

# Dissenyar gràficament una aplicació de mòbil

Consultor: Roman Roset [roman@uoc.edu](mailto:roman@uoc.edu)

## 1. Objectius d'aquesta unitat

Quin és el procés que segueix el disseny gràfic, quines eines usen els dissenyadors i quines consideracions hem de fer per al nostre projecte, són els objectius d'aquesta unitat.

Anem a començar. Però abans, un apunt pel que fa al nostre projecte àgil:

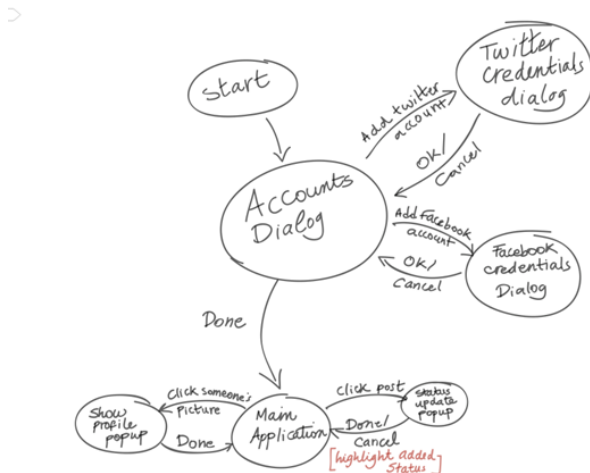
- Farem el disseny per a cada història d'usuari, és a dir, cada història d'usuari ha de tenir el seu disseny.
- Hem d'estimar el cost de la història d'usuari en dies ideals, tenint en compte el disseny de la història d'usuari.

## 2. Procés del disseny gràfic

Personalment m'agrada distingir dos moments o etapes diferents:

1. El flux de pantalles ( en cada història d'usuari).
2. El disseny de pantalles (en una història d'usuari).

Amb el flux de pantalles vull dir:



Amir khella

Amb el disseny de pantalles vull dir:



Sobre el diagrama de flux no hi ha cap secret, un diagrama de fluxe, un graf dirigit, etc... ens serviran.

Sobre com fer el disseny de les pantalles, depèn de les preferències de cada dissenyador. Actualment s'estan treballant sobre tres enfocis principals no exclusius:

- Paper
- Digital amb eina de disseny vectorial
- Digital amb aplicació de wireframes

### 3. Dissenyar en paper

Us agraden les manualitats? Per què en paper? Són absolutament útils en el disseny d'aplicacions mòbils, per què pots treballar amb mides reals i tocar un prototip físic del producte.

Dos introduccions (amb mètode inclòs) de l'ús del paper:

- <http://uxmag.com/articles/paper-in-screen-prototyping>
- <http://www.metasite.it/en/news/paper-prototyping-with-paper-iphone-vertical-tube-version-for-mobile-web/90>

Ara ja sabem com fer per a tenir un mòbil de categoria! Mai s'espatlla! ☺ Ara mirarem quines eines tenim per construir les pantalles.

- Ús d'[stencils](#):

- <http://www.uistencils.com/products/iphone-stencil-kit>

b. Imprimir, retallar, enganxar:

- <http://webdesignledger.com/freebies/10-free-printable-web-design-wireframing-templates>
- <http://www.geekchix.org/blog/2010/01/03/a-collection-of-printable-sketch-templates-and-sketch-books-for-wireframing/>
- <http://developer.yahoo.com/ypatterns/about/stencils/>

Però de totes, totes, el lloc on trobareu més stencils d'imprimir, enganxar és aquí:

- <http://graffletopia.com/categories/iphone>

Bé, ja veieu com és el mètode. Si us agrada, llavors, trieu el kit d'stencils que mes ús convingui i a dissenyar pantalles! (que fàcil que és així ☺ )

## 4. Digital amb eina de disseny vectorial

Ja les coneixeu no? El photoshop (aquesta no és vectorial però s'usa), el inkscape, fireworks, etc... El dissenyador fa directament el prototip des de zero, o agafa una plantilla per copiar i enganxar digitalment. A continuació us enumero algunes plantilles:

- <http://keynotopia.com/themes/>
- <http://www.mercuryintermedia.com/blog/index.php/2009/03/iphone-ui-vector-elements>
- <http://www.teehanlax.com/blog/iphone-gui-psd/>
- <http://www.designerstoolbox.com/designresources/iphone/>
- <http://www.smashingmagazine.com/2008/11/26/iphone-psd-vector-kit/>
- <http://www.morford.org/iphoneosdesignstencil/>
- <http://speckyboy.com/2010/04/30/iphone-and-ipad-development-gui-kits-stencils-and-icons/>

### 1. Digital amb eines de wireframe o mockups

Què són els wireframes? [Wikipedia](#): En disseny web, un wireframe és una representació esquemàtica d'una pàgina web sense elements gràfics que mostren contingut i comportament de les pàgines. Serveixen com a eina de comunicació i discussió entre arquitectes d'informació, programadors, dissenyadors i clients. També es poden utilitzar per comprovar la usabilitat d'un lloc web.

