

Introducció a les aplicacions per a telefonia mòbil

Consultor: Roman Roset roset@uoc.edu

1. Tipus d'aplicacions que podem trobar en un dispositiu

Les aplicacions que podem trobar en un mòbil, son en general, del mateix tipus que trobem en un pc ([veure wikipedia](#)). Si les classifiquem en funció de la comunicació amb un servidor, podem trobar:

- Aplicacions d'escriptori: l'aplicació funciona en l'escriptori de cada usuari. En principi, poden usar o no comunicació amb un servidor, però la clau és que s'instal·len i s'executen dins l'escriptori del dispositiu.
- Aplicacions basades en web: També conegudes com aplicacions en el núvol. No confondre amb llocs web, donat que els web-site, no són aplicacions personals per a dispositius.

El projecte que nosaltres farem es basa en usar tecnologia d'aplicacions basades en web (que és quasi la mateixa que els web-site) per a fer aplicacions d'escriptori. Per tant, el nostre producte és un producte que s'instal·la en un dispositiu i funcionarà offline.

2. Com es programen aplicacions per a mòbils? 3 tipus d'enfocs

Llegir i analitzar els següents textos:

1. Mobile Application Development - Native or Web? ([Part 1](#), [Part2](#), [Part3](#))
2. [Mobile Application Development: Web vs. Native](#)

Punts importants d'anàlisi d'aquest text:

- Conèixer els dos enfocs: nadius i aplicacions web per a mòbils.
- Conèixer els avantatges i inconvenients dels dos
- Veure que hi ha un tercer tipus híbrid: El que busca programar en un sol llenguatge y interpretar en més d'un dispositiu.
- Conèixer els factors decisius en el desenvolupament de tecnologia mòbil: experiència de l'usuari, el rendiment

Per a més informació, només per a que la considereu si la necessiteu: [documents oficials sobre dispositius](#). (Veure també el “beyond Mobile Boilerplate” en [mobile boilerplate documentation](#))

El nostre projecte és un desenvolupament híbrid donat que ho programarem com a una aplicació web i funcionarà com una aplicació nativa. La forma de convertir una aplicació web

en nativa és utilitzant una eina que empaqueta els fitxers web en una aplicació nativa. Les més famoses de totes són el [Sencha touch](#), el [phoneGap](#) i el [titanium appcelerator](#).

3. Elements de les aplicacions web per a mòbils

Els elements per a implementar aplicacions web per a mòbils són:

- HTML5
- CSS3
- JAVASCRIPT

En les pròximes unitats parlarem sobre cadascun d'aquests elements.

Introducció no tan breu sobre html5 i css3

1. Objectius d'aquesta unitat

Conèixer les principals característiques del html5 que el fan apropiat per fer aplicacions web i simular que son d'escriptori.

Hi ha moltíssima informació en internet sobre aquest tema. Si no ens centrem en els objectius segurament mirarem moltes més coses de les que ens seran d'utilitat. Per tant, preneu aquesta unitat com un tast, una iniciació de html5 i css3. Focalitzeu-vos en els vostres propis objectius, i no us allunyeu massa d'aquests.

No és la meua intenció que us memoritzeu i llegiu de principi a final els diferents recursos i textos que adjuntaré. Fer-los servir de punts d'informació claus per a navegar quan necessiteu la informació adient.

2. Immersió de contacte brusc al html5

Lectura ràpida del següent document:

1. [Presentació HTML5](#)
2. [Dive into html5](#) Centrarse en:
 - a. <http://diveintohtml5.org/introduction.html>
 - b. <http://diveintohtml5.org/semantics.html>
 - c. <http://diveintohtml5.org/offline.html>

- d. <http://diveintohtml5.org/forms.html>

Punts importants d'anàlisi d'aquest text:

- Les noves funcionalitats que ofereix html5 (offline i gràfics sobretot)
- Les noves paraules claus semàntiques d'html5
- Els elements dels formularis
- Les funcionalitats amb els CSS3

Quedeu-vos amb les coses immediates que esteu segurs fareu servir en la vostra aplicació, deixeu les altres per més endavant.

