**page**”, indica que el contenido dentro de la etiqueta con dicho atributo, corresponde a una pagina, por lo que en cada momento solo se muestra el contenido de dicha pagina.

# Paginas

- Para poder visualizar otra pagina, se ha de identificar la etiqueta con atributo **data-role="page"**, con un **id** y crear un vinculo hacia ella.

```
<div data-role="page" id="home">  
    <a href="#acercaDe">acerca de</a>  
</div>  
<div data-role="page" id="acerdaDe">  
    <a href="#home">Inicio</a>  
</div>
```

# Paginas

- En caso de que la pagina sea externa, se procede de forma similar, la diferencia esta en el valor del atributo **href**, que en este caso será la URL.

```
<div data-role="page" id="home">  
    <a href="AcercaDe.html">acerca de</a>  
</div>
```

- Si no existe ningún elemento con data-role=page, el framework, envuelve todo el contenido dentro de un div con data-role=page.
- Aunque esta situación no es recomendable.



# Paginas

- El comportamiento con respecto a las paginas del framework, con motivo de no tener un DOM excesivamente grande, consiste en eliminar aquellas paginas no visibles que hayan sido cargadas vía AJAX, pudiendo ser cargadas de la cache del navegador cuando sea necesario.
- Si se desean mantener en el DOM, se puede activar la propiedad.

```
$.mobile.page.prototype.options.domCache = true;
```

- Para almacenar una pagina en concreto, porque sea muy accedida, se puede indicar con **data-dom-cache="true"**

# Paginas

- Cuando se navega a una pagina externa, se puede modificar la url con la que se carga dicha pagina incluyendo en la etiqueta con **data-role="page"**, la etiqueta **data-url="/<nuevo path de pagina>"**.
- Se puede definir un titulo para la pagina con **data-title="<titulo de la pagina>"**.

```
<div data-role="page" id="pagina2" data-url="/nuevapagina/">
```

```
<div data-role="page" id="pagina2" data-title="pagina2">
```

# Paginas

- Se puede hacer que e la cabecera salga de forma automática un botón de vuelta, añadiendo a la pagina el atributo **data-add-back-btn="true"**.

```
<div data-role="page" id="pagina2" data-add-back-btn="true">
```

- Este botón es añadido a la cabecera de la pagina, siempre que exista y haya una pagina a la que retornar.
- Se puede definir también **data-back-btn-text="previous"** y **data-back-btn-theme="b"**.

# Paginas: AJAX

- No se recomienda realizar una navegación AJAX (la por defecto) a htmls, con multi-paginas, ya que al cargar solo la pagina principal y no todo el html, puede provoca errores de funcionamiento en diálogos o al navegar a las paginas internas.
- Para evitar este efecto, se ha de configurar la navegación sin AJAX, para ello se ha de definir o bien **rel="external"** para sitios externos en otro dominio, o **data-ajax="false"** para paginas internas sin comportamiento AJAX, o **target=""** cuando se carga el contenido en otra pestaña y por tanto no se emplea el mismo DOM.
- Por defecto el framework, siempre accede a los sitios externos sin emplear AJAX.
- Cuando se aplican estos atributos, se pierden las transiciones.

# Diálogos

- La forma recomendada de mostrar un dialogo, es aplicando a la pagina que conforma el dialogo, la propiedad **data-dialog="true"**

```
<div data-role="page" data-dialog="true" id="dialogo">...</div>
```

- Otra opción ya **deprecated**, es aplicar la propiedad **data-rel**, con valor **dialog** al enlace que abre el dialogo.

```
<a href="#dialogo" data-rel="dialog">abrir dialogo</a>
```

- Normalmente el contenido del dialogo, se define como una pagina interna.
- Los diálogos, no se almacenan como pagina, si en el momento de visualizar el dialogo se guarda en favoritos.

# Diálogos

- Se puede definir un botón de cerrar en la **cabecera** del dialogo, añadiendo **data-close-btn="right"**

```
<div data-role="page" data-close-btn="right" data-dialog="true">  
  <div data-role="header">...</div>  
</div>
```

# Diálogos

- Para crear un botón de cerrar personalizado, únicamente habrá que añadir un botón con **data-rel="back"**

```
<div data-role="page" data-close-btn="right" data-dialog="true">
  <div data-role="header">...</div>
  <div data-role="content">
    <a href="#" data-role="button" data-rel="back">
      Volver
    </a>
  </div>
</div>
```

# Diálogos

- Para establecer el texto del botón de cerrar, se puede indicar con la propiedad **data-close-btn-text=""**
- O afectando a todos los botones de todos los diálogos, estableciendo la propiedad global

```
$.mobile.dialog.prototype.options.closeBtnText
```

- Se puede establecer un tema diferente para el fondo sobre el que aparece el dialogo con **data-overlay-theme** aplicado sobre la pagina del dialogo.



# Popup

- Para añadir un popup a la pagina, se ha de anotar un componente con **data-role="popup"** y el link que lo abra con **data-rel="popup"**.

```
<div data-role="popup"></div>
```

- Se pueden abrir de forma programática con el método popup, al que se le pueden pasar dos parámetros
  - "open"
  - "close"
- Cuando se abre, adicionalmente se puede definir una opciones de configuracion
  - **x**: Coordenada X.
  - **y**: Coordenada Y.
  - **transition**: Transición para mostrar el popup.
  - **positionTo**.

# Botones

- Se consiguen aplicando a
  - enlaces, etiquetas `<a>`
  - botones, etiquetas `<input type="button">`
  - botones, etiquetas `<input type="submit">`
- el rol **button**.

```
<a href="AcercaDe.html" data-role="button">acerca de</a>
```

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

```
<input type="submit" data-role="button" value="Acerca De"/>
```

# Botones

- Alternativamente, se puede aplicar la clase CSS **ui-btn**
- Si además se desean aplicar redondeos, se aplica también la clase CSS **ui-corner-all**.

```
<a href="AcercaDe.html" class="ui-btn">acerca de</a>
```

- Si se quiere mostrar un icono se añade **ui-icon-delete**, y si se quiere quitar el texto **ui-btn-icon-notext**

```
<a href="AcercaDe.html" class="ui-btn ui-corner-all">acerca de</a>
```

- Para añadir sombras, se añade **ui-shadow**
- Para ver los botones en la misma línea **ui-btn-inline**

```
<a href="AcercaDe.html" class="ui-btn ui-corner-all ui-icon-delete ui-btn-icon-notext">acerca de</a>
```

# Botones

- Para aplicar un estilo diferente al de la pagina **ui-btn-<tema>**.
- Para ver el formato mini **ui-btn-mini**.
- Para posicionar los iconos **ui-btn-icon-left**.
- Para deshabilitar **ui-state-disabled**.



















































# Botones: Iconos

- Se puede definir un icono en el botón, con el atributo **data-icon**

```
<a href="AcercaDe.html" data-role="button" data-icon="plus">acerca de</a>
```

- Algunos de los iconos definidos
  - delete (x)
  - arrow-l (<)
  - arrow-r (>)
  - arrow-u (^)
  - arrow-d (v)
  - plus (+)
  - minus (-)
  - check (v)
  - gear (\*)

# Botones: Iconos

 action	 alert	 arrow-d
 arrow-d-l	 arrow-d-r	 arrow-l
 arrow-r	 arrow-u	 arrow-u-l
 arrow-u-r	 audio	 back
 bars	 bullets	 calendar
 camera	 carat-d	 carat-l
 carat-r	 carat-u	 check
 clock	 cloud	 comment
 delete	 edit	 eye
 forbidden	 forward	 gear
 grid	 heart	 home
 info	 location	 lock
 mail	 minus	 navigation
 phone	 plus	 power
 recycle	 refresh	 search
 shop	 star	 tag
 user	 video	

# Botones: Iconos

- Se pueden definir nuevos iconos, generando una clase css, que tenga el siguiente formato **ui-icon-<nombre icono>**, que defina una imagen de 18x18 como imagen de fondo

```
.ui-icon-custom-icon {  
    background-image: url("custom-icon.png");  
}
```

