



# jQuery Mobile

Formularios

# jQuery Mobile

- Se crean de manera similar a como se crea un formulario HTML normal
- Hay unas pocas diferencias entre un formulario estándar y uno JQM
- Cuando se envía el formulario, JQM intercepta la petición y realiza su propia petición al servidor basada en AJAX
- Cuando recibe la respuesta HTML, inserta el HTML en el DOM de la página

# jQuery Mobile

- Para enviar el formulario con el método tradicional, no AJAX, debe añadirse el atributo **data-ajax="false"** en la etiqueta del formulario

```
<form action="miFormulario.php" method="post" data-ajax="false">
```

- Esta opción es necesaria en formularios que contengan campos **<input type="file">**, ya que JQM no puede gestionar este tipo de campo utilizando AJAX



# jQuery Mobile

- o La gestión AJAX se puede desactivar por completo, tanto en enlaces como en formularios usando la opción **ajaxEnabled="false"**, en el gestor del evento **mobileinit**
- o JQM re-estiliza todos los campos del formulario de modo que sean más grandes, más fáciles de tocar y flexibles

# jQuery Mobile

- o Si se quiere evitar que JQM modifique un elemento particular del formulario, se debe añadir el atributo **data-role="none"** al campo

```
<input type="text" name="nombreusuario"  
data-role="none">
```

- o Es posible desactivar en forma global la mejora, utilizando la opción **keepNative** en el gestor del evento **mobileinit**

# jQuery Mobile

- Por ejemplo para desactivar las mejoras para todos los campos de texto y áreas de texto con la clase CSS **noenhance**:

```
$(document).on("mobileinit", function() {  
    $.mobile.page.prototype.options.keepNative = "input[type='text'],  
    textarea.noenhance";  
});
```



# jQuery Mobile

- Una vez que JQM ha mejorado un campo de un formulario, éste puede ser manipulado a través de JavaScript
- JQM respeta el atributo HTML **“disabled”**, hace que el campo mejorado quede inactivo y le da una apariencia gris
- El atributo **“disabled”** no funciona en los links con atributo **dataq-role=“button”**. Para desactivar estos botones debe añadirse la clase CSS **class=“ui-disabled”**

# jQuery Mobile

- Para mejorar la experiencia de usuario, es buena idea encapsular cada pareja etiqueta/campo en una **div** o elemento **fieldset** con el atributo **data-role="fieldcontain"**

```
<div data-role="fieldcontain">  
  <label for="usuario">Usuario</label>  
  <input type="text" name="usuario"  
    id="user">  
</div>
```



# jQuery Mobile

## Etiquetas y accesibilidad

- o Para mejorar la accesibilidad de la aplicación es buena idea asociar un elemento **<label>** con cada campo del formulario
- o Esto es útil tanto para los lectores de pantalla, como para los motores de búsqueda, así como para que JQM aplique los estilos sobre los formularios de manera consistente

# jQuery Mobile

- Para mantener un formulario compacto, se puede utilizar el atributo **placeholder** para colocar el nombre de un campo de texto dentro del propio campo
- JQM para ocultar la etiqueta de un campo facilita dos clases CSS, **ui-hidden-accessible** que se agrega a la etiqueta o la clase **ui-hide-label** que se agrega a un elemento con **data-role="fieldcontain"**

# jQuery Mobile

◌ Ejemplo:

```
<div data-role="fieldcontain" class="ui-hide-label">  
  <label for="usuario">Usuario</label>  
  <input type="text" name="usuario"  
    id="usuario" placeholder="Usuario">  
</div>
```



# jQuery Mobile

## IDs de campos

- Dos elementos dentro de la misma aplicación **no pueden tener el mismo atributo id**
- JQM carga la página objetivo en el DOM de la página actual. Si dos páginas contienen elementos con el mismo **id** al cargar la segunda página en la primera se creará un documento inválido por el conflicto de los **ids**