3. Anem a veure-ho funcionant més d'a prop

Aneu a aquesta web:

- <http://www.html5rocks.com/en/>

Com veieu es un lloc on la informació està molt ben organitzada per funcionalitats. Hi ha tutorials i laboratoris. El que us proposo es anar a fer una ullada de les diferents funcionalitats que hem estat llegint en la secció "features":

- <http://www.html5rocks.com/en/>

Centreu-vos en el html i el CSS i deixeu el javascript per més endavant, donat que ens donarà per diversos temes. Penseu que tot el que puguin fer el html i el css3 és feina que no hem de fer en javascript i a més a més permet que l'aplicació sigui més eficient (en general).

Sobre el CSS el que hem de considerar és que per a executar-se en el navegador, cada fabricant usa un render diferent, i per tant, en molts casos hem de descriure la mateixa funció en més d'un render. Us deixo per a vosaltres que mireu quin render usa en general Android i quin render de CSS usa l'iPhone. (Nota: en les aplicacions de mòbil, no té sentit renderitzar per a mozilla).

4. Anem a fer proves: hands on!

4.1 Que provem?

Per a fer proves necessitem saber què i com. El que ho podeu triar vosaltres. Jo us suggereixo triar algun tutorial d'aquí:

- <http://www.html5rocks.com/en/tutorials/flexbox/quick/>
- <http://www.html5rocks.com/en/tutorials/canvas/integrating/>
- <http://css-tricks.com/examples/CSS3Gradient/>
- <http://css-tricks.com/examples/ResponsiveTables/responsive.php>
- <http://css-tricks.com/examples/SlideupBoxes/>

- <http://www.html5rocks.com/en/tutorials/notifications/quick/>

4.2 Com?

Sempre que es pugui, intenteu fer les proves sense usar un servidor web. De fet, entrarem més en detall en la unitat sobre TDD, però si hi ha crides al servidor en json, xml, etc... intentarem fer servir [objectes simulats \(mock\)](#). Per tant, farem les proves en el mateix sistema de fitxers, amb arxius *.html, *.css. De fet, això no hauria de canviar en tot el PFC.

Si féssim servir un servidor web, voldria dir que el que tenim és una aplicació web que retorna una pàgina html i per tant no necessitaríem una plataforma per smartphones (directament ens connectaríem al servidor de l'aplicació)... tampoc pretenem crear llocs webs que es visualitzin de forma correcta en el mòbil... per tant, si necessitem un servidor web per a que ens retorni pàgines senceres d'html llavors, hauríem d'intentar transformar les pàgines en estàtiques i obtenir el contingut amb serveis web. En cert sentit, crec que això facilita les coses.

També us animo a que considereu fer servir api's obertes (com per exemple la api del twitter). Teniu exemples d'api's en: <http://www.programmableweb.com/apis>

4.3 Quin editor usem?

Per a fer aquestes proves... el que vulgueu! Si heu instal·lat ja l'entorn Aptana... eclipse és perfecte. Si voleu quelcom més senzill: notepad++.

Ara bé, hi ha un editor per a fer proves magnífic en el núvol: [jsfiddle](#). Tasteu-lo!

4.4 Quin navegador?

Ahà! Depen. Per a quin dispositiu desenvoluparé? Si és un Android, llavors el seu navegador es basa en webkit pel css i en Chrome v8 per JavaScript. Si és Iphone, llavors el seu navegador es basa en safari que usa webkit2 i Nitro Javascript Engine.

4.5 I si som d'aquells que ens agrada depurar?

Si depurem amb Chrome, llavors usarem això: [eines per desenvolupadors en chrome](#). Proveu-la! Podem analitzar el html, el css, el rendiment, etc...

Si depurem per a safari (o per firefox) podem usar la següent extensió: [buscar firebug](#).