# Botones: Iconos

- Por defecto los iconos se sitúan en la parte izquierda.
- Se puede modificar su posición con **data-iconpos**, que puede tomar valores
  - right
  - left
  - top
  - bottom
  - notext

```
<input type="button"  
      data-icon="delete"  
      data-iconpos="notext"  
      value="Icon only"/>
```





# Transiciones

- Se obtiene con el atributo **data-transition**.
- El atributo se aplica sobre el elemento que dispara la navegación, por ejemplo el <a>.

```
<a href="AcercaDe.html" data-transition="slideup">acerca de</a>
```

- Puede tener los siguientes valores.
  - slideup
  - Slidedown
  - slide (movimiento hacia un lado)
  - fade (desaparece)
  - flip (giro de pantalla)
  - pop (aparición desde el centro de la pantalla)

# Transiciones

- Se obtiene con el atributo **data-transition**.
- El atributo se aplica sobre el elemento que dispara la navegación, por ejemplo el <a>.

```
<a href="AcercaDe.html" data-transition="slideup">acerca de</a>
```

- Puede tener los siguientes valores.
  - slideup
  - slide
  - fader
- Para cuando se quiere que los enlaces que vuelven a una pagina anterior, realicen el efecto contrario de transición, se emplea **data-direction="reverse"**

```
<a href="AcercaDe.html" data-direction="reverse">acerca de</a>
```

# Transiciones

- También se puede emplear para volver a la pagina anterior **data-rel="back"**, que hará lo mismo que el botón atrás del navegador, deshacer la navegación, y por defecto aplicara la transición inversa a la aplicada al llegar a la pagina actual.

```
<a href="AcercaDe.html" data-rel="back">acerca de</a>
```

# Cabecera

- Aunque cuando se inserta en una cabecera una etiqueta <a>, se crea por defecto un botón y no es necesario aplicar el rol **data-role="button"** o los estilos CSS, a partir de 1.4 se desaconseja.
- Aunque los botones se van colocando pegados uno a la izquierda y el siguiente a la derecha, repitiendo en otra fila el mismo proceso si hay mas de dos botones, este comportamiento esta desaconsejado, por lo que se han de incluir los estilos.
- Los elementos <h1>, los coloca centrados por defecto.
- Si se quiere anular los estilos, colocación por defecto, se puede emplear **data-role="none"**.

# Cabecera

- Si se quiere situar el botón a la derecha, se puede emplear la clase CSS **class="ui-btn-right"** y para la izquierda **ui-btn-left**
- Se puede aplicar tanto a cabecera como pie de pagina, que se oculten al clickar sobre la pagina con **data-fullscreen="true"**.
- Sino se incluye una titulo en la cabecera, etiquetas H, se ha de incluir un elemento **span**, con el estilo CSS **class="ui-title"**, para que se pinte correctamente los botones dentro de la cabecera.

```
<span class="ui-title" />
```

# Pie de pagina

- Al igual que con la cabecera, no es necesario poner el rol **data-role="button"** a la etiqueta `<a>` para mostrar el estilo de botón.
- Los botones los coloca in-line.
- Los botones, quedan ajustados a la zona dedicada al pie de pagina, se puede aplicar un estilo css, que deja unos márgenes con la zona de Pie de Pagina, este es **ui-bar**.

```
<div data-role="footer" class="ui-bar">  
  <a href="#nuevo" >nuevo</a>  
  <a href="#borrar" >borrar</a>  
  <a href="#editar" >editar</a>  
</div>
```

# Pie de pagina

- Al igual que con la cabecera y el pie, para definir el contenido de una pagina, se tiene **data-role="content"**.

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

- A partir de la versión 1.4, se sustituye por el atributo HTML5 **role="main"**, en conjunto con la clase CSS **class="ui-content"**.

```
<div role="main" class="ui-content"></div>
```



# Otros atributos

- **data-inline**: booleano que permite definir que los elementos se pongan en la misma línea.
- **data-mini**: booleano que permite indicar que los controles de formulario o botones, se creen en un tamaño mas pequeño que el habitual.
- **data-position**: Permite definir el atributo position de CSS a **fixed**, para que los componentes sean visualizados en todo momento, se aplica sobre todo a la **cabecera** y **pie de pagina**.
- **data-id**: Permite mantener un elemento fijo, al que no se le aplique las transiciones. En todas las paginas, debe existir el mismo elemento, y este ha de tener el mismo valor en esta propiedad.

# Otros atributos

- **data-prefetch**: Aplicado en los enlaces o botones, hace que el framework, precargue la pagina a la que hace referencia, aunque no se tenga que mostrar. Es una carga ágil.
- **data-corners**: Permite definir si la caja tiene o no bordes redondeados.
- **data-enhanced**: Permite mejorar el rendimiento al manejar los componentes, ya que se renderizan previamente a su uso.
- **data-rel="back"**: Permite realizar la misma acción que pulsando el botón atrás del navegador, ignorando el atributo href.

# Sombras

- Se puede habilitar la sombra en los elementos con la propiedad **data-shadow**.
- Algunos elementos la tienen habilitada por defecto, como
  - Botones.

```
<input type="button" data-shadow="false" value="Unset shadow">
```

# NavBar

- Se pueden aplicar tanto a Heder como a Footer.
- Se aplica con el rol **navbar**.
- Dentro del elemento con el rol **navbar**, se crea una lista ordenada o no, etiqueta `<ul>` o `<ol>`.

```
<div data-role="header" class="ui-bar">
  <div data-role="navbar">
    <ul>
      <li><a href="#nuevo" >nuevo</a></li>
      <li><a href="#borrar" >borrar</a></li>
      <li><a href="#editar" >editar</a></li>
    </ul>
  </div>
</div>
```

# NavBar

- En una misma línea, se pueden poner máximo 5 botones, de pone mas, se reorganiza en columnas.
- El uso típico, es el de navegación por pestañas, para que la primera pestaña este seleccionada, se puede emplear el estilo CSS **class="ui-btn-active"**.

# Collapsible

- Se pueden definir bloques de contenido, que se oculten o se muestren según el usuario quiera, para ello, habrá que crear la siguiente estructura.

```
<div data-role="collapsible">  
  <h3>Cabecera</h3>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do  
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut  
    enim ad minim veniam, quis nostrud exercitation ullamco laboris  
    nisi ut aliquip ex ea commodo consequat.</p>  
</div>
```

- Por defecto el bloque se muestra oculto, pero, se puede mostrar visible, con la propiedad **data-collapsed="false"**

# Collapsible

- Se toma como título del bloque la primera etiqueta <H>, que se encuentre.
- Se puede aplicar un tema al contenido desde la etiqueta contenedor con la propiedad **data-content-theme**.
- Se pueden definir los iconos empleados con los atributos **data-collapsed-icon** y **data-expanded-icon**.
- Así como la posición de los iconos con **data-iconpos**.

# Collapsible

- El objeto que representa un elemento **collapsibleset**, tiene un método **collapsibleset()**, al que se le puede pasar el parametro "refresh", que permite actualizar el **collapsibleset**

```
<HTML>
<div data-role="collapsible-set" id="set">
  <h3>Cabecera</h3>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
  enim ad minim veniam, quis nostrud exercitation ullamco laboris
  nisi ut aliquip ex ea commodo consequat.</p>
</div>

<JS>
$("#set").collapsibleset("refresh");
```



# Accordion

- Se pueden agrupar elementos ocultables, con el rol **collapsible-set**.

```
<div data-role="collapsible-set">  
  <div data-role="collapsible" data-collapsed="false">  
    <h3>Section 1</h3>  
    <p>I'm the collapsible set content for section 1.</p>  
  </div>  
  <div data-role="collapsible">  
    <h3>Section 2</h3>  
    <p>I'm the collapsible set content for section 2.</p>  
  </div>  
</div>
```

# Grids

- Para poder distribuir de forma uniforme los contenidos en la pantalla, se recurre a unos estilos CSS predefinidos.
- Así se ha de definir un bloque html, típicamente un div, que tenga como class CSS, alguna de las siguientes.
  - **ui-grid-solo**: Una columna.
  - **ui-grid-a**: Dos columnas
  - **ui-grid-b**: Tres columnas
  - **ui-grid-c**: Cuatro columnas.
  - **ui-grid-d**: Cinco columnas

# Grids

- Dentro de esta etiqueta contenedora, se han de definir otros bloques html contenedores, con estilos CSS como
  - **ui-block-a**: Primera celda.
  - **ui-block-b**: Segunda celda.
  - **ui-block-c**: Tercera celda.
  - **ui-block-d**: Cuarta celda.
  - **ui-block-e**: Quinta celda.
- Para definir una nueva fila, se ha de repetir la misma secuencia, es los elementos de la primera columna, son siempre **ui-block-a**.



