

Tema 6 Parte III

Frameworks web para dispositivos móviles

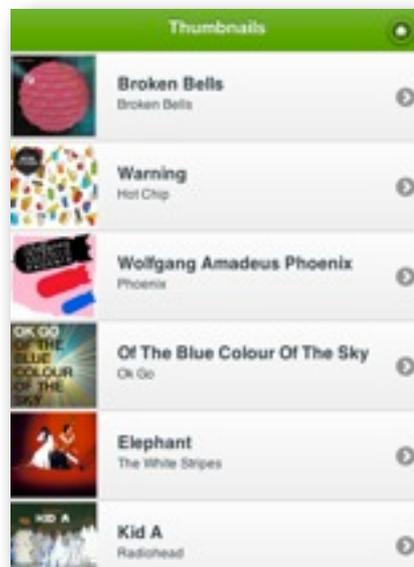


1

Introducción

Frameworks web para
dispositivos móviles

- Nos permiten desarrollar un sitio web con aspecto y funcionalidad típica de móviles sin tener que escribir todo el HTML/CSS/Javascript “desde cero”
 - Elementos de interfaz adaptado a móviles (botones grandes, listas,...)
 - Transiciones entre pantallas
 - Compatibilidad entre plataformas



Tipos de frameworks

“De marcado”

- Definimos los elementos de pantalla con etiquetas HTML convencionales, con clases CSS o atributos HTML propios del framework
- En la inicialización, el *framework* les asigna un aspecto (CSS) y un “comportamiento” (Javascript) especiales



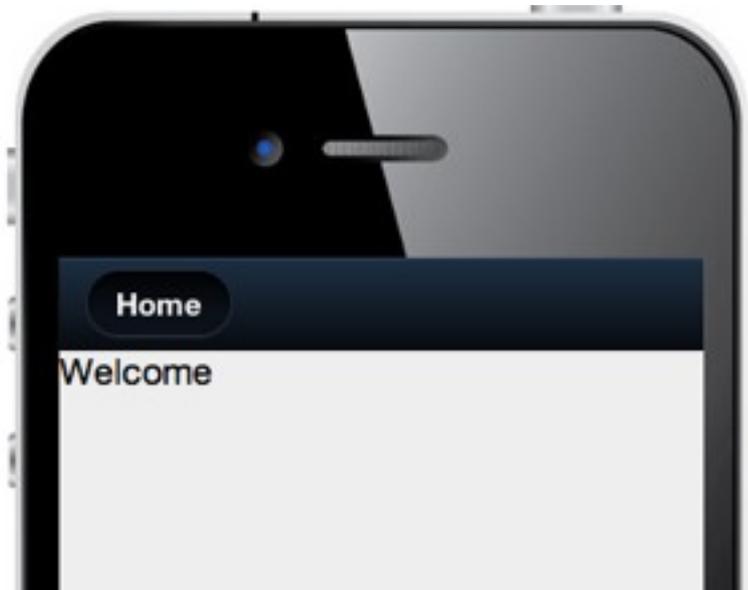
```
<!-- ejemplo con jQTouch (jQTouch.com) -->
<div id="home" selected="true" class="edgetoedge form">
<div class="toolbar">
  <h1>Todos</h1>
  <a href="#add" class="btn2 slideup">Add</a>
</div>
<ul>
  <li><input type="checkbox"/>Do something<small></small></li>
  <li><input type="checkbox"/>Do something else</li>
  <li><input type="checkbox"/>Finish these damn demos</li>
</ul>
<h4>Completed</h4>
<ul>
  <li><input type="checkbox" checked/></li>
  <li><input type="checkbox" checked />Write eblast</li>
</ul>
</div>
```

Tipos de frameworks (II)

- “De código”

- Más parecidos al concepto típico de “librería”. Cuando queremos definir por ejemplo un *widget* lo hacemos con código Javascript

```
Ext.application({
  name: 'Sencha',
  launch: function() {
    Ext.create("Ext.TabPanel", {
      fullscreen: true,
      items: [
        {
          title: 'Home',
          iconCls: 'home',
          html: 'Welcome'
        }
      ]
    });
  }
});
```



Algunos frameworks

- De marcado
 - **jQTouch** <http://dojotoolkit.org/features/mobile>
 - **jQuery Mobile** <http://jquerymobile.com/>
- Programativos
 - **Sencha Touch** <http://www.sencha.com/products/touch/>
 - **Jo** <http://joapp.com/>
 - **DOJO mobile** <http://dojotoolkit.org/features/mobile>

