

---

# TFC - Technology Enhanced Learning - UOC Algorithms

Autor: **Albert Mata**

Enginyeria Tècnica en Informàtica de Gestió

Consultor: **Manel Zaera**

10 de gener de 2012

# Agraïments

---

En poques àrees del coneixement humà es produeix un intercanvi d'informació de manera lliure, gratuïta i desinteressada tan important com succeeix en el món de la informàtica. De qualsevol llenguatge de programació, arquitectura de disseny o metodologia se'n poden trobar centenars d'articles, vídeos, arxius d'àudio o exemples de codi font. Existeixen, a més, multitud de repositoris públics de programari lliure que poden servir tant d'inspiració com d'aprenentatge.

És impossible concebre un estudiant d'informàtica, en qualsevol de les seves múltiples vessants, que no participi activament o passiva de totes aquestes fonts de coneixement. Particularment mai podria haver arribat on sóc ara, ni on seré en el futur, sense aquesta munió de gent entusiasta i desinteressada que ha posat al meu abast tot allò que he necessitat per entendre des dels conceptes més bàsics fins a les tècniques més complexes.

El meu únic agraïment, doncs, no pot ser altre que cap a aquesta comunitat anònima, formada per moltes persones arreu del món que, sense demanar mai res a canvi, són un suport constant en l'intent de ser cada dia un millor professional.

## 1. Resum

---

El projecte UOC Algorithms que es vol desenvolupar en aquest treball de fi de carrera (TFC) és una aplicació pel sistema operatiu iOS, que és el que utilitzen els dispositius mòbils d'Apple. En concret funcionarà especialment bé en els dispositius iPad, tot i que en futures versions és possible que també es procuri desenvolupar una versió optimitzada per aparells iPhone i iPod Touch.

Aquest projecte cau dins de l'àrea de Technology Enhanced Learning, que és l'àrea del TFC en què es treballen les diferents maneres com les noves tecnologies poden millorar els processos d'aprenentatge. UOC Algorithms forma part d'aquesta àrea perquè pretén ser una aplicació que, aprofitant les fantàstiques característiques visuals i tàctils dels aparells iPad, mostri tota una sèrie de continguts didàctics relacionats amb les diferents branques de la informàtica.

Els tipus de continguts que UOC Algorithms pot incloure són molt variats. Poden haver-hi, per exemple, algorismes de tota mena, amb la seva explicació, codi font i anàlisi de costos. També hi tenen cabuda explicacions teòriques i pràctiques de diferents patrons de disseny, arquitectures de software o tècniques de desenvolupament de programari. En definitiva els continguts seran heterogenis en la seva naturalesa, per bé que sempre dins del món de la informàtica, però es buscarà que siguin homogenis en la seva presentació, per tal de resultar el més didàctics possible.

Finalment cal destacar que UOC Algorithms serà totalment lliure, tant pel que fa al propi codi font de l'aplicació com, sobretot, pel que fa als continguts didàctics que inclogui. D'aquesta manera l'usuari podrà fer-ne completament l'ús que en desitgi.

## 2. Taula de continguts

---

### 1. Resum 3

---

### 2. Taula de continguts 4

---

### 3. Taula de figures 6

---

### 4. Desenvolupament de l'aplicació UOC Algorithms 8

---

#### 4.1. Capítol 1: Introducció 8

- 4.1.1. Justificació i context 8
- 4.1.2. Objectius 9
- 4.1.3. Enfocament i mètode seguit 10
- 4.1.4. Planificació del projecte 12
- 4.1.5. Productes obtinguts 18
- 4.1.6. Breu descripció de la resta de capítols d'aquest document 19

#### 4.2. Capítol 2: Anàlisi 19

- 4.2.1. Diagrama de casos d'ús 20
- 4.2.2. Consideracions sobre llicències privatives i lliures 22
- 4.2.3. Estructuració dels continguts 24
- 4.2.4. Consideració inicial sobre la interfície gràfica 25

#### 4.3. Capítol 3: Disseny 26

- 4.3.1. Diagrama de classes model - diagrama ER 26
- 4.3.2. Detalls de la persistència dels continguts en base de dades 28
- 4.3.3. Utilització de formes normals (FN) a la base de dades 31
- 4.3.4. Utilització del model MVC 32
- 4.3.5. Selecció dels continguts per la primera versió 35

#### 4.4. Capítol 4: Implementació 36

- 4.4.1. Eines utilitzades 36
- 4.4.2. Detalls tècnics de l'estructura dels continguts 36
- 4.4.3. Introducció al framework Cocoa Touch 37
- 4.4.4. Format i ús de marques per la creació de continguts 39

## Apartat 2. Taula de continguts

4.4.5. Incorporació dels continguts a la base de dades	42
4.4.6. Presentació del funcionament de l'aplicació UOC Algorithms	44
4.5. Capítol 5: Documentació	51
4.6. Capítol 6: Consideracions econòmiques	53
4.7. Capítol 7: Conclusions	54
4.7.1. Línies de treball futures	56

## **5. Glossari 58**

---

## **6. Bibliografia 61**

---

## **7. Annexos 63**

---

7.1. Annex 1: Llicència d'ús de la llibreria Libupskirt 1.1	63
7.2. Annex 2: Llicència d'ús del programari Markdown	63
7.3. Annex 3: Llicència d'ús de la font programari Yanone Kaffeesatz	64