# Panel

- Son los elementos que permiten crear menús, columnas plegables, ... Es decir aquellos elementos que se salen de la distribución típica de la página.
- Se añaden con **data-role="panel"**.
- Han de ser hijos de **data-role="page"**, pero han de estar definidos antes o después de todos los elementos de **header**, **footer** o **content**.
- Los paneles llevarán asociado un **id**, que será empleado por los enlaces, para mostrarlos, de forma análoga a como se hace con las páginas.
- Se puede definir la propiedad **data-position**, con valores **right** y **left**, que indica por donde aparecerá el panel.

# Panel

- Se puede definir **data-display**, que define la interacción del panel con la pagina, puede tomar valores
  - **reveal**: Da la sensación de que el panel esta bajo la pagina y esta lo muestra mientras se desliza.
  - **push**: Da la sensación de que el panel esta pegado a la pagina, y se muestra en el grado que desaparece la pagina, moviéndose los dos a la vez.
  - **overlay**. Se muestra sobre la pagina tapando lo que haya debajo
- Se puede activar o desactivar la animación con **data-animate="false"**.

# Panel

- Para mostrar el contenido del panel, que se haya añadido de forma dinámica, hay que lanzar.

```
$( "#mypanel" ).trigger( "updatelayout" );
```

- Para abrir un panel

```
$( "#idofpanel" ).panel( "open" , optionsHash );
```

- Para cerrarlo

```
$( "#idofpanel" ).panel( "close" );
```

- Para cambiar el estado

```
$( "#idofpanel" ).panel( "toggle" );
```

# Panel

- El objeto panel, lanza eventos
  - panelbeforeclose
  - panelbeforeopen
  - panelclose
  - panelcreate
  - panelopen



# Formularios

- Se obliga para una correcta renderización, que todos los elementos de formulario vayan acompañados de su correspondiente etiqueta `<label>`.

```
<label for="username" >Username:</label>  
<input type="text" name="username" id="username" value=""/>
```

- Si se desea compactar la representación, se puede emplear el estilo **`class="ui-hidden-accessible"`**, para el **label** y definir el **placeholder** en el campo input.

```
<label for="username" class="ui-hidden-accessible">Username:</label>  
<input type="text" name="username" id="username" value=""  
      placeholder="Username"/>
```

# Formularios

- Para deshabilitar los componentes, con el atributo estándar **disabled**.
- Todos los pares de **label** e **input**, deben ir rodeados de un elemento con el rol **data-role="fieldcontain"**.
- Para agrupar componentes, se emplea la etiqueta **fieldset**, sobre la que se aplica **data-role="controlgroup"**.

```
<fieldset data-role="controlgroup">  
  <input type="checkbox" name="one" id="one" />  
  <label for="one" >One</label>  
  <input type="checkbox" name="two" id="two" />  
  <label for="two" >Two</label>  
</fieldset>
```

# Formularios

- Para los **checkbox** y **radio**, se puede establecer la posición del icono con **data-iconpos**.

```
<fieldset data-role="controlgroup" data-iconpos="right">  
  <input type="checkbox" name="one" id="one" />  
  <label for="one" >One</label>  
  <input type="checkbox" name="two" id="two" />  
  <label for="two" >Two</label>  
</fieldset>
```

- También se puede establece el tema con **data-theme** sobre el **fieldset**.
- La etiqueta **Legend** dentro del **fieldset**, se toma como titulo del formulario.

# Formularios

- Para añadir de forma dinámica componentes a un **data-role="controlgroup"**, se ha ejecutar el evento **create**, después de haber añadido los componentes.

```
<JS>
var count = 0;
$(document).ready(function() {
    $("#cargar").on("click", function(){
        $("#ctrlGroup").append(crearInput(count));
        $("#ctrlGroup").trigger('create');
        count++;
    });
});
<HTML>
<a id="cargar">cargar</a>
<fieldset data-role="controlgroup" id="ctrlGroup"></fieldset>
```

# Listas

- Se proporciona un rol, que permite definir las listas, este rol es **listview**.
- Este rol se aplica sobre una lista ordenada o desordenada.

```
<ul data-role="listview">  
  <li>item 1</li>  
  <li>item 2</li>  
  <li>item 3</li>  
  <li>item 4</li>  
</ul>
```

- Por efecto la lista se expande a toda la pantalla, para enmarcar la lista, se emplea **data-inset="true"**.

# Listas

- Si dentro de un elemento de la lista se sitúa una etiqueta `<a>`, se modifica la representación de la lista incluyendo el icono de flecha derecha, en la parte derecha del item, haciendo que el item sea clickable.
- Se puede cambiar este icono con **data-icon** en el elemento `<li>`

```
<li data-icon="home"></li>
```

# Listas

- Se puede incluir una etiqueta <a> a mayores, definiendo el icono que se quiere que tenga con el atributo **data-split-icon**.

```
<ul data-role="listview" data-split-icon="gear">  
  <li>  
    <a href="">Item 1</a>  
    <a href=""></a>  
  </li>  
</ul>
```

# Listas

- Si el primer elemento que se define dentro de un item, es una imagen, con la etiqueta `<img>`, por defecto está se situara alineada al a izquierda, ocupando de arriba abajo, todo el espacio del item en vertical.



# Listas

- Cualquier otro elemento se sitúa centrado y uno debajo de otro.

```
<ul data-role="listview" data-split-icon="gear">
  <li>
    <a href="">
      
      <h3>Broken Bells</h3>
      <p>Broken Bells</p>
    </a>
    <a href=""></a>
  </li>
</ul>
```

# Listas

- Se puede incluir una división dentro de una lista, para separar los elementos de la lista, para ello se define un elemento de la lista con el rol **data-role="list-divider"**.

```
<ul data-role="listview">  
    <li data-role="list-divider">Divisor</li>  
    <li> item </li>  
</ul>
```

# Listas

- Se puede definir un elemento para que se muestre con una fuente reducida alineado al margen derecho, con el estilo CSS **class="ui-li-aside"**.

```
<ul data-role="listview" >  
  <li>  
    item  
    <p class="ui-li-aside"><strong>6:24</strong>PM</p>  
  </li>  
</ul>
```

# Sublistas

- Dentro de una lista, se puede incluir otra lista, el efecto que muestra JQuery Mobile, será el de mostrar el item de la lista principal, como clickable, aunque no tenga una etiqueta <a>, y al clickar sobre el, abrirá otra pantalla con los elementos de la sublista.

# Sublistas

- Ejemplo de sublista

```
<ul data-role="listview">  
  <li>item 1  
    <ul>  
      <li>sub item 1</li>  
      <li>sub item 2</li>  
      <li>sub item 3</li>  
    </ul>  
  </li>  
  <li>item 2</li>  
</ul>
```

# Sublistas

- Se puede añadir un elemento visual, que muestre los subelementos de un ítem concreto de la lista, creando un elemento html, típicamente un `<span>`, que es un elemento de línea, con la clase CSS **`class="ui-li-count"`**.

```
<ul data-role="listview">
  <li>
    <a href="index.html">
      Inbox
      <span class="ui-li-count">12</span>
    </a>
  </li>
</ul>
```

# Filtro Búsqueda

- Se puede añadir a las listas, un cuadro de filtrado, que se encarga de filtrar los elementos mostrados por la lista, a los que cumplan con el texto introducido en el cuadro de búsqueda, sin mas que añadir el atributo a la lista **data-filter="true"**.
- Se puede hacer que un determinado item de la lista, sea resultado de la búsqueda, aunque en su contenido, no aparezca lo escrito en el cuadro de búsqueda, definiendo las palabras clave en la propiedad del item **data-filtertext="texto sobre el que buscar"**

# Formularios en las listas

- Si en los item de la lista, se van a representar elementos de un formulario, se ha de incluir el atributo **data-role="fieldcontain"** en los item.
- Se pueden aplicar estilos a los elementos que aparecen como item de las listas con **data-divider-theme**, **data-split-theme** y **data-count-theme**.