2

JQuery Mobile: conceptos básicos

Frameworks web para
dispositivos móviles

- Probablemente el *framework* web para móviles más conocido y difundido
 - <http://jquerymobile.com/>
 - Bajo el “paraguas” de jQuery
 - Multiplataforma: iOS, Android, WP7, ...
- Proporciona
 - widgets optimizados para móviles
 - personalización del estilo visual (temas)



- Incluir en el proyecto
 - El .js de jQuery
 - El .js, CSS e iconos de jQuery Mobile
- La mayoría de *widgets* se obtienen poniendo atributos especiales (`data-*`) a etiquetas HTML convencionales

```
<div data-role="header" data-position="inline">  
  <a href="index.html" data-icon="delete">Cancel</a>  
  <h1>Edit Contact</h1>  
  <a href="index.html" data-icon="check">Save</a>  
</div>
```

✕ Cancel

Edit Contact

✓ Save

- En un solo archivo HTML hay varias páginas (pantallas) de móvil
 - De este modo, se reduce el número de peticiones al servidor mientras navegamos por la aplicación
 - Cada pantalla será una etiqueta con **data-role="page"**
 - Habitualmente se usa <div>, pero se puede usar cualquier etiqueta
- Cada pantalla suele tener cabecera, contenido y pie

El “hola mundo” de jQuery Mobile

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hola mundo</title>
    <link rel="stylesheet"
          href="jquery.mobile-1.0.min.css"/>
    <script src="jquery-1.7.min.js"></script>
    <script src="jquery.mobile-1.0.min.js"></script>
    <meta charset="utf-8"/>
  </head>
  <body>
    <div data-role="page">
      <div data-role="header">
        <h1>Cabecera</h1>
      </div>
      <div data-role="content">
        <p>Esto es el contenido de la página</p>
      </div>
      <div data-role="footer">
        <h4>Pie</h4>
      </div>
    </div>
  </body>
</html>
```

Cabecera

Esto es el contenido de la página

Pie

- Entre otras cosas, cuando se carga el HTML
 - Se disparan eventos a los que si nos interesa podemos asociar *listeners* (`pageinit`, `pagebeforeshow`,...)
 - Se asocian clases CSS a los elementos con un data-role especial
 - Se asocian manejadores de evento javascript a los widgets
 - Se posicionan las cabeceras y pies

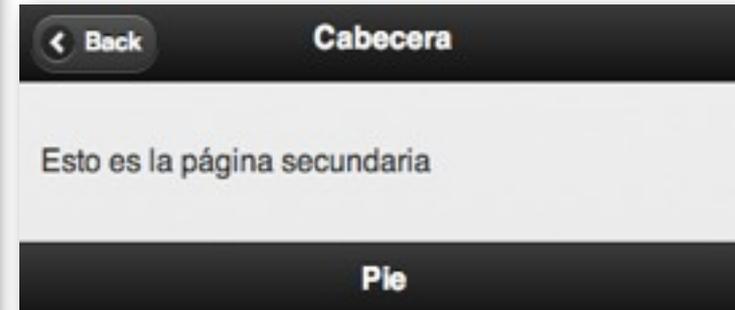
3

Navegación entre “páginas”

Frameworks web para
dispositivos móviles

Navegación entre páginas

