



Tema 6

Parte I

# Introducción al desarrollo de aplicaciones web móviles

Introducción. HTML y CSS para aplicaciones web móviles

Desarrollo de  
aplicaciones web móviles

1

¿Qué tienen de  
especial los  
dispositivos  
móviles...

...con respecto a los  
equipos “de escritorio”?

# Dispositivos móviles

- Mercado muy **heterogéneo**
  - Móviles “convencionales” (gama baja-media-alta)
  - *Smartphones*: iPhone, Android, algunos Nokia
  - Otros dispositivos: tablets, ebooks
- Nos vamos a centrar en **smartphones**, por cuestiones de tiempo y espacio
  - Android, iOS, WP7, Blackberry
  - Esto no quiere decir que los demás no sean importantes en el **mundo real**



- Lo más inmediato, la pantalla
  - La resolución varía de 128x128 (gama baja) a 640x960 (iPhone 4), siendo la media de 240x320 (QVGA o Quarter VGA), y **320x480** (HVGA o Half-VGA) o más
  - Aunque la resolución sea alta no podemos colocar **demasiada información** ya que la pantalla es físicamente muy pequeña
  - Más alta que ancha, al contrario que en el escritorio.
    - ◆ Es incómodo hacer *scroll* horizontal, por lo que la información tiende a colocarse en **una columna**



- Tipos
  - Teclado (numérico y cursores, alfanumérico)
  - Pantallas táctiles (dedo y/o *stylus*)
  - Reconocimiento de voz y/o de escritura
- En general, introducir datos en un móvil es **incómodo y tedioso**.
  - La aplicación debe reducir esta necesidad al mínimo imprescindible
- Hay que tener en cuenta las características del dispositivo de entrada
  - Los controles táctiles tienen que ser grandes, para poder ser pulsados con el dedo. No así si puede usarse un *stylus*

# Recomendaciones de estilo

- El espacio (sobre todo horizontal) es muy limitado
  - Eliminar información accesoría
  - Colocar la información más importante al principio
  - No usar columnas (tablas, etc)
- La entrada de datos y navegación puede ser tediosa
  - Hacer fáciles las tareas comunes
- Si se usa una interfaz táctil
  - Dejar un área suficiente entre botones o enlaces
  - Pensar qué área tapará el dedo del usuario al pulsar



- Apple Human Interface Guidelines for Web Applications
- Android Guidelines for Web Apps [http://  
developer.android.com/guide/webapps/index.html](http://developer.android.com/guide/webapps/index.html)
- Recomendaciones del W3C para la web móvil. Versión resumida en castellano en [http://www.w3c.es/Divulgacion/  
Tarjetas/MWBP/](http://www.w3c.es/Divulgacion/Tarjetas/MWBP/)
- “Diseñar para móviles” [http://www.slideshare.net/obaba/  
diseñar-para-mviles-8663126](http://www.slideshare.net/obaba/diseñar-para-mviles-8663126) (muy buena presentación con consejos, muchas referencias al final de la presentación)

2

# Programación de aplicaciones para móviles

Aplicaciones nativas vs.  
web

Desarrollo de  
aplicaciones web móviles



## ● Aplicaciones “nativas”

- Aplicaciones del SO del móvil. Por ejemplo
  - ◆ **Android:** desarrollo en Java, con APIs propios del sistema
  - ◆ **iOS:** Objective-C, librerías Cocoa Touch
  - ◆ **WP7**
- **Runtimes:** capas por encima del SO, permiten una mayor portabilidad
  - ◆ **JavaME:** el *runtime* más difundido del mundo...pero con poco futuro. Carente de soporte táctil estándar y de actualizaciones constantes

## ● Aplicaciones web

- Sitios web convencionales usando Javascript+HTML. Si el dispositivo soporta HTML5 pueden funcionar *offline* gracias a la cache

- No es necesario aprender APIs propios del dispositivo, ni nuevos lenguajes de programación
- La aplicación es fácil de distribuir y actualizar a la nueva versión
  - La instalación solo requiere una visita al sitio web. La actualización es automática usando cache de HTML5
  - No hay barreras artificiales a la distribución (ej. App Store de Apple)
- Es portable entre todos los dispositivos independientemente del SO

- Por cuestiones de rendimiento, no son apropiadas para algunos tipos de aplicaciones (juegos con gráficos intensivos, multimedia, ...)
- No son accesibles todas las capacidades del dispositivo
  - Aunque hay un API estándar de geolocalización, no lo hay de cámara o acelerómetro, por ejemplo
- Es difícil darle a la interfaz el “look” propio de la plataforma
- Es más complicado obtener un provecho económico
  - Las aplicaciones web no se pueden distribuir directamente en la App Store o en el Android Market, por ejemplo.