# Tablas

- Se crea una tabla con el **data-role="table"**, aplicado sobre una etiqueta **table**.
- Se puede hacer que la tabla muestre u oculte columnas, incluyendo a nivel de tabla **data-mode="columntoggle"** y en cada una de las cabeceras de las columnas **data-priority**.

# Tablas

- Ejemplo de tabla

```
<table data-role="table" data-mode="columntoggle" >
  <thead>
    <tr>
      <th data-priority="1">id</th>
      <th data-priority="2">titulo</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>El padrino</td>
    </tr>
  </tbody>
</table>
```

# Tablas

- El botón que se genera con **data-mode="columntoggle"**, aparece con el texto **"Columns..."**, para modificarlo, se aplica **data-column-btn-text**.
- El **data-mode** por defecto es **reflow**.
- También se puede modificar el tema del botón con **data-column-btn-theme**, y el tema del pop-up con **data-column-popup-theme**.
- Se pueden aplicar estilos a la tabla como **table-stripe**, que hace que cada fila se pinte de un color.

# Tablas

- O **ui-responsive** que muestra todas las columnas, y no solo aquellas que tengan un **data-priority="critical"**, aplicándolas el ancho que sigue
  - data-priority="1": columna de 320px (20em)
  - data-priority="2": columna de 480px (30em)
  - data-priority="3": columna de 640px (40em)
  - data-priority="4": columna de 800px (50em)
  - data-priority="5": columna de 960px (60em)
  - data-priority="6": columna de 1,120px (70em)

# Tablas

- Se pueden agrupar cabeceras, añadiendo a la etiqueta **th**, la propiedad **colspan** con el numero de columnas a unir.

# Loader

- Es el componente empleado para mostrar el gif animado de progreso en las peticiones AJAX manejadas por JQuery Mobile.
- Es reutilizable a través del API con **\$.mobile.loading()**, que puede recibir como parámetros

- **show**

```
$.mobile.loading( 'show' );
```

- **hide**

```
$.mobile.loading('hide');
```

# Loader

- El show, puede recibir adicionalmente las siguientes opciones
  - **html**
  - **text**
  - **textVisible**
  - **textonly**
  - **theme**

```
$.mobile.loading( 'show', {  
    text: 'foo',  
    textVisible: true,  
    theme: 'z',  
    html: ""  
});
```

# Toolbar

- Permite crear encabezados y pies de pagina que estén definidos fuera de la pagina.

```
<div data-role="header" data-position="fixed" data-theme="a"></div>  
<div data-role="page" id="home" data-title="Home"></div>  
<div data-role="footer" data-position="fixed" data-theme="a"></div>
```

- Para ello se ha de ejecutar el método **toolbar()** sobre los elementos que representan la cabecera.

```
$( "[data-role='header'], [data-role='footer']" ).toolbar();
```



# Toolbar

- Al método **toolbar()**, se le pueden pasar opciones
  - **addBackBtn**: Crea un botón de volver de forma automática, siempre que haya una pagina ala que volver. Es equivalente a poner en la etiqueta con **data-role="page"**, el atributo **data-add-back-btn="true"**.
  - **backBtnText**: Define el texto del botón de volver.
  - **backBtnTheme**: Define el tema a aplicar al botón de volver.
  - **disablePageZoom**: Cuando los toolbar son fixed, no se permite hace zoom.
  - **disable**: Deshabilita la toolbar.
  - **fullscreen**: configura las **toolbar** para que se oculten al pinchar sobre la pagina y poder ver la pagina a pantalla completa.

# Toolbar

- Al método **toolbar()**, se le pueden pasar opciones
  - **hideDuringFocus**: Define los selectores CSS de los elementos, que cuando reciben el foco hacen que se oculte la barra. Actúa cuando las toolbar son fixed.
  - **position**: Permite definir si la posición es fixed.
  - **supportBlacklist**: Booleano que indica si el browser soporta **position: fixed**.
  - **tapToggle**: Establece si se puede controlar la visibilidad de la toolbar con el evento tap.
  - **tapToggleBlacklist**: Selector que indica los elementos que no harán modificaciones a la visibilidad del toolbar con el evento tap.
  - **theme**: Tema del toolbar.
  - **transition**: Transiciones

# Toolbar

- Al método **toolbar()**, se le pueden pasar opciones
  - **visibleOnPageShow**: Si se muestra la toolbar con la pagina.
- El **toolbar** tiene métodos
  - `hide()`
  - `show()`
  - `toggle()`
  - `refresh()`
  - `enable()`
  - `disable()`
- Y lanza eventos
  - `toolbarcreate`

# Tab

- Se pueden crear pestañas de navegación, añadiendo el **data-role="tabs"**.
- Este componente, toma un **navbar** como los **tabs**, permitiendo la navegación, conservando el estilo.

```
<div role="main" class="ui-content">
  <div data-role="tabs">
    <div data-role="navbar">
      <ul>
        <li><a href="#fragment-1">One</a></li>
        <li><a href="#fragment-2">Two</a></li>
      </ul>
    </div>
    <div id="fragment-1">
      <p>This is the content of the tab 'One', with the id fragment-1.</p>
    </div>
    <div id="fragment-2">
      <p>This is the content of the tab 'Two', with the id fragment-2.</p>
    </div>
  </div>
</div>
```

# PageContainer

- Es el objeto que maneja la colección de paginas. Es único.
- Es el que se encarga de lanzar las transiciones, de añadir al DOM las paginas, ...
- Este objeto ofrece una serie de métodos que permiten trabajar con paginas.
  - **change(url, options)**: Permite cambiar de pagina visualizada dentro de las paginas cargadas.
  - Donde **url** es la url a la que se quiere navegar
  - Y **options** una serie de parámetros que configuran como se hace la navegación.

```
$.mobile.pageContainer.pagecontainer("change", "detalle.html", { data : { 'id' : id }});
```

# PageContainer

- Este objeto ofrece una serie de métodos que permiten trabajar con paginas.
  - **getActivePage()**: Retorna la pagina que actualmente se muestra.
  - **load(url, options)**: Permite cargar una nueva pagina.

# PageContainer

- Los valores posibles de **options**
  - **allowSamePageTransition**: Permite transición a la misma página.
  - **changeHash**: Crea una entrada en la navegación del navegador.
  - **dataUrl**: Url para incluir en el navegador
  - **data**: Datos a mayores que se pueden enviar, pero solo cuando el destino sea una **url**, no es válido para páginas internas.
  - **loadMsgDelay**: Retardo para la aparición del mensaje de cargando, si se carga antes, no se llega a mostrar.
  - **reload**: fuerza la recarga de la página, aunque esta este en el DOM, se emplea cuando hay parámetros en la url.

# PageContainer

- Los valores posibles de **options**
  - **reverse**: si al volver se aplica la transición contraria
  - **role**: data-role del elemento cuando se renderice, sino se establece será el definido en el elemento.
  - **showLoadMsg**: Si se muestra el mensaje de agina cargando
  - **transition**: transicion a emplear al realizar el cambio.
  - **type**: tipo de peticion AJAX (GET, POST)



# Flipswitch

- Permite dibujar un **check** de una forma mas amigable.
- Para ello se ha de añadir **data-role="flipswitch"** al **ckeck**.
- Se pueden configurar los texto que salgan en el componente con los atributos **data-on-text** y **data-off-text**.
- Se puede aplicar también a un **select**.

```
<input type="checkbox" data-role="flipswitch"/>
```

# Temas

- Se aplica con el atributo “data-theme”.
- El API proporciona estilos de la “A” a la “F”.
  - A es el por defecto.
  - B es azul degradado.
  - C es gris claro.
  - D es gris medio.
  - E es amarillo degradado.
  - F es blanco.