```
<div data-role="page">
  <div data-role="header" id="principal">
    <h1>Cabecera</h1>
  </div>
  <div data-role="content">
    <p>Esto es el contenido de la página</p>
    <p>Ir a la
      <a href="#secundaria"
          data-transition="slideup">
        secundaria</a>
    </p>
  </div>
  <div data-role="footer">
    <h1>Pie</h1>
  </div>
</div>
<div data-role="page" id="secundaria"
  data-add-back-btn="true">
  ...
</div>
```

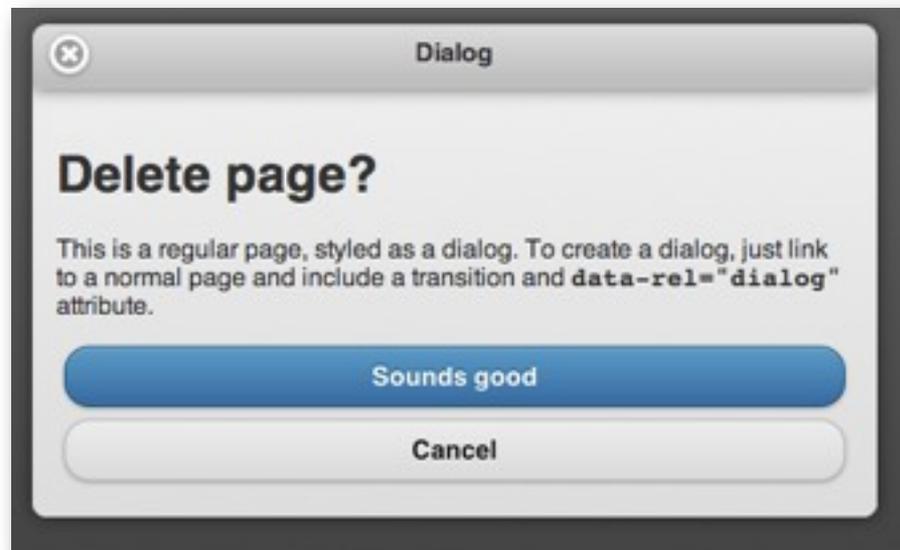


- **jQuery Mobile “secuestra” el evento de click sobre los enlaces**
 - Se dispara una transición CSS3 con la animación de cambio de página
 - ◊ Por defecto es slideright, pero también hay slideleft, slideup, slidedown, flip,...
 - Dinámicamente se cambian las clases CSS de la página actual para ocultarla y de la anterior para mostrarla
 - ◊ Disparando además una serie de eventos: pagebeforeshow, pagebeforehide, pageshow, pagehide
 - Se actualiza con javascript el historial de navegación (para que si pulsamos el “back” del navegador todo funcione correctamente)

- Si el href no lleva el “#”, estamos enlazando a otro HTML
 - Por defecto no se cambia de página
 - ◇ Automáticamente se hace una petición AJAX para obtener el nuevo HTML
 - ◇ Se “examina” el nuevo HTML en busca de la primera etiqueta con data-role=”page”
 - ◇ Se inserta dinámicamente esta etiqueta en el DOM del HTML actual, ignorando el resto
 - ◇ Se hace la transición a la nueva página
 - Podemos hacer una navegación convencional (cambiando totalmente de HTML)
 - ◇ Ponerle al enlace un `rel="external"` o un `data-ajax="false"`

- Páginas convencionales a las que se llega con un enlace con `data-rel="dialog"`
 - No se incluyen en el historial de navegación para que si hacemos “atrás” no volvamos a ellas

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



4

Elementos de interfaz de usuario

Frameworks web para
dispositivos móviles

• Etiquetas HTML convencionales

- Se les da un “look” de móvil y en algunos casos, funcionalidad táctil adicional