- Últimamente han surgido herramientas que permiten desarrollar aplicaciones web con ciertas ventajas de las nativas
  - Por ejemplo acceso a ciertos APIs con Javascript: sonido, vibración, acelerómetro, ...
  - El “truco” consiste en que se genera una aplicación nativa pero que es una ventana del navegador
- Ejemplo: <http://www.phonegap.com>

3

# Navegadores en dispositivos móviles

Desarrollo de aplicaciones web móviles

# Algunos navegadores para móviles

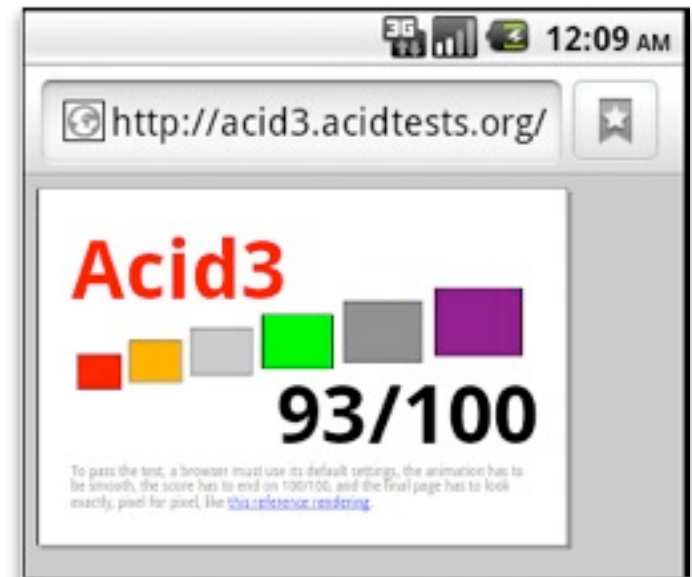
- **Gama baja/media**
  - Netfront (Sony Ericsson, Samsung, LG)
  - OpenWave/Myriad (Motorola, LG, Sharp,...)
- **Nokia**
  - S40: basado en foco. A partir de 5a ed, igual a S60
  - Nokia S60 (Symbian browser): cursor y táctil
- **Mobile Safari (iPhone)**
- **Android**
- **Blackberry**
- **Internet Explorer Mobile - WP7 (similar al IE de escritorio pero táctil)**
- **Instalables por el usuario**
  - Opera Mobile (win, Symbian)
  - Firefox mobile



- Es el motor de renderizado que usa la mayoría de navegadores móviles. Original de Apple, ahora open source (<http://webkit.org>)
  - Nokia, iPhone, Android, Blackberry,...en escritorio, Chrome y Safari
- No obstante, no todos los navegadores usan la misma versión, y algunos incorporan extensiones propias
  - Tabla de compatibilidad
    - ◊ <http://www.quirksmode.org/webkit.html>



- WebKit ofrece un buen soporte de estándares HTML/CSS
- Ejemplo: test **ACID3**: comprueba DOM1, DOM2, selectores CSS3, javascript,...
- iOS 4, Android 4.0 lo pasan 100/100





4

# HTML para dispositivos móviles

Desarrollo de  
aplicaciones web móviles

- La mayoría de móviles son compatibles con HTML 4 y los smartphones (android, iPhone, blackberry) con HTML5
- No obstante, existen lenguajes de marcado específicos de móviles
  - XHTML MP (Mobile Profile) y WAP CSS (OMA-Open Mobile Alliance)
  - XHTML Basic y CSS MP (W3C)

- El estándar *de facto* en dispositivos de gama media
  - Es como una versión reducida del XHTML de escritorio ( XHTML Basic es un poco más limitado)
  - Carece de elementos como
    - ◆ Ciertas etiquetas de tabla como `thead`, `tbody`, ...
    - ◆ No se pueden anidar tablas (*¿quién querría, de todos modos?*)
    - ◆ Campos de `type="file"`

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN"
"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Prueba de HTML-MP</title>
</head>
<body>
  <h1>Prueba para móviles</h1>
  <p>Como se ve, es muy parecido al XHTML de "escritorio"</p>
</body>
</html>
```

# Viewport

- Los móviles hacen un escalado para mostrar más píxeles de los reales (para que una página diseñada para escritorio se “vea completa”)
  - Viewport por defecto: iOS 980px, android 800px
- Podemos controlar el viewport