# Temas

- JQuery Mobile, proporciona una herramienta para la generación de temas **themeroller**.
  - <http://themeroller.jquerymobile.com/>
- Una vez generado el tema, se añade antes del CSS de JQuery Mobile

```
<link rel="stylesheet" href="jqm/jquery.mobile-1.4.5.css">  
<link rel="stylesheet" href="themes/Personal.min.css">
```

# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$mobile**.
  - **activeBtnClass**: Class CSS para los botones activos. (**DEPRECATED**)
  - **activePageClass**: Class CSS para la pagina activa. (**DEPRECATED**)
  - **ajaxEnabled**: (boolean) Con true, maneja los enlaces y los submit como AJAX. Por defecto es **true**.
  - **allowCrossDomainPages**: (boolean) Define si se permite la carga de paginas de distintos dominios. Es controlado por **\$mobile.loadPage()**. Por defecto es **false**. En Apps **Phoneygap**, se ha de poner a true, junto con **\$support.cors**
  - **buttonMarkup.hoverDelay**: (integer) Permite ajustar en los terminales táctiles, la aplicación de los estilos **hover** y **down**. Reducirlo puede mejorar la sensación, pero puede aparecer el cambio e clases al hacer scroll. Por efecto 200. (**DEPRECATED**)

# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$mobile**.
  - **autoInitializePage**: (boolean) Indica si se procesan las etiquetas de JQuery Mobile al cargar el framework. De tener valor **false**, para que se procesen las etiquetas de JQuery Mobile, se ha de ejecutar **\$mobile.initializePage()** una vez se procesa el árbol DOM (**\$(document).ready()**).

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

- **defaultDialogTransition**: (string) Establece la transición por defecto para los diálogos que usan Ajax. El valor por defecto es **pop**. Con **none** no hay transiciones. (**DEPRECATED**)

# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$.mobile**.
  - **defaultPageTransition**: (string) establece la transición por defecto para las paginas que usa Ajax. Por defecto **fade**. Con **none** no hay transiciones.
  - **gradeA**: Función que retorna un boolean, que indica si el navegador cumple con las condiciones necesarias para emplear JQuery Mobile. Por defecto, solo testea **\$.support.mediaquery**. De no pasar el navegador esta validación, no se aplica el framework.
  - **hashListeningEnabled**: (boolean) Permite habilitar la interpretación de la propiedad **location.hash**, que indica el trozo de la url que tiene la referencia a un ancla (**#ancla**). Por defecto **true**.

# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$.mobile**.
  - **ignoreContentEnabled**: (boolean) Permite habilitar a zonas del documento las modificaciones que hace el framework, en cuanto al comportamiento. Su habilitación, influiré notablemente en el rendimiento, ya que se buscara en cada nodo el atributo **data-enhance**, en aquellos que se encuentre su valor a false, en sus hijos no se aplicaran los comportamientos del framework. Por defecto **false**.
  - **linkBindingEnabled**: (boolean) Habilita a JQuery Mobile a procesar los clicks, sobre los enlaces, Se establecerá a false, para que sea otro framework, el que se encargue. Por defecto **true**.

# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$.mobile**.
  - **degradeInputs**: (Objeto) Permite modificar la tipología de una etiqueta input, en una mas simple. Tiene por propiedades los distintos tipos de campos input, que indican si se degradan, y de hacerlo a que. Por defecto solo se degradan **Range** y **Search**.
    - color = false
    - date = false
    - datetime = false
    - datetime-local = false
    - email = false
    - month = false
    - number = false
    - range = number
    - search = text
    - tel = false
    - time = false
    - url = false
    - week = false



# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$mobile**.
  - **minScrollBack**: (integer) Scroll mínimo que se recordará cuando se retorne a una página. Por defecto es 250. (**DEPRECATED**)
  - **dynamicBaseEnabled**: (boolean) Permite habilitar la modificación dinámica del prefijo de las url. Afectará por tanto a todos los recursos referenciados de forma dinámica. Se ha de emplear solo con aquellos frameworks, que trabajan con la misma etiqueta base para todos los recursos.
  - **focusClass**: (string) Nombre de la clas CSS, empelada para los botones activos. Por defecto **ui-focus** (**DEPRECATED**)
  - **getMaxScrollForTransition**: Función que retorna un entero, y si el valor del scroll es mayor a esta posición, se emplea la transición **none**. Por efecto, su valor es 3 veces el alto de la pantalla.

# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$.mobile**.
  - **keepNative**: (selector) Permite deshabilitar el comportamiento del framework para aquellos elementos seleccionados, no afecta a los hijos. Por defecto toma el valor **[data-role='none']**

```
$.mobile.keepNative = "[data-role='button']";
```

- **maxTransitionWidth**: (integer) Permite establecer un ancho máximo para las transiciones, si se establece un valor en pixel y la ventana es mas ancha que el valor se ejecuta la transición **none**. Por defecto **false**.
- **ns**: (string) Permite redefinir los atributos a interpretar por el framework, incluyendo un texto intercalado. Por defecto es ""

```
$.mobile.ns = 'prueba-';
```

```
<div data-prueba-role="page" id="home">
```

# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$mobile**.
  - **pageLoadErrorMessage**: (string) Establece el texto que aparece al ocurrir un error con Ajax. Por defecto '**Error Loading Page**'.
  - **pageLoadErrorMessageTheme**: (string) Establece el tema a emplear por el dialogo. Por defecto es **e**.
  - **phonegapNavigationEnabled**: (boolean) Reemplaza las llamadas a **window.history.back** por el **helper** de PhoneGap cuando está disponible. Este se ocupa de cuestiones de navegación en la página en aplicaciones de Android PhoneGap utilizando jQuery Mobile.
  - **transitionFallbacks**: (string) Define la transición de repliegue para aquellos navegadores que no soportan transiciones 3D. Por defecto **fade**.

# Configuración

- Se pueden configurar las siguientes características, a través del objeto **\$.mobile**.
  - **orientationChangeEnabled**: (boolean) Permite activar el soporte para cambio de orientación. Por defecto es **true**.

# Configuración

- Se puede extender en objeto **\$.mobile** con parámetros de configuración propios de la aplicación.

```
$.extend( $.mobile ,  
    {  
        propiedad: valor  
    }  
);
```

# Eventos: mobileinit

- JQuery Mobile, define un evento de inicialización **mobileinit**, que se lanza inmediatamente después de cargar la librería de JQuery Mobile.

```
$(document).bind("mobileinit", function(){  
});
```

- Como se lanza inmediatamente después de cargar la librería de JQuery Mobile, se ha de cargar el **Listener** antes

```
<script src="jquery.js"></script>  
<script src="custom-scripting.js"></script>  
<script src="jquery-mobile.js"></script>
```

# Eventos: hashchange

- **hashchange**: Evento que responde al cambio del hash de la url.

```
<JS>
$(document).bind("mobileinit", function(){
    $.mobile.linkBindingEnabled = false;
});
$(window).on("hashchange",function(event) {
    console.log("Cambio el hash: " + location.hash);
});
$(window).trigger('hashchange');
```

```
<HTML>
<a href="#test1" >test 1</a>
<a href="#test2" >test 2</a>
<a href="#test3" >test 3</a>
<a href="#test4" >test 4</a>
```

# Eventos: orientationchange

- **orientationchange**: Evento que responde al cambio de orientación del terminal.
  - Se puede obtener información del estado de la orientación con el atributo **window.orientation**, que tendrá valores 0 (portrait) o 90 (landscape)

```
$(window).on("orientationchange",function(event) {  
    console.log("Cambio de orientacion: " + window.orientation);  
});  
  
$(window).trigger('orientationchange');
```



# Eventos: scrollstart

- **scrollstart**: Se dispara cuando empieza un scroll

```
$(document).on("mobileinit", function(){  
    $(document).on('scrollstart', function(e, data){  
        console.log("Comienza el scroll");  
    });  
});
```

# Eventos: scrollstop

- **scrollstop**: Se dispara cuando el scroll termina.

```
$(document).on("mobileinit", function(){  
    $(document).on('scrollstop', function(e, data){  
        console.log("Termino el scroll");  
    });  
});
```

# Eventos: swipe

- **swipe**: Se dispara cuando se hace **drag** sobre la pantalla, con un desplazamiento vertical de al menos 30px y un horizontal de menos de 75px, durante menos de un segundo.
- Se puede configurar con las propiedades
  - **\$.event.special.swipe.scrollSupressionThreshold**: Mas valor que el por defecto puede anular el scroll. Por efecto 10px.
  - **\$.event.special.swipe.durationThreshold**: Mas tiempo que el por defecto, ya no es un swipe. Por defecto 1000ms.
  - **\$.event.special.swipe.horizontalDistanceThreshold**: Debe tomar valores mayores que el por defecto de 30px.
  - **\$.event.special.swipe.verticalDistanceThreshold**: Debe tomar valores menores que el por defecto de 75px.