```
<div data-role="fieldcontain">
  <label for="usuario" class="ui-hidden-accessible">Usuario:</label>
  <input type="text" id="usuario" value="" placeholder="Usuario..." />
  <label for="flip">Guardar datos:</label>
  <select name="flip" id="flip" data-role="slider">
    <option value="no">No</option>
    <option value="si">Sí</option>
  </select>
</div>
```



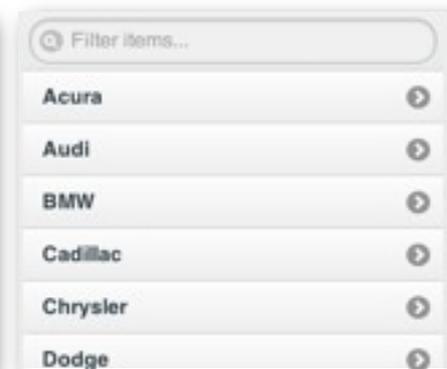
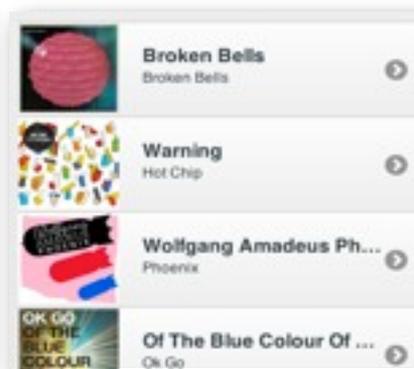
Usuario...

Guardar datos:

No

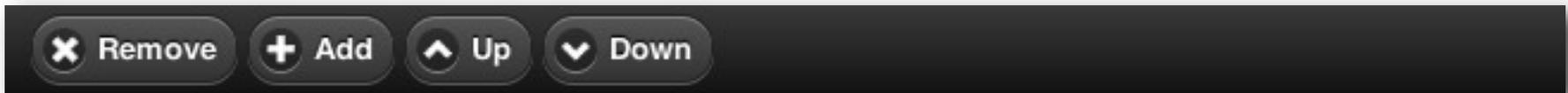
- En móviles es muy típico tener listas que ocupan toda la pantalla
 - Lista de tareas, contactos, eventos cercanos ...
 - ◊ En jQuery Mobile, `` con `data-role="listview"`
 - También es muy típico que al pulsar en un elemento de la lista vayamos a otra pantalla para ver los detalles
 - ◊ En jQuery Mobile, se pone simplemente un enlace dentro de cada `` a la página adecuada

- La variedad y sofisticación de las listas en jQuery Mobile es considerable. **Algunas** posibilidades:
 - Thumbnails (dentro de cada , jQuery automáticamente ajustará el tamaño)
 - Separadores (etiquetas con `data-role="list-divider"`)
 - “Badges” (típica “burbuja” que indica cuántos elementos hay): `class=ui-li-count`
 - Filtrar automáticamente elementos: lista con `data-filter="true"`

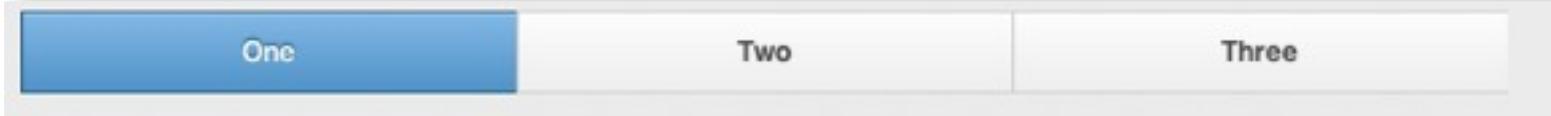


Barras de botones

```
<div data-role="footer" class="ui-bar">
  <a href="index.html" data-role="button" data-icon="delete">Remove</a>
  <a href="index.html" data-role="button" data-icon="plus">Add</a>
  <a href="index.html" data-role="button" data-icon="arrow-u">Up</a>
  <a href="index.html" data-role="button" data-icon="arrow-d">Down</a>
</div>
```



```
<div data-role="navbar">
  <ul>
    <li><a href="a.html" class="ui-btn-active">One</a></li>
    <li><a href="b.html">Two</a></li>
    <li><a href="b.html">Two</a></li>
  </ul>
</div>
```



5

Eventos

Frameworks web para
dispositivos móviles

- Usar el `$(document).ready` para inicializar **aspectos globales** de la aplicación
- Usar el evento `pageinit` de cada página para inicializar lo propio de ella
 - Por ejemplo vincular un *listener* a un botón
- Usar el `pagebeforeshow` para inicializar lo que puede cambiar cada vez que se muestra
 - Por ejemplo, si es una lista generada dinámicamente, insertar los ``

Ejemplo

```
//Esto se ejecuta una sola vez para el documento entero
$(document).ready(function() {
    //personalizar el mensaje que sale cuando se esperan datos AJAX
    $.mobile.loadingMessage = "Cargando datos"
    //personalizar el texto del botón para ir a la página anterior
    $.mobile.page.prototype.options.backBtnText = "Atrás"
})

//Esto se ejecuta una sola vez para cada página, una vez inicializada
//por jQuery (añadidas las clases CSS, ...)
$('#pagBusqueda').bind('pageinit', function() {
    $('#botonBuscar').click(buscar)
})

//Aquí rellenaríamos la página "detalles" con los datos pertinentes,
//justo antes de mostrarla
$('#pagDetalles').bind('pagebeforeshow', function() {
    ...
})
```

- Aquí hemos visto solo unos cuantos conceptos, porque la cantidad de atributos `data-*` es considerable
- La documentación de jQuery Mobile no es tan exhaustiva como la de jQuery, pero tiene muchas demos integradas
 - <http://jquerymobile.com/demos/1.0/docs>

Referencias: libros

- <http://proquestcombo.safaribooksonline.com/> (solo desde la UA)
- *jQuery Mobile*, Jon Reid, O'Reilly 2011
- *Pro Android Web Apps, Develop for Android Using HTML5, CSS3 & JavaScript*, Apress 2010
 - ◆ Información sobre jQuery Mobile y otros frameworks como Sencha touch o Jo