```
<meta name="viewport" content="width=320" />
```

```
<!-- darle el ancho de la pantalla -->  
<meta name="viewport"  
      content="width=device-width" />
```

```
<!-- impedir que el usuario pueda hacer zoom -->  
<meta name="viewport"  
      content="width=device-width;user-scalable=no" />
```

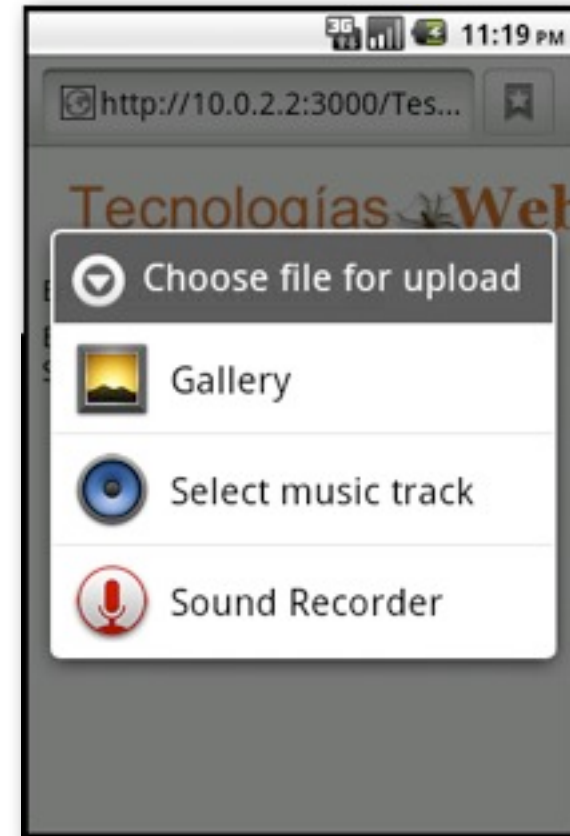


- Dado el poco espacio horizontal
  - No usar tablas para *layout*
  - Usar atributo **placeholder** para “etiquetas” de campo (solo en compatibles con HTML5, se puede imitar con JS)
- Dadas las dificultades de entrada
  - Reducir las entradas al mínimo
  - Usar selects en lugar de radio buttons o checkboxes (select múltiple)
  - ¿Son necesarios los `type="password"`?

# Campos tipo <input>

- Aunque el `input type="file"` no es estándar en XHTML-MP, muchos navegadores lo aceptan
  - En iOS no funciona.
  - En Android se pueden subir solo imágenes o sonidos
- Podemos definir "máscaras de entrada" para los campos con WAP CSS
  - No en iOS o Android)

```
<input type="text" name="tfno" style="-wap-input-format:'9N'; -wap-input-required: true">
```



- Recordad que en HTML5 se pueden usar `type="email"`, `type="number"`, `type="url"`, `type="tel"`.
  - El dispositivo mostrará un teclado virtual adaptado (iOS, Android)
- Atributo `autocomplete=off` para que no nos moleste con sugerencias

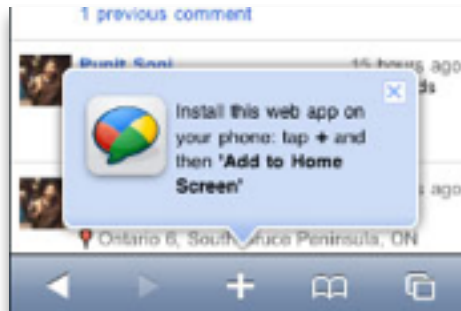


# Algunos “trucos”

- Se puede hacer que una aplicación web tenga un icono en el “escritorio” móvil, como si fuera una aplicación nativa

```
<!-- solo iPhone, el sistema le añade ciertos
efectos visuales (brillo, ...)-->
<link rel="apple-touch-icon" href="icono.png" />
<!-- iPhone y Android -->
<link rel="apple-touch-icon-precomposed"
```

- El usuario debe añadir manualmente el “acceso directo”
  - ◆ *Script* ejemplo: “Mobile bookmark bubble” <http://code.google.com/p/mobile-bookmark-bubble/>





# Algunos “trucos” (II)

- Enlaces especiales para teléfonos
  - Se llamará o enviará SMS/MMS al pinchar en el enlace