# Eventos: swipe

- **swipe:**

```
$(document).ready(function(){  
    $("div.box").on('swipe', function(e){  
        $( event.target ).addClass( "swipe" );  
    });  
});
```

# Eventos: swipe

- **swipe**: existen asociados otros dos eventos, mas concretos **swipeleft** y **swiperight**.

```
$(document).ready(function(){
    $("div.box").on('swipeleft', function(e){
        $( event.target ).addClass( "swipeleft" );
        $( event.target ).removeClass( "swiperight" );
    });
    $("div.box").on('swiperight', function(e){
        $( event.target ).addClass( "swiperight" );
        $( event.target ).removeClass( "swipeleft" );
    });
});
```

# Eventos: tap

- **tap**: Se dispara cuando se hace **click** en una pantalla táctil. Es equivalente al evento **click**.
- Se recomienda su uso en aplicaciones puramente móviles, dado que **click**, presenta un retardo de unos 300ms en Android.

```
$(document).ready(function(){  
    $("div.box").on('tap', function(e){  
        $( event.target ).addClass( "tap" );  
    });  
});
```

# Eventos: taphold

- **taphold**: Se dispara cuando se hace click sostenido en una pantalla táctil.
- Se puede configurar con las propiedades
  - **\$.event.special.tap.tapholdThreshold**: Tiempo que ha de mantenerse la pulsación. Por defecto 750.
  - **\$.event.special.tap.emitTapOnTaphold**: (boolean) Indica si se lanza un **tap** al lanzar el evento de **taphold**. Por defecto es true.

```
$(document).ready(function(){  
    $("div.box").on('taphold', function(e){  
        $( event.target ).addClass( "taphold" );  
    });  
});
```

# Eventos: throttledresize

- **throttledresize**: Se dispara cuando se redimensiona la pantalla incluso en un cambio de orientación.
- Es de uso interno para evitar el lanzamiento abusivo e eventos al redimensionar. Ya que se absorben y se lanzan al pasar un tiempo razonable entre el comienzo de la redimensión y el fin de la misma.

```
$(document).ready(function(){  
    $( window ).on( "throttledresize", function ( event ) {  
        $( "#output-text" ).html( "Event Count: " + ++count );  
    });  
  
    $( window ).trigger( "throttledresize" );  
});
```



# Eventos: Virtualizados

- El API ofrece una serie de eventos virtualizados, que pretenden mediar, entre los eventos en dispositivos móviles, que deberían emplear **tap** y eventos en dispositivos de escritorio, que deberían emplear **onclick**, ....
- Se recomienda su uso en aplicaciones que sean accedidas en ambos entornos, salvo para los casos en los cuales se modifique el contenido, ya que se introduce un retardo de 300ms con respecto a **click**, que puede ocasionar que el target cambie.
- Los eventos son
  - vclick
  - vmousecancel
  - vmousedown
  - vmousemove
  - vmouseout
  - vmouseover
  - vmouseup

# Metodos

- **`$(<selector>).buttonMarkup(options)`**: Permite aplicar los estilos de botón al elemento de forma programática.

```
$("#myAnchor").buttonMarkup({ corners: false });
```

- Las opciones son
  - **`corners`** (boolean) Por defecto true
  - **`icon`** (string) Por defecto ""
  - **`iconpos`** (string) Por defecto "left"
  - **`iconshadow`** (boolean) Por defecto false.
  - **`inline`** (boolean) Por defecto false
  - **`mini`** (boolean) Por defecto false.
  - **`shadow`** (boolean) Por defecto true.
  - **`theme`** (string) Por defecto inherit

# Metodos

- **`$(<selector>).enhanceWithin()`**: Aplica las mejoras de JQuery Mobile a los hijos del componente al que afecta.

```
<JS>
var content = "<div><a
href='http://en.wikipedia.org/wiki/Sydney'>Sydney</a></div>";

$(content).appendTo( "#page-content" ).enhanceWithin();

<HTML>
<div data-role= "content" id="page-content">
```

# Metodos

- **`$(<selector>).jqmEnhanceable()`**: Filtra los componentes que no tengan la propiedad **`data-enhance="false"`**.
- Esto funciona siempre que se haya definido la variable global **`$.mobile.ignoreContentEnabled`** a **`true`**.

```
<JS>
$.mobile.ignoreContentEnabled = true;
$("a").jqmEnhanceable().buttonMarkup({ corners: false });
```

```
<HTML>
<a href="#dialogo" data-enhance="false">boton</a>
<a href="#dialogo">boton</a>
```

# Metodos

- **`$(<selector>).jqmHijackable()`**: Filtra los componentes que no tengan la propiedad **`data-ajax="false"`**.
- Esto funciona siempre que se haya definido la variable global **`$.mobile.ignoreContentEnabled`** a **`true`**.

```
<JS>
$.mobile.ignoreContentEnabled = true;
$("a").jqmHijackable().buttonMarkup({ corners: false });
```

```
<HTML>
<a href="#dialogo" data-ajax="false">boton</a>
<a href="#dialogo">boton</a>
```

# Metodos

- **`$(<selector>).jqmData()`**: Permite almacenar y recuperar datos asociados al elemento.

```
<JS>
$( "#home" ).jqmData( "title" ); //Recupera el String "Home"

<HTML>
<div data-role="page" id="home" data-title="Home">
```

- Se puede emplear en los selectores de Jquery, cumpliéndose que **`$("div[data-role='page']")`**, es equivalente a **`$("div:jqmData(role='page')")`**.

# Metodos

- **`$(<selector>).jqmRemoveData()`**: Permite borrar datos asociados al elemento.

```
<JS>
```

```
$( "#home" ).jqmRemoveData( "title" );
```

```
<HTML>
```

```
<div data-role="page" id="home" data-title="Home">
```

# Metodos

- **`$(<selector>).fieldcontain()`**: Añade un contenedor de estilo a los campos.
- **Deprecated** en favor de aplicar el estilo CSS **`ui-field-contain`**





# Metodos

- **\$.mobile.path.getDocumentBase(boolean)**: Retorna la url que da lugar al DOM.
- Si el parámetro enviado es false (el por defecto), retorna un String, si es true, retorna un objeto parseado con **\$.mobile.path.parseUrl()**.

# Metodos

- **\$.mobile.path.getDocumentUrl(boolean)**: Retorna la url que da lugar al DOM.
- Si el parámetro enviado es false (el por defecto), retorna un String, si es true, retorna un objeto parseado con **\$.mobile.path.parseUrl()**.

# Metodos

- **\$.mobile.path.get(url)**: Retorna la parte del path sin el documento.

```
$.mobile.path.get("http://foo.com/a/index.html");
```

```
//retorna http://foo.com/a/
```

- **\$.mobile.path.getLocation()**: Retorna el **location.href** sin el user/password.
- **\$.mobile.path.isAbsoluteUrl(url)**: Retorna **true** si la url es absoluta.
- **\$.mobile.path.isRelativeUrl(url)**: Retorna **true** si la url es relativa.