## 3. Taula de figures

---

### 4. Desenvolupament de l'aplicació UOC Algorithms 8

---

#### 4.1. Capítol 1: Introducció 8

Figura 4.1.1. Diagrama de Gantt: fases principals del projecte (en setmanes)	13
Figura 4.1.2. Diagrama de Gantt: fase de planificació (en dies)	14
Figura 4.1.3. Diagrama de Gantt: fase d'anàlisi i disseny (en setmanes)	15
Figura 4.1.4. Diagrama de Gantt: fase d'implementació (en setmanes)	16
Figura 4.1.5. Diagrama de Gantt: fase de publicació (en setmanes)	17

#### 4.2. Capítol 2: Anàlisi 19

Figura 4.2.1. Diagrama de casos d'ús	21
Figura 4.2.2. Aplicació Mail oficial d'Apple per dispositiu iPad	26

#### 4.3. Capítol 3: Disseny 26

Figura 4.3.1. Diagrama de classes model amb Core Data	28
Figura 4.3.2. Taula AMArea a Core Data	29
Figura 4.3.3. Taula AMAuthor a Core Data	29
Figura 4.3.4. Taula AMCategory a Core Data	30
Figura 4.3.5. Taula AMContent a Core Data	30
Figura 4.3.6. Esquema del patró de disseny Model - Vista - Controlador	33
Figura 4.3.7. Xcode gestionant grups separats per les classes MVC	34

#### 4.4. Capítol 4: Implementació 36

Figura 4.4.1. UOC Algorithms amb l'iPad en posició horitzontal	38
Figura 4.4.2. UOC Algorithms amb l'iPad en posició vertical	39
Figura 4.4.3. Exemple de text utilitzant el format Markdown	40
Figura 4.4.4. Flux de treball amb arxius en format Markdown	41
Figura 4.4.5. Taula AMAuthor a SQLite	42
Figura 4.4.6. Contingut de l'arxiu de text per la taula AMAuthor	43
Figura 4.4.7. UOC Algorithms en el moment d'iniciar l'aplicació	45
Figura 4.4.8. UOC Algorithms quan s'ha seleccionat un contingut	46
Figura 4.4.9. UOC Algorithms amb l'opció de cerca activada	47
Figura 4.4.10. UOC Algorithms mostrant els crèdits de l'aplicació	48
Figura 4.4.11. UOC Algorithms mostrant les accions auxiliars	49
Figura 4.4.12. UOC Algorithms amb l'iPad en posició vertical	50
Figura 4.4.13. Pàgina web de l'aplicació UOC Algorithms	51

4.5. Capítol 5: Documentació 51

Figura 4.5.1. Exemple d'utilització d'etiquetes # pragma mark en Xcode 52

Figura 4.5.2. Exemple de mètode documentat 53

## 4. Desenvolupament de l'aplicació UOC Algorithms

---

Com succeeix en tot procés de desenvolupament de programari, per dur a terme l'aplicació UOC Algorithms ha estat necessari passar per tot un seguit de fases que inclouen aspectes com els motius pels quals es seleccioni un projecte i no un altre, els objectius que es persegueixen i la manera de treballar amb la qual es volen assolir, una planificació inicial del projecte, detalls tècnics de tota mena que van sorgint al llarg de l'execució del treball, i en darrer lloc tota una sèrie de valoracions finals. Són tots aquests elements els que, de manera ordenada, es presenten al llarg d'aquest apartat.

### 4.1. Capítol 1: Introducció

En aquest primer capítol introductori s'analitzen totes les qüestions relatives al projecte que no suposen entrar en detalls tècnics però que són igualment importants: fonamentalment aspectes relacionats amb les motivacions i la planificació del mateix.

#### 4.1.1. Justificació i context

---

Després d'haver cursat la pràctica totalitat dels estudis d'Enginyeria Tècnica en Informàtica de Gestió a la UOC, s'ha detectat que, malauradament, en moltes assignatures hi ha mancança de recursos pedagògics que facilitin l'aprenentatge en els materials docents. Tot sovint, l'alumne ha de recórrer a materials publicats per altres universitats, tant espanyoles com, sobretot, americanes, per entendre determinats conceptes (per exemple algorismes d'ordenació, algorismes de grafs, determinats conceptes d'UML, tècniques de programació, etc.). Com a resultat, s'acaba amb un conjunt de recursos molt interessants però, al mateix temps, molt dispersos: un article d'un lloc, una presentació en PowerPoint d'una altra banda, un vídeo de determinada universitat...

Al mateix temps es considera que la UOC, atesa la seva condició d'universitat virtual, hauria de ser qui disposés dels millors materials didàctics per facilitar al màxim el procés d'aprenentatge dels conceptes presentats a cada assignatura. Els estudiants que formen part de la UOC habitualment compaginen els seus estudis amb altres activitats professionals i, per tant, disposen de menys temps per assimilar els continguts. Així, els



suposa certament un inconvenient que a banda de l'estudi dels mòduls docents hagin d'invertir temps cercant explicacions complementàries a la Xarxa.

És per aquest motiu que es planteja desenvolupar una eina de tipus educatiu que doni solució a aquesta problemàtica, facilitant a l'estudiant l'aprenentatge de determinats conceptes, concretament dins de la branca informàtica.