Mockup (maqueta) : En la fabricació i disseny, una maqueta, o d'una maqueta, és una escala o mida completa model d'un disseny o un dispositiu, utilitzat per a l'ensenyament, demostració, avaluació del disseny, promoció, i per a altres fins. Una maqueta que es diu un prototip si s'ofereix almenys una part de la funcionalitat d'un sistema i permet la inspecció d'un disseny.

Hi ha eines per a fer wireframes i mockups digitals exclusives per iphone o android? Si. No n'hi ha gaires però si. Us deixo un exemple per a iphone:

<http://iphonemockup.lkmc.ch/>

I un post d'un blog amb una llista:

- <http://iphoneized.com/2009/11/21-prototyping-mockup-wireframing-tools-iphone-app-development/>

## 5. Consideracions finals

Algunes consideracions finals:

- Hauríem de fer dissenys horitzontals i verticals. Però per no augmentar la complexitat, proveu primer amb el vertical. En el moment de tenir que fer el disseny en vertical hi dos enfocis per detectar el canvi d'orientació: fer-ho amb javascript i amb media-queries de css.:
  - Amb javascript: <http://net.tutsplus.com/tutorials/tools-and-tips/learn-how-to-develop-for-the-iphone/>
  - Amb css: <http://lessframework.com/>
  - Amb sencha hi ha events que avisen del canvi d'orientació.
- Busqueu inspiració en les botigues android, itunes, i webs d'aplicacions.
- Sobre l'ús de frameworks en css, com [compass](#), [sass](#), [less](#). No els explico per què no crec que els necessitem. Vull dir, que són d'ús per a tots els navegadors i això no és òptim. Hauríem d'escriure el css oportú per a cada webkit que vulguem usar. No obstant ara que sabeu això sigueu lliures d'usar el que vulgueu :) Qui utilitzi Sencha Touch, llavors, haurà d'usar compass. Teniu un tutorial aquí:
  - <http://www.netmagazine.com/tutorials/styling-user-interface-sencha-touch-application>

## 6. Recursos

### Icones

<http://raphaeljs.com/icons/>

<http://thenounproject.com/es/>

### Exemples de codi

<http://try.sencha.com/touch/2.0.1/>

<http://www.sencha.com/blog/meetcha-using-sencha-touch-to-build-a-mobile-app-for-meetup-com>

### **Inspiration**

<http://iphone.meer.li>

<http://mobile-patterns.com>

# Preparació de l'entorn de desenvolupament

## 1. Objectius d'aquesta unitat

Finalitzarem aquest seguit d'unitats d'aprenentatge amb una sèrie de recomanacions, propostes i suggeriments.

Abordarem aquests dos temes:

- Preparació de l'entorn de desenvolupament
- Preparant la memòria del PFC.

En el primer punt parlarem sobre com hauria de ser una bon entorn de desenvolupament i veurem la necessitat d'un control de versions pel projecte, i seguint la filosofia TDD com presentar aquest entorn de desenvolupament per al treball col·laboratiu. En concret parlarem directament de la plataforma github i el que s'espera trobar quan s'escriu un projecte en equip.

En el segon punt parlarem de com preparar una memòria i recordarem que el que nosaltres pretenem no és escriure el millor programa del món, sinó el millor projecte que puguem fer. Les dues coses s'assemblen, però son un pèl diferents.

## 2. Preparació de l'entorn de desenvolupament:

### 2.1 Els sistemes de control de versions

Copio de la wikipedia: [http://es.wikipedia.org/wiki/Control\\_de\\_versiones](http://es.wikipedia.org/wiki/Control_de_versiones)

Una versió, revisió o edició d'un producte, és l'estat en què es troba aquest producte en un moment donat del seu desenvolupament o modificació. Es diu control de versions a la gestió dels diversos canvis que es realitzen sobre els elements d'algun producte o una configuració d'aquest. Els sistemes de control de versions faciliten l'administració de les diferents versions de cada producte desenvolupat, així com les possibles especialitzacions realitzades (per exemple, per a algun client específic).

El control de versions es realitza principalment en la indústria informàtica per controlar les diferents versions del codi font. No obstant això, els mateixos conceptes són aplicables a altres àmbits com documents, imatges, llocs web, etc.

I Per què el volem nosaltres un control de versions? Doncs, és necessari per rastrejar els canvis i evitar el caos general. Un bon CVS fa el següent:

- **Backup and Restore.** Els arxius es guarden a mesura que s'editen, i es pot anar a qualsevol moment del temps. Necessitem l'arxiu del mes passat? Cap problema.