# Metodos

- **\$.mobile.path.isSameDomain(url1, url2)**: Comprueba que dos url tienen el mismo dominio.
- **\$.mobile.path.parseLocation()**: Parsea el Location obtenido vía **\$.mobile.path.getLocation()**, con **\$.mobile.path.parseUrl()**

# Metodos

- **\$.mobile.path.parseUrl(url)**: Retorna la url formateada en un objeto con propiedades.
  - authority:
  - directory:
  - domain:
  - doubleSlash:
  - filename:
  - hash:
  - host:
  - hostname:
  - href:
  - hrefNoHash:
  - hrefNoSearch:
  - password:
  - pathname:
  - port:
  - protocol:
  - search:
  - username:

# Metodos

- **\$.mobile.getInheritedTheme(elementos, temaPorDefecto):**  
Retorna el tema heredado por lo elementos, y de no encontrar, el por defecto.

```
$.mobile.getInheritedTheme( $("#contenido"), "x" );
```

- **\$.mobile.silentScroll(y):** Realiza un scroll en la pantalla, sin lanzar los eventos de **scrollstart** y **scrollstop**.

```
$.mobile.silentScroll(100);
```

# Metodos

- **\$.mobile.changePage()**: cambia a una pagina de forma programática.
  - Deprecated en favor de **\$.mobile.pagecontainer("change")**.
- **\$.mobile.loadPage()**: carga una pagina externa, aplicando la mejora dejQuery Mobile al contenido y añadiendo la pagina al DOM de forma programática.
  - Deprecated en favor de **\$.mobile.pagecontainer("load")**.

# Eventos: Ciclo de vida

- El ciclo de vida al cargar una pagina es el siguiente
  - **Pagebeforechange** (Primer evento de antes del cambio de pagina)
  - **Pagebeforeload** (Antes de cargar pagina destino)
  - **Pageload** (Carga de pagina destino)
  - **Pagebeforecreate** (Antes de creación de pagina destino)
  - **Pagecreate** (Creación de pagina destino)
  - **Pageinit** (Iniciar pagina destino)
  - **Pagebeforechange** (Segundo evento de antes de cambio de pagina, cuando pagina destino esta preparada)
  - **Pagebeforehide** (Antes de ocultar pagina origen)
  - **Pagebeforeshow** (Antes de mostrar pagina destino)
  - **Pagehide** (Ocultando pagina origen)
  - **Pageshow** (Mostrando pagina destino)
  - **Pagechange** (Cambio de pagina)



# Eventos: pagebeforechange

- **pagebeforechange**: Se lanza cuando cambia de pagina.
  - El handler, recibe un objeto con el evento, y un objeto **data**, con las propiedades
    - **absUrl**
    - **options**
    - **prevPage**
    - **toPage**
  - Se ejecuta dos veces,
    - La primera, antes de realizar la operación, y entonces toPage contiene un string.
    - La segunda, cuando se ha completado la carga de la pagina de destino, pero todavía no se ha realizado la navegación y toPage tiene un objeto.

```
$(window).on("pagebeforechange",function(event, data) {  
    console.log("Cambio de pagina: " + data.toPage);  
});
```

# Eventos: pagebeforeload

- **pagebeforeload**: Se lanza cuando se va a cargar una pagina en otro fichero con **\$.mobile.changePage()**.
  - El handler, recibe un objeto con el evento, y un objeto **data**, con las propiedades
    - **toPage**
    - **prevPage**
  - **Deprecated** en favor de **pagecontainerbeforeload**

```
$(window).on("pagebeforeload",function(event, data) {  
    console.log("Se va a cargar la pagina: " + data.toPage);  
});
```

# Eventos: pageload

- **pageload**: Se lanza cuando se va a cargar una pagina en otro fichero con **\$.mobile.changePage()**.
  - El handler, recibe un objeto con el evento, y un objeto **data**, con las propiedades
    - **toPage**
    - **prevPage**
    - **textStatus**
  - **Deprecated** en favor de **pagecontainerload**

```
$(window).on("pageload",function(event, data) {  
    console.log("Se cargo la pagina: " + data.toPage + data.textStatus);  
});
```

# Eventos: pageloadfailed

- **pageloadfailed**: Se lanza cuando se va a cargar una pagina en otro fichero con **\$.mobile.changePage()**.
  - El handler, recibe un objeto con el evento, y un objeto **data**, con las propiedades
    - **absUrl**
    - **dataUrl**
    - **deferred**
    - **errorThrown**
    - **prevPage**
    - **textStatus**
    - **toPage**
  - **Deprecated** en favor de **pagecontainerloadfailed**

```
$(window).on("pageloadfailed",function(event, data) {  
    console.log("Fallo la carga de la pagina: " + data.toPage);  
});
```

# Eventos: pagebeforecreate

- **pagebeforecreate**: Se dispara cuando se va a crear una pagina de JQuery Mobile (data-role="page"), no cada vez que se muestra sino en la creación.
  - El evento tiene en la propiedad **target**, el objeto al que afecta.

```
$(window).on("pagebeforecreate",function(event) {  
    console.log("Se va a crear la pagina: " + event.target.id);  
});
```

# Eventos: pagecreate

- **pagecreate**: Se dispara cuando se va a crear una pagina de JQuery Mobile (data-role="page"), no cada vez que se muestra sino en la creación.
  - El evento tiene en la propiedad **target**, el objeto al que afecta.

```
$(window).on("pagecreate",function(event) {  
    console.log("Se a creado la pagina: " + event.target.id);  
});
```

# Eventos: pageinit

- **pageinit**: Se dispara cuando se va a crear una pagina de JQuery Mobile (data-role="page"), no cada vez que se muestra sino en la creación.
  - El evento tiene en la propiedad **target**, el objeto al que afecta.
- **Deprecated** en favor de **pagecreate**

```
$(window).on("pageinit",function(event) {  
    console.log("Se a creado la pagina: " + event.target.id);  
});
```

# Eventos: pagebeforehide

- **pagebeforehide**: Se lanza cuando se va a ocultar una pagina.
  - El evento tiene en la propiedad **target**, el objeto al que afecta.
  - También llega un objeto data, con las siguientes propiedades
    - nextPage
    - prevPage
    - samePage
    - toPage
  - **Deprecated** en favor de **pagecontainerbeforehide**

```
$(window).on("pagebeforehide",function(event) {  
    console.log("Se va a ocultar la pagina: " + event.target.id);  
});
```



# Eventos: pagehide

- **pagehide**: Se lanza cuando se va a ocultar una pagina.
  - El evento tiene en la propiedad **target**, el objeto al que afecta.
  - También llega un objeto data, con las siguientes propiedades
    - nextPage
    - prevPage
    - samePage
    - toPage
  - **Deprecated** en favor de **pagecontainerhide**

```
$(window).on("pagehide",function(event) {  
    console.log("Se oculta la pagina: " + event.target.id);  
});
```

# Eventos: pagebeforeshow

- **pagebeforeshow**: Se lanza cuando se va a mostrar una pagina.
  - El evento tiene en la propiedad **target**, el objeto al que afecta.
  - También llega un objeto data, con las siguientes propiedades
    - prevPage
    - toPage
  - **Deprecated** en favor de **pagecontainerbeforeshow**

```
$(window).on("pagebeforeshow",function(event) {  
    console.log("Se va a mostrar la pagina: " + event.target.id);  
});
```

# Eventos: pageshow

- **pageshow**: Se lanza cuando se va a mostrar una pagina.
  - El evento tiene en la propiedad **target**, el objeto al que afecta.
  - También llega un objeto data, con las siguientes propiedades
    - prevPage
    - toPage
  - **Deprecated** en favor de **pagecontainershow**

```
$(window).on("pageshow",function(event) {  
    console.log("Se muestra la pagina: " + event.target.id);  
});
```

# Eventos: pagechange

- **pagechange**: Se lanza cuando cambia de pagina.
  - El handler, recibe un objeto con el evento, y un objeto **data**, con las propiedades
    - **absUrl**
    - **options**
    - **prevPage**
    - **toPage**
  - **Deprecated** en favor de **pagecontainerchange**.

```
$(window).on("pagechange",function(event, data) {  
    console.log("Cambio de pagina: " + data.toPage);  
});
```

# Eventos: pagechangefailed

- **pagechangefailed**: Se lanza cuando en el cambio de pagina se produce un error, por ejemplo cuando no existe la pagina destino.
  - El handler, recibe un objeto con el evento, y un objeto **data**, con las propiedades
    - **absUrl**
    - **options**
    - **prevPage**
    - **toPage**
  - **Deprecated** en favor de **pagecontainerchangefailed**.

```
$(window).on("pagecontainerchangefailed",function(event, data) {  
    console.log("Cambio de pagina: " + data.toPage);  
});
```

# Eventos: pageremove

- **pageremove**: Se invoca cuando se elimina una pagina externa, con **remove()**.
  - El evento tiene en la propiedad **target**, el objeto al que afecta.
- **Deprecated** en favor de **pagecontainerremove**.

```
$('#[data-role="page"]').on('pageremove',function(e){  
    console&&console.log($(this).attr('id')+" - pageremove!!");  
});
```

# Servidor con Node.js

- Instalar el modulo http-server

```
npm install -g http-server
```

- Arranca el servidor en el directorio de la aplicación.

```
http-server
```

# Emulador

- Google Chrome, tiene una extensión llamada **Ripple**.
- Para instalarla

Configuración -> Extensiones

- Buscar e instalar.