### **4.1.2. Objectius**

---

El projecte que es proposa, per tant, consisteix en la creació d'una aplicació de tipus repositori on els alumnes d'estudis d'Enginyeria Informàtica -en qualsevol de les seves vessants- puguin cercar-hi algorismes, tècniques i conceptes vistos al llarg dels itineraris formatius. És, per tant, una aplicació que pot resultar d'utilitat no tan sols als alumnes de la UOC sinó també als de qualsevol altra universitat del món.

Aquest repositori no actuarà simplement ajuntant recursos que es poden trobar a la Xarxa, sinó que vetllarà en tot moment perquè mantinguin uns formats homogenis i s'estructurin de manera organitzada. Es vol aconseguir una eina on sigui fàcil trobar-hi exemples de codi font, representacions gràfiques i explicacions de tota mena. En cap cas es tracta de crear materials que substituïen els mòduls docents de la UOC, sinó més aviat complements per aquests mòduls en determinats punts.

De tots els formats que aquesta aplicació podria prendre i de totes les plataformes sobre les que podria funcionar, s'escull desenvolupar-la pels coneguts dispositius iPad de la companyia Apple. Aquesta decisió es pren atenent les excel·lents prestacions que un iPad ofereix per mostrar informació i com a dispositiu de consulta.

Abans d'emprendre aquest projecte, s'ha dut a terme una recerca exhaustiva per comprovar si existeix ja una eina de característiques similars. El resultat ha estat negatiu: tot i que hi ha una enorme quantitat de recursos disponibles a la Xarxa en tots els formats, no s'ha trobat cap aplicació específica per dispositius iPad que actuï com a repositori de continguts docents en l'àrea informàtica. És per aquest motiu que es pensa que aquesta aplicació podria no només resultar d'utilitat pels alumnes de la UOC, sinó també tenir una bona acceptació per part de la comunitat informàtica en general.

El projecte consistirà, així, en el desenvolupament d'una aplicació educativa per dispositius iPad que inclogui un repositori de continguts intern. L'abast d'aquest projecte pel que fa al TFC inclourà el desenvolupament del marc general de l'aplicació, el sistema de persistència en base de dades i un subconjunt petit de continguts docents per mostrar la capacitat de l'aplicació. Complementat, és clar, amb la documentació i els diagrames

UML bàsics que siguin necessaris. A banda, si la planificació ho permet s'afegiran també funcionalitats extres com un sistema de recerca de continguts o un altre per canalitzar la petició de nous continguts.

Queden fora de l'abast altres qüestions com la internacionalització de l'aplicació, la possibilitat de fer-la multiplataforma i, en general, tot allò que no està explícitament indicat en aquest document. Aquests punts queden pendents per implementar en futures versions de l'aplicació.

Es vol prestar una atenció molt especial als formats visuals dels continguts. Els usuaris de dispositius iPad són especialment sensibles a la presentació de les aplicacions i, sovint, desestimen aquelles que no mostren una interfície gràfica ben cuidada. És per això que en cap cas es persegueix fer una recopilació ràpida de continguts disponibles a altres llocs de la Xarxa, sinó que es vol crear un marc on s'hi mostrin continguts treballats a mida, o convenientment adaptats, per tal de garantir en tot moment aquest especial detall per l'aspecte visual.

Per tal d'assolir l'objectiu principal de desenvolupar aquest producte final, es contemplen altres objectius de caire personal que també caldrà satisfer per dur a terme el projecte amb èxit:

- Analitzar convenientment les funcionalitats de les que ha de disposar l'aplicació.
- Planificar adequadament tot el treball que caldrà dur a terme en el termini del que es disposa per tirar endavant el TFC.
- Investigar i aprendre tot el que sigui necessari sobre les tecnologies de desenvolupament d'aplicacions per iPad, incloent-hi un nou llenguatge de programació, nous patrons de disseny i nous frameworks (com Cocoa Touch o Core Data).
- Generar un codi net, eficient, llegible i suficientment documentat per tal que el projecte sigui fàcil de mantenir en el futur i agradable de consultar per qualsevol col·laborador que pugui sorgir.
- Saber escoltar les propostes que es rebin, especialment per part del consultor, i analitzar-les a fons abans de decidir la conveniència o no d'aplicar-les al projecte.

### **4.1.3. Enfocament i mètode seguit**

---

Per afrontar aquest projecte amb garanties ha estat necessari prendre en consideració tres grans aspectes: les tecnologies a emprar, els temes relatius a llicències i la gestió dels continguts didàctics. De tots tres se'n parla més extensament en apartats posteriors

d'aquest mateix document, però tot seguit s'exposen algunes idees principals vinculades amb la metodologia de treball seguida en aquest TFC.

En primer lloc, pel que fa a les tecnologies a utilitzar ha calgut decidir primer amb quines es volia treballar per poder-hi aprofundir després. Quan es vol desenvolupar una aplicació per dispositius iPad (i, en general, per dispositius mòbils de la companyia Apple) hi ha tres grans alternatives en relació a les eines disponibles per fer-ho:

- La primera és la que Apple recomana. Consisteix en la utilització de l'entorn de desenvolupament Xcode en una màquina Mac i en l'ús d'Objective-C com a llenguatge de programació. Com a framework principal es fa servir Cocoa Touch, que és la base de qualsevol aplicació nativa per iPad. A partir d'aquí hi ha altres frameworks auxiliars disponibles per altres tasques, com per exemple Core Data per la gestió de la base de dades.