```
<a href="tel:+34965903400">Universidad de Alicante</a>  
<a href="sms:+555666333111">Envíame un SMS</a>  
<!-- estándar WTAI (Android, Symbian, Blackberry,...) -->  
<a href="wtai://wp/mc;+34965903400">Llámame a la UA</a>
```

5

# CSS para dispositivos móviles

Desarrollo de aplicaciones web móviles

- **Compatibilidad:**
  - Casi todos los dispositivos modernos son compatibles con CSS2
  - Los *smartphones* con CSS3
- Recordar que podemos enlazar una hoja de estilos distinta para cada dispositivo de salida
  - En teoría `media="handheld"` es para móviles, pero en la práctica los dispositivos más modernos usarán el CSS asociado a `"screen"`

```
<link rel="stylesheet" media="screen" href="desktop.css" />  
<link rel="stylesheet" media="handheld" href="mobile.css" />
```

- Estándar de CSS3, nos permite aplicar un CSS en función de una serie de condiciones:

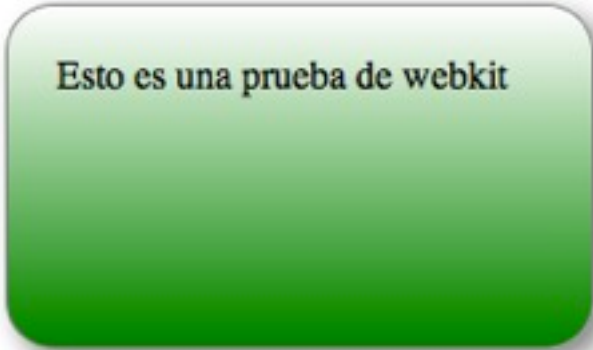
```
<link media="screen and (max-device-width:480px) "  
href="iphone.css" rel="stylesheet"/>  
<link media="screen and (min-device-width: 481px) "  
href="desktop.css" rel="stylesheet" />
```

- Extensión al estándar, para detectar la orientación actual del dispositivo

```
<link media="screen and (orientation:landscape) "  
rel="stylesheet" href="land.css" />  
<link media="screen and (orientation:portrait) "  
rel="stylesheet" href="port.css" />
```

- Como casi todos los navegadores móviles modernos se basan en webkit, podemos hacer uso de sus extensiones todavía no totalmente estándar

```
.boton {  
  width: 200px;  
  height: 100px;  
  border-radius: 20px;  
  padding: 20px;  
  border: 1px solid grey;  
  -webkit-box-shadow: grey 5px 5px 10px;  
  background-image:-webkit-gradient(linear,  
    left top,  
    left bottom,  
    from(white),  
    to(green)  
  )  
}
```



Esto es una prueba de webkit

- La mayoría de dispositivos solo tiene una fuente disponible (Android, Symbian) o muy pocas
- `position:fixed` no funciona
  - Sería muy útil para barras de botones :(
  - Se puede usar javascript para “fijar” algo mientras se hace *scroll*, (en realidad, ir desplazándolo a medida que se desplaza la página)

# Plantillas HTML/CSS para móviles

- Una plantilla con
  - Scripts, etiquetas HTML, estilos CSS, ... útiles para móviles
- Ejemplo: mobile boilerplate
  - <http://html5boilerplate.com/mobile>

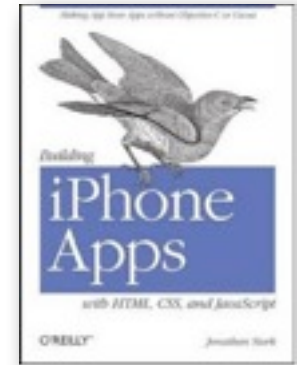


## Jonathan Stark:

- *Building iPhone Apps with HTML, CSS and Javascript*
  - ◆ Versión completa libre acceso en <http://ofps.oreilly.com/titles/9780596805784/>
- *Building Android Apps with HTML, CSS and Javascript*
  - ◆ Versión completa libre acceso en <http://ofps.oreilly.com/titles/9781449383268/>

## Los fabricantes suelen tener sitios web para desarrolladores con documentación, ejemplos, simuladores, etc.

- En general [developer.xxx.com](http://developer.xxx.com), por ejemplo <http://developer.nokia.com> o <http://developer.android.com>





# Referencias (II)

- <http://proquestcombo.safaribooksonline.com/> (solo desde la UA)
- *Build Mobile Websites and Apps for Smart Devices*, Ed. SitePoint, 2011
- *Programming the mobile web*, O'Reilly 2010