Ripple Emulator

- En la configuración del plugin, habrá que habilitar

Permitir acceso a URL de archivo



# Emulador

- Una vez instalado, aparece un botón junto a la barra de navegación, que permite habilitar/deshabilitar la extensión.
- Una vez habilitado, se mostrará

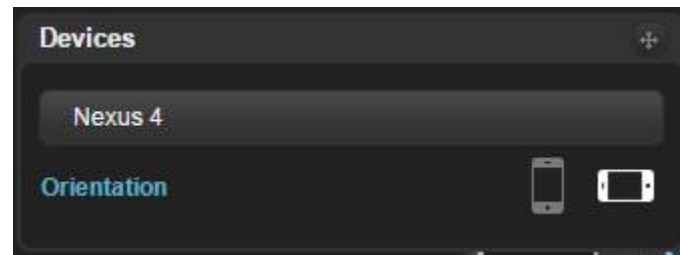


# Emulador

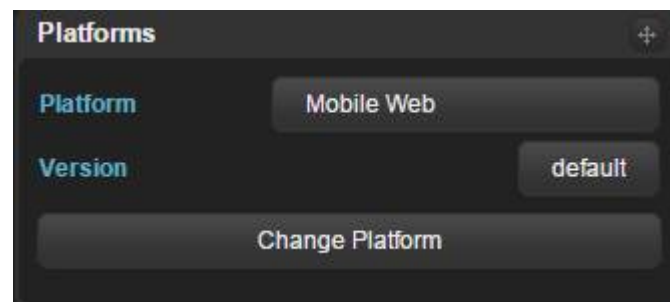
- Se tiene capacidad de configurar a través de pestañas el emulador, teniendo
  - Device
  - Platform
  - Information
  - Accelerometer
  - Settings
  - Geo Location

# Emulador

- Device
  - Permite elegir un terminal a emular.
  - Orientación del terminal

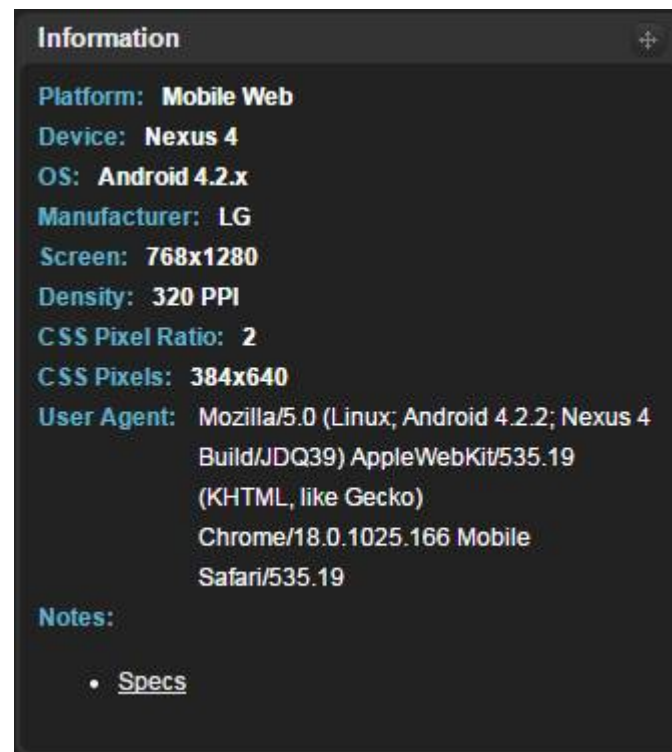


- Platform



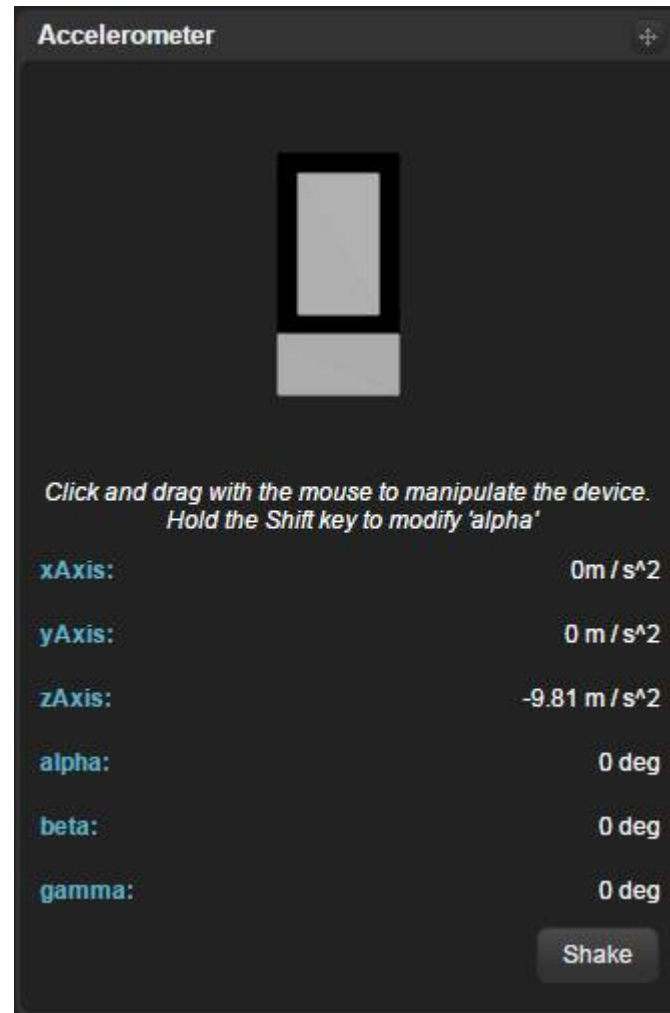
# Emulador

- Information



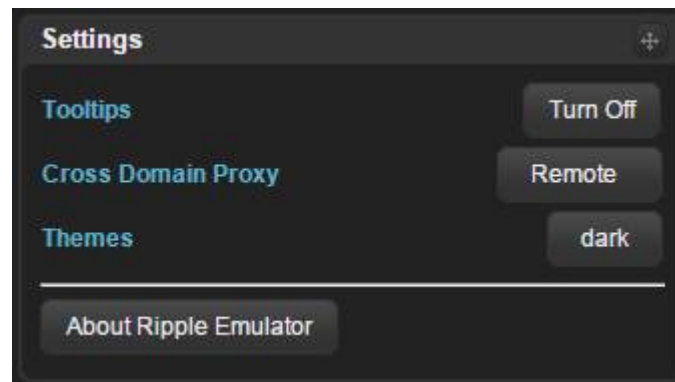
# Emulador

- Accelerometer



# Emulador

- Settings



# Emulador

- Geo Location

Geo Location

Latitude

43,465187

Longitude

-80,522372

Altitude

100

Cell Id

321654

Accuracy

150

Altitude Accuracy

80

GPS Delay (seconds)

0

Simulate GPS Timeout

☐

Navigation Simulator

How do I create a GPX file? [Learn more](#)

Seleccionar archivo

Ningún archivo seleccionado

Go

1x