Aquesta tria és, sense dubte, la més interessant si es vol aconseguir una aplicació totalment nativa per la plataforma seleccionada i si es vol treure el màxim rendiment del dispositiu iPad. Per contra, Objective-C és un llenguatge de programació amb una corba d'aprenentatge alta: tot i ser orientat a objectes de la mateixa manera que altres llenguatges coneguts i estudiats al llarg dels estudis d'Enginyeria Tècnica en Informàtica de Gestió, com per exemple Java, té moltes diferències tant a nivell de sintaxi com a nivell dels patrons de disseny que s'hi fan servir si es fa ús del framework Cocoa Touch. Per tant l'estudi i domini d'aquestes noves tecnologies suposen un afegit important de dificultat de cara al projecte.

- La segona consisteix en desenvolupar una aplicació web que es pugui visualitzar perfectament amb el navegador del dispositiu iPad (Safari per dispositius mòbils). Aquesta tria és òptima pel que fa a dificultat i aprenentatge, atès que utilitza únicament eines com HTML5, CSS3 i JavaScript, d'ús prou habitual com per no suposar cap problema. Per contra, les aplicacions que es poden fer amb aquestes tecnologies estan molt limitades pel que fa a l'ús dels recursos propis d'un iPad, ja que l'aplicació s'executa -de fet, es visualitza- dins del navegador web i no pot sortir d'aquest marc.
- La darrera fa referència a l'ús de tecnologies, com ara PhoneGap o Appcelerator, que permeten desenvolupar aplicacions per dispositius mòbils de qualsevol plataforma utilitzant llenguatges de programació ja coneguts, principalment HTML5 i JavaScript. Amb aquestes eines no es genera una aplicació de tipus web, sinó que al final s'obté una aplicació quasi nativa instal·lable en un dispositiu iPad. Presenta l'avantatge evident de ser més fàcil de desenvolupar que utilitzant les eines pròpies d'Apple presentades a la primera opció, però l'inconvenient que, al

generar aplicacions que necessiten una capa intermèdia –la que proveeix per exemple PhoneGap- el rendiment sempre és lleugerament inferior.

Prenent en consideració tots aquests punts, la decisió ha estat fer servir les eines pròpies d'Apple. S'assumeix que la corba d'aprenentatge esmentada fa necessària una dedicació suplementària, però es considera que la contrapartida és suficientment interessant, ja que s'obté un producte totalment optimitzat per dispositius iPad. I, al mateix temps, es considera que el desenvolupament del TFC és un moment idoni per aprendre i dominar un nou llenguatge de programació i una nova tecnologia.

En segon lloc, pel que fa als temes relatius a llicències comercials, al llarg del projecte s'ha fet un canvi de plantejament important. Inicialment s'havia concebut aquesta aplicació de manera tancada: el propi codi font de l'aplicació havia de ser privat i els continguts didàctics mostrats també podien tenir llicències privatives.

No obstant, en base a les recomanacions del consultor d'aquest projecte, Manel Zaera, i després d'haver analitzat profundament la situació, la viabilitat econòmica de l'aplicació resultant i les diferents alternatives, s'ha decidit plantejar un projecte lliure, tant pel que fa el codi font, que serà obert, com, sobretot, pel que fa als continguts didàctics, que també tindran llicències lliures per facilitar tant la incorporació de recursos a l'aplicació com l'ús que qualsevol usuari vulgui fer-ne.

Finalment, la gestió dels continguts didàctics cal entendre-la de manera descentralitzada. Aquest TFC no pretén desenvolupar continguts extensos i amplis, sinó únicament generar el marc en el qual s'hi puguin crear continguts de manera fàcil per tal que el repositori de coneixement vagi creixent. Es vol, doncs, facilitar la creació i incorporació de continguts, cosa que amb l'enfocament cap a les llicències lliures comentat anteriorment s'aconsegueix amb relativa facilitat.

#### **4.1.4. Planificació del projecte**

---

El projecte UOC Algorithms s'ha planificat agafant en consideració les fases habituals del cicle de vida del programari i procurant que, d'alguna manera, coincidissin amb les entregues parcials previstes per l'assignatura a què aquest TFC correspon.

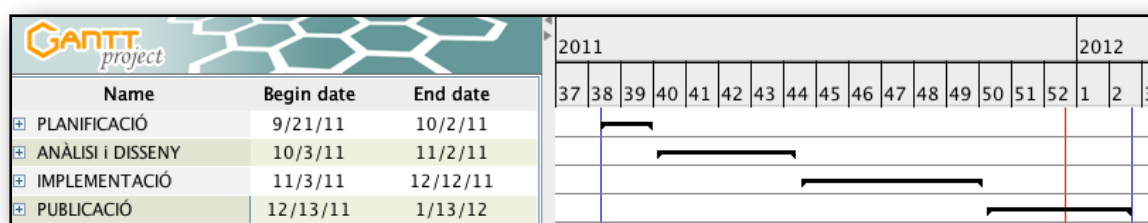
Així, s'ha dut a terme una divisió en quatre fases principals (planificació, anàlisi i disseny, implementació i publicació) i s'han identificat les principals tasques a desenvolupar en cada una d'aquestes fases, assignant al mateix temps les durades estimades per a cada tasca.

No obstant, aquesta planificació ha estat subjecta a diversos errors tant relacionats amb el factor humà, com amb el procés, el propi producte i la tecnologia emprada. En alguns casos, fins i tot, la planificació queda fora del control de qui desenvolupa aquest projecte, amb la qual cosa hi ha tasques que deliberadament s'han deixat fora de la planificació. És el cas, per exemple, de la publicació de l'aplicació final a l'AppStore<sup>1</sup> d'Apple. Aquest procés es pot estimar amb una durada d'un parell de setmanes però, atès que depèn exclusivament del departament corresponent d'Apple, tant podria ser que quedés reduït a només uns dies com que s'allargués diverses setmanes si els revisors d'Apple consideren que cal fer modificacions a l'aplicació. Per tant, s'estableix que el producte final fruit d'aquest projecte no serà una aplicació que es pugui descarregar des de la botiga oficial d'Apple, sinó únicament una versió executable en el simulador o bé en un dispositiu iPad al qual es tingui accés físic directe.

Pel que fa a la resta d'errors de planificació, els que sí que són controlables, s'ha intentat que tinguessin la menor repercussió possible en la temporització del projecte. Per aconseguir-ho, a banda d'haver realitzat les estimacions temporals amb la màxima cura, s'han establert petits períodes lliures al llarg de la planificació del projecte<sup>2</sup>. En aquests períodes no hi ha cap tasca assignada: hi són únicament per tal d'intentar recuperar el ritme previst en la planificació quan es produeix un endarreriment, i també per tenir un marge de seguretat per tasques de depuració i refactorització de codi.

La distribució temporal de tot el projecte en les quatre fases esmentades es pot veure de manera molt resumida en el següent diagrama de Gantt:

**Figura 4.1.1. Diagrama de Gantt: fases principals del projecte (en setmanes).**



Tot seguit s'amplien els detalls de cadascuna de les quatre fases, incloent una breu descripció de les tasques incloses, que s'ha procurat que complissin les característiques

<sup>1</sup> L'AppStore és la botiga oficial d'aplicacions per dispositius iPhone, iPod i iPad d'Apple. Es tracta de l'únic mitjà oficial per comercialitzar i distribuir aquestes aplicacions.

<sup>2</sup> Aquests períodes lliures es reflecteixen als diagrames de Gantt sota el concepte 'Tasca de reprogramació' i tenen les seves barres dibuixades de color verd clar.

SMART<sup>3</sup>, és a dir, que fossin tasques concretes, quantificables, assolibles, rellevants pel projecte i fàcils de comprovar-ne el grau d'assoliment.

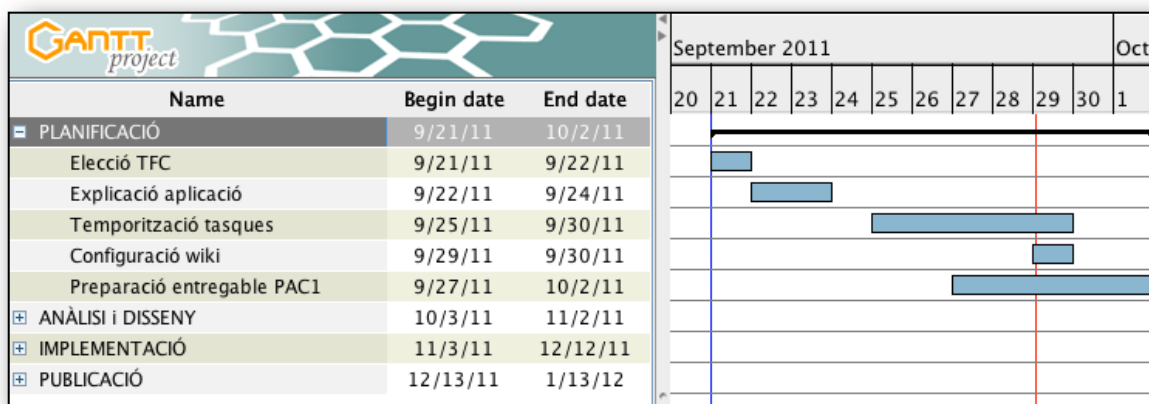
## Fase de planificació

Aquesta fase correspon a la part prèvia al desenvolupament del projecte. Aquí s'escull el projecte concret que es vol dur a terme i es fa la planificació de tasques i durades per la resta del projecte. Les principals tasques identificades són les següents:

- Elecció del projecte concret a realitzar com a TFC.
- Explicació de l'aplicació: problemàtica que es vol resoldre i proposta concreta de projecte que es fa, establint què estarà inclòs i què no.
- Temporització de les tasques: assignació de durades estimades per a cada tasca.
- Configuració d'un sistema wiki<sup>4</sup> per dur la documentació interna.

Tot això queda recollit en el següent diagrama de Gantt:

**Figura 4.1.2. Diagrama de Gantt: fase de planificació (en dies).**



## Fase d'anàlisi i disseny

Aquesta fase correspon a la part en què es prenen les principals decisions relatives al disseny de l'aplicació i es comencen a implementar. Les principals tasques identificades són aquestes:

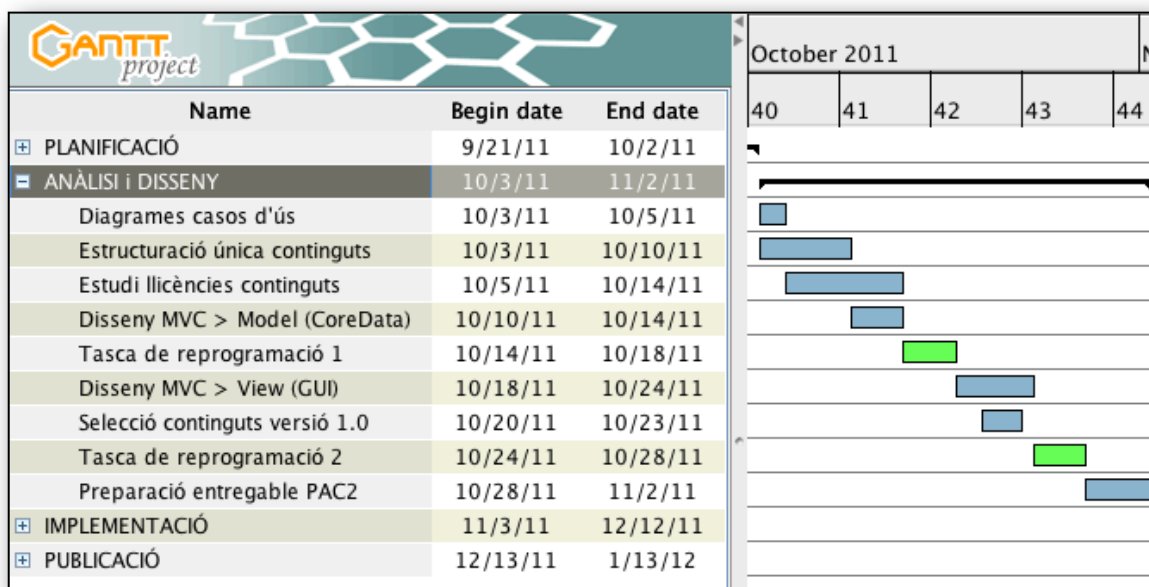
<sup>3</sup> SMART és un acrònim anglès pels adjectius 'specific', 'measurable', 'achievable', 'relevant' i 'trackable'.

<sup>4</sup> Un sistema wiki consisteix en un lloc web fàcilment editable per l'autor o autors a través d'un navegador web en què, utilitzant una sintaxi determinada, és fàcil donar format als continguts.

- Creació de diagrames de casos d'ús.
- Definició de l'estructura dels continguts.
- Estudi de les llicències relatives als continguts: decisió sobre si adoptar llicències lliures o privatives.
- Disseny del component model dins del patró de disseny MVC: inclou el disseny de la persistència utilitzant la tecnologia Core Data.
- Disseny de les vistes dins del patró de disseny MVC: bàsicament la interfície gràfica de l'aplicació.
- Selecció dels continguts per la primera versió: tria d'una petita mostra de continguts.

Aquestes tasques queden recollides en el següent diagrama de Gantt:

**Figura 4.1.3. Diagrama de Gantt: fase d'anàlisi i disseny (en setmanes).**



## Fase d'implementació

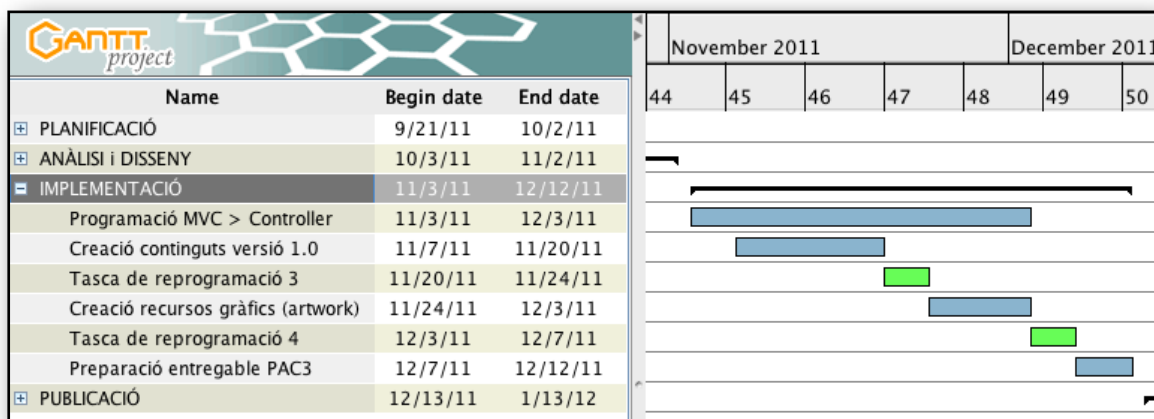
Aquesta fase correspon a la part en què s'implementarà la lògica principal de l'aplicació i es crearan els continguts que han de formar part de la primera versió. Les principals tasques identificades són aquestes:

- Disseny del controlador dins del patró de disseny MVC: comunicació entre model i vistes.

- Creació de recursos gràfics: cerca de recursos gràfics amb llicències lliures i creació des de zero d'altres imatges a utilitzar en l'aplicació.
- Creació dels continguts de la primera versió.
- Integració de model, vistes i controladors en el patró de disseny MVC per aconseguir una primera versió alfa funcional de l'aplicació UOC Algorithms.

Aquesta divisió del treball es reflecteix en el següent diagrama de Gantt:

**Figura 4.1.4. Diagrama de Gantt: fase d'implementació (en setmanes).**



## Fase de publicació

Aquesta fase dona per finalitzat el projecte. Es tracta, potser, de la fase més important, ja que és en aquest moment que cal recollir tot el treball desenvolupat al llarg del projecte i crear un producte únic, lligat, que doni imatge de completesa i que s'apropi tant com sigui possible a un projecte professional. És el moment d'acabar d'ajustar els detalls que hi pugui haver pendents i de crear la memòria final i la presentació que cal entregar, prenent com a base tota la documentació que s'ha anat generant al llarg de tot el TFC. Així, les tasques identificades són les següents:

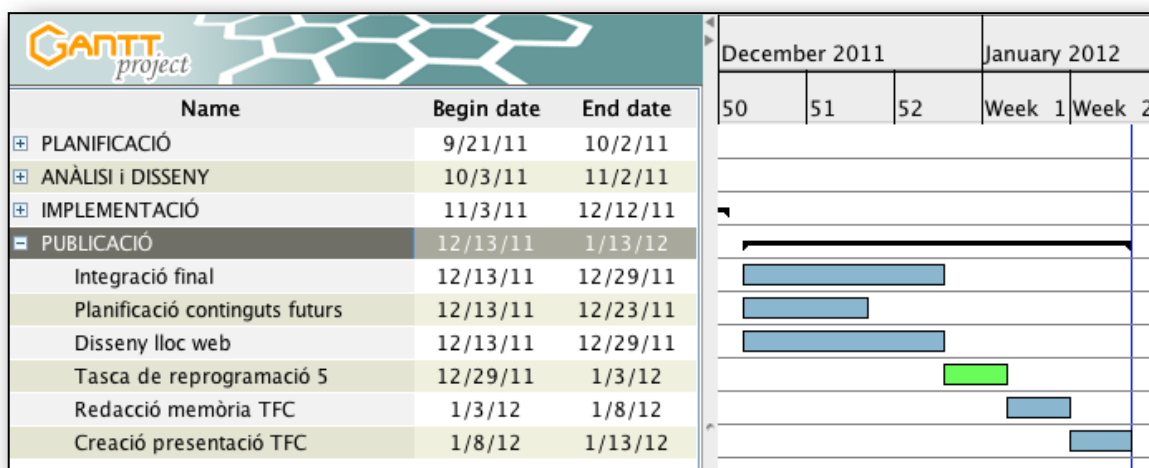
- Integració final de l'aplicació: realització de tests i correcció d'errors.
- Planificació de continguts futurs.
- Disseny d'un lloc web senzill per l'aplicació: es tracta d'un requeriment d'Apple per totes les aplicacions que es publiquen a la botiga oficial.
- Redacció de la memòria del TFC.



- Creació de la presentació del TFC: síntesi de les idees més importants que es volen transmetre, motiu pel qual es farà servir la regla de Guy Kawasaki<sup>5</sup> convenientment adaptada.

Tots aquests punts queden recollits en el següent diagrama de Gantt:

**Figura 4.1.5. Diagrama de Gantt: fase de publicació (en setmanes).**



## Disparadors de cost

Entre les tasques indicades en els apartats anteriors n'hi ha dues que actuen com a disparadors de cost, ja que per la seva naturalesa poden fer que endarreriments en aquestes activitats tinguin efectes negatius importants en la temporització del projecte. Es tracta de les següents tasques:

- Disseny del model del patró de disseny MVC.
- Disseny dels controladors del patró de disseny MVC.

El disseny del model és crític perquè inclou tota la part relativa a la persistència dels continguts en base de dades, que és un aspecte totalment específic del disseny d'aplicacions per iPad, atès que fa servir un framework propi d'Apple (Core Data). Per tal d'intentar minimitzar el risc d'aquesta tasca, durant la primera fase del projecte ja es duu

<sup>5</sup> Guy Kawasaki és un conegut emprenedor en el sector de les noves tecnologies que, sovint, defensa un model minimalista en l'àmbit de les presentacions de negocis. En el seu lloc web explica la que, per ell, és la clau d'una bona presentació: la regla 10-20-30, que pren el nom de 10 diapositives a mostrar, 20 minuts de durada i 30 com a mida mínima de font a emprar. En la bibliografia d'aquest document es pot trobar la referència sencera a l'article on exposa aquest concepte.

a terme un estudi teòric a fons d'aquesta tecnologia. A més, just després d'aquesta tasca se'n programa una de reprogramació per tal de tenir un marge de seguretat adient.

El disseny dels controladors també és crític perquè en una arquitectura amb patró de disseny MVC el controlador actua, d'alguna manera, com a lligam entre tots els elements de l'aplicació. Per tant aquí és on hi ha més lògica a programar i on el codi font és més sensible a l'aparició d'errors.

#### 4.1.5. Productes obtinguts

---

L'únic producte generat en aquest TFC, a banda d'aquest mateix document de memòria i d'una presentació amb diapositives, és la pròpia aplicació UOC Algorithms. Consisteix en una aplicació senzilla que posa a l'abast de l'usuari tot un conjunt de recursos didàctics, relacionats sempre amb l'enginyeria informàtica, que anirà creixent amb les posteriors versions i actualitzacions. No es pretén crear un producte que destaquï per una espectacular interfície gràfica, sinó més aviat el contrari: com més desapercebut passi tot el marc de l'aplicació i més es permeti que l'usuari es pugui centrar totalment en els continguts, major serà l'èxit assolit amb aquest projecte.

Al tractar-se d'una aplicació específica per dispositius iPad no és possible entregar un arxiu executable de tal manera que es pugui provar el seu funcionament en qualsevol ordinador. El que s'entregarà, doncs, serà el codi font en llenguatge de programació Objective-C. Si es vol provar el seu funcionament serà necessari disposar d'un ordinador Mac amb l'entorn de desenvolupament Xcode instal·lat (en la seva versió 4.2 o posterior). Amb aquesta configuració només caldrà obrir el projecte i executar-lo en un simulador de dispositiu iPad.

Com que això no sempre serà possible, en el **Capítol 4: Implementació** es mostren un conjunt de captures de pantalla que permeten visualitzar quin serà el resultat final i entendre com hi interactuarà l'usuari.

Alternativament, l'aplicació també estarà disponible a la botiga oficial d'aplicacions d'Apple (l'AppStore) per tal que tothom qui ho vulgui se la pugui descarregar en un dispositiu iPad real i provar-la amb interacció física amb l'aparell. No obstant, la publicació d'una aplicació en aquest repositori oficial no és immediata, sinó que ha de passar el procés d'aprovació per part dels tècnics revisors d'Apple. Aquest procés no té un temps fix establert, de manera que no és possible saber en quin moment estarà disponible la descàrrega de la primera versió de UOC Algorithms. Per tant, el producte obtingut en aquest TFC inclourà només el codi font executable en un simulador, però en cap cas l'aplicació disponible a l'AppStore.

#### **4.1.6. Breu descripció de la resta de capítols d'aquest document**

---

En els propers capítols s'exposen les diferents fases per les quals passa el desenvolupament de l'aplicació UOC Algorithms i algunes consideracions complementàries que cal tenir en compte. De manera breu, es tracten els següents aspectes:

- Capítol 2: Anàlisi. S'analitzen les funcionalitats que ofereix l'aplicació i es reflecteixen els casos d'ús més habituals. També es tenen en consideració qüestions relatives a les llicències que tindrà l'aplicació i a l'estructura dels continguts didàctics que s'hi volen incloure.
- Capítol 3: Disseny. Es mostren diagrames UML que ajuden a veure la manera com està dissenyada l'aplicació. De la mateixa manera, s'adjunten diagrames ER i detall de com es defineix la persistència dels continguts en la base de dades. Finalment es parla àmpliament del patró de disseny MVC, sense cap mena de dubte el més popular quan es tracta de desenvolupament d'aplicacions en plataformes Apple.
- Capítol 4: Implementació. Es tracta d'un capítol que inclou una primera part eminentment tècnica, amb algunes qüestions relatives a la codificació del projecte que poden resultar rellevants, i una segona part més funcional on es mostren algunes captures de pantalla de l'aplicació per mostrar-ne el funcionament.
- Capítol 5: Documentació. Es parla de com s'ha enfocat la documentació de l'aplicació UOC Algorithms al llarg de tot el seu desenvolupament.
- Capítol 6: Consideracions econòmiques. S'exposen algunes idees relacionades amb la vessant econòmica i comercial del projecte.
- Capítol 7: Conclusions. S'ofereix una versió personal de quines han estat les conclusions que s'han extret al llarg del desenvolupament del projecte i es parla de possibles línies futures d'actuació.

## **4.2. Capítol 2: Anàlisi**

En aquest capítol es duu a terme una anàlisi del projecte tant en l'àmbit funcional, pensant en totes les opcions que es volen oferir a l'usuari, com en un àmbit més general, plantejant quines llicències es volen adoptar i quina estructura visual es desitja que tinguin els continguts que s'inclouen a l'aplicació.

### 4.2.1. Diagrama de casos d'ús

---

Tal com s'ha comentat quan s'han establert els objectius d'aquest projecte, les aplicacions per dispositius iPad es caracteritzen per una utilització senzilla que, per tant, fa que no hi hagi gaires casos d'ús. En aquest sentit és important no confondre un iPad amb un ordinador convencional, ja que presenten diferències pel que fa a la usabilitat<sup>6</sup>.

Els ordinadors, des del punt de vista de la funcionalitat, tenen una doble vessant clara, ja que s'utilitzen tant per la producció de continguts com pel seu consum. Els dispositius tipus tauleta com l'iPad, en canvi, estan clarament orientats al consum de continguts, deixant la producció com una activitat residual que no resulta còmoda per les pròpies característiques d'aquests aparells.

Així, mentre que en una aplicació d'ordinador és habitual que l'usuari tingui disponibles multitud d'opcions per dur a terme moltes tasques diferents, en una aplicació per iPad cal definir molt clarament què és el que es vol oferir a l'usuari i mostrar-li d'una manera tan clara i neta com sigui possible.

És per aquest motiu que una aplicació per iPad ha de tenir una funcionalitat bàsica, senzilla d'entendre i de fer anar i basada en molt pocs casos d'ús diferents. Cal tenir present que les recomanacions d'Apple van en la línia de fer aplicacions que l'usuari entengui com funcionen sense necessitar cap mena d'indicació ni explicació addicional.

Tenint en compte tots aquests factors, els casos d'ús que presenta l'aplicació UOC Algorithms són els que es mostren en la figura 4.2.1 de la pàgina següent.

Sense cap dubte, el cas d'ús principal és el de visualització de continguts educatius. Dins d'aquest n'hi ha dos més que l'estenen i li afegixen funcionalitat: enviar el contingut mostrat per correu i informar d'un error detectat.