- **Sincronització.** Permet a l'equip estar al dia amb l'última versió.
- **Desfer a curt termini.** Si per equivocació hem pujat un arxiu que funciona malament el llencem a les escombraries i tornem a la versió "última bona coneguda".
- **Control de canvis.** A cada canvi podem incloure un comentari dient les noves funcionalitats que hem afegit amb una nota incorporada. Això fa que sigui fàcil de veure com s'està desenvolupant un arxiu al llarg del temps.
- **Seguiment de qui està fent un canvi.** Cada persona que agafa un tros de codi per escriure una nova funcionalitat ho comunica a la resta de l'equip, i també serveix per a fer el seguiment.
- **Sandbox, o l'assegurança contra tu mateix.** Fer un gran canvi? Això ho arreglo amb una nit!! I s'embolica la troca (o no!) Si es poguessin fer canvis temporals en una zona aïllada llavors seria fantàstic
- **Branques i la merging.** Podem fer versions del projecte a través de branques. Un branca per android, una altra per ios, etc... Si més tard volem passar una branca al master, llavors, cap problema.

## .2 Alguns sistemes de control de versions

A continuació us deixo un post que detalla els CVS més populars que s'usen actualment:

- <http://www.smashingmagazine.com/2008/09/18/the-top-7-open-source-version-control-systems/>

Nosaltres per a aquesta guia didàctica hem escollit veure una mica més en profunditat el GIT:

- <http://git-scm.com/>

Els motius són que actualment és gairebé el més usat per a produir codi open source, a través d'un servei web que ofereix hospedatge i altres serveis socials. El github

- <http://github.com/>

Però anem a introduir primer el Git i seguirem explicant la plataforma de codi social github.

## .3 Git CVS

De tots els tutorials que existeixen sobre git, un dels més didàctics és el següent:

- <http://www-cs-students.stanford.edu/~blynn/gitmagic/intl/es/>

Us proposo Feu una lectura ràpida per entendre el funcionament d'aquest CVS i continueu amb el següent tema de github. Un cop tingueu un projecte per fer proves amb el github, torneu al manual per seguir aquells punts que vulgueu aprendre amb més profunditat.

Per a qui li vagi millor, també existeix una guia de referència sobre git:

- <http://gitref.org/http://gitref.org>
- <http://byte.kde.org/~zrusin/git/git-cheat-sheet-medium.png>

## .4 Github

Primer veurem una introducció easy al gitub:

- <http://net.tutsplus.com/tutorials/other/getting-the-hang-of-github/>

En aquesta introducció ens explica en què consisteix el github, alguns projectes famosos que l'usen i algunes funcionalitats que té.

Per a poder treballar amb github ens em de registrar, donar-dos d'alta i registrar un repositori amb una clau privada ssh . Podem veure tot això en aquest tutorial:

- <http://help.github.com/set-up-git-redirect/>
- <http://help.github.com/create-a-repo/>

I a partir d'aquest moment ja podem treballar amb un repositori de github. Us recomano baixar-vos algun dels clients que hi hagi pel vostre sistema:

- <http://help.github.com/win-set-up-git/>
- <http://help.github.com/mac-set-up-git/>
- <http://help.github.com/linux-set-up-git/>

## .5 Git en la programació àgil

Git és un dels millors CVS per a la programació àgil degut a la facilitat per crear branques. Us recomano llegir el següent article on proposa una estructura de branques temporal i indefinides per als nostres projectes:

<http://nvie.com/posts/a-successful-git-branching-model/>

Penseu que nosaltres fem un desenvolupament incremental. Això vols dir que si seguim aquest model:

- Si fem SCRUM, cada cop que finalitzem un sprint, llavors la demo hauria d'anar a la branca de master i hauria de ser la pròpia branca de master.
- Si fem SCRUM, cada sprint hauria de ser desenvolupat en la branca de develop.
- Si fem SCRUM cada historia d'usuari hauria d'anar a una branca temporal de "feature"
- Si fem KANBAN, cada cop que finalitzem una història d'usuari (això vol dir que funciona correctament) hauria d'anar a la branca de master
- Si fem KANBAN, sempre desenvoluparem en la branca de develop.
- Si fem KANBAN, les històries d'usuari les desenvoluparem en la branca de "feature".

A més a més, és important dir que s'han de pujar en el projecte tan els codis com totes les eines per automatitzar els tests. Així per exemple, si usem jazmine o ant, els scripts de test han de quedar dins el projecte que pugem al github. Fixeu-vos en aquest projecte:



- <https://github.com/jquery/jquery-mobile>

Com podeu observar conté un directori de test, on s'hi afegixen tots els test que es desenvolupen amb la metodologia TDD.