# jQuery Mobile

## Disposición de formulario

- JQM muestra automáticamente pares etiqueta/campo dependiendo del ancho de la ventana del navegador
- Si la ventana es menor a 480px, la etiqueta aparecerá sobre el campo asociado
- Para ventanas más anchas, la etiqueta y el campo aparecerán uno al lado del otro

```

<!doctype html>
<html>
  <head>
    <title>Mi Formulario</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <link rel="stylesheet" href="css/jquery.mobile-1.2.0.min.css" />
    <link rel="stylesheet" href="css/jquery.mobile.structure-1.2.0.min.css" />
    <script src="js/jquery-1.8.3.min.js"></script>
    <script src="js/jquery.mobile-1.2.0.min.js"></script>
  </head>
  <body>
    <div data-role="page">
      <div data-role="header">
        <h1>Título del Formulario</h1>
      </div>
      <div data-role="content">
        <form action="index.html" method="post">
          <div data-role="fieldcontain">
            <label for="ctBuscar">Buscar:</label>
            <input type="search" name="cajaTextoBuscar" id="ctBuscar">
          </div>
          <div data-role="fieldcontain">
            <fieldset data-role="controlgroup">
              <legend>Seleccionar:</legend>
              <input type="radio" name="widgetShape" id="widgetShape-round" value="round" checked>
              <label for="widgetShape-round">Seleccionar 1</label>
              <input type="radio" name="widgetShape" id="widgetShape-square" value="square">
              <label for="widgetShape-square">Seleccionar 2</label>
            </fieldset>
          </div>
          <input type="submit" name="enviar" value="Enviar">
        </form>
      </div>
      <div data-role="footer">
        <h4>Pie del formulario</h4>
      </div>
    </div>
  </body>
</html>

```



# jQuery Mobile

## Estructura del formulario

- **Elemento form**, se crea de manera habitual con los atributos **action** y **method**
- El **data-role="fieldcontain"** permite agrupar etiquetas y elementos del formulario y es utilizado por JQM para mejorar su diseño visual

# jQuery Mobile

## Estructura del formulario

- El **data-role="controlgroup"** permite agrupar botones o selectores de modo que formen un único bloque al estilo de los componentes de navegación
- Las etiquetas **fieldset** y **legend** se utilizan en conjunto. **Fieldset** permite agrupar lógicamente a los elementos del formulario y **legend** permite etiquetar o nombrar al grupo creado con **fieldset**.

# jQuery Mobile

## Elementos de formulario

- Entre los elementos que podemos integrar en un formulario encontramos:
  - Campos y áreas de texto
  - Campos de contraseña
  - Campos HTML 5
  - Campos de búsqueda
  - Barras de desplazamiento de rango
  - Campos de tipo interruptor (SI/NO, ON/OFF)
  - Botones de opción
  - Casillas de verificación
  - Menús de selección



# jQuery Mobile

## Elementos de formulario

### ◦ Ejemplos:

#### ◦ Campo de texto

```
<div data-role="fieldcontain">  
  <label for="fullName">Nombre completo:</label>  
  <input type="text" name="fullName" id="fullName">  
</div>
```

#### ◦ Campo de contraseña

```
<div data-role="fieldcontain">  
  <label for="password">Contraseña:</label>  
  <input type="password" name="password" id="password">  
</div>
```

# jQuery Mobile

## Elementos de formulario

### ◦ Ejemplos:

#### ◦ Campo de correo

```
<div data-role="fieldcontain">  
  <label for="emailAddress">Dirección email:</label>  
  <input type="email" name="emailAddress" id="emailAddress">  
</div>
```

#### ◦ Campo de fecha

```
<div data-role="fieldcontain">  
  <label for="initDate">Fecha inicio:</label>  
  <input type="date" name="initDate" id="initDate">  
</div>
```

# jQuery Mobile

## Elementos de formulario

### ○ Ejemplos:

#### ○ Campo de búsqueda

```
<div data-role="fieldcontain">  
  <label for="searchClient">Buscar Cliente:</label>  
  <input type="search" name="searchClient"  
id="searchClient">  
</div>
```



# jQuery Mobile

## Elementos de formulario

### ◦ Ejemplos:

#### ◦ Casilla de verificación

```
<div data-role="fieldcontain">
  <fieldset data-role="controlgroup">
    <legend>Modo Turbo:</legend>
    <input type="checkbox" name="turboMode"
      id="turboMode-on" value="on">
    <label for="turboMode-on">On</label>
  </fieldset>
</div>
```

# jQuery Mobile

## Elementos de formulario

### ◦ Ejemplos:

#### ◦ Menú de selección simple

```
<div data-role="fieldcontain">  
  <label for="widgetColor">Seleccionar Color:</label>  
  <select name="widgetColor" id="widgetColor">  
    <option value="rojo">Rojo</option>  
    <option value="violeta">Violeta</option>  
    <option value="blanco">Blanco</option>  
    <option value="naranja">Naranja</option>  
  </select>  
</div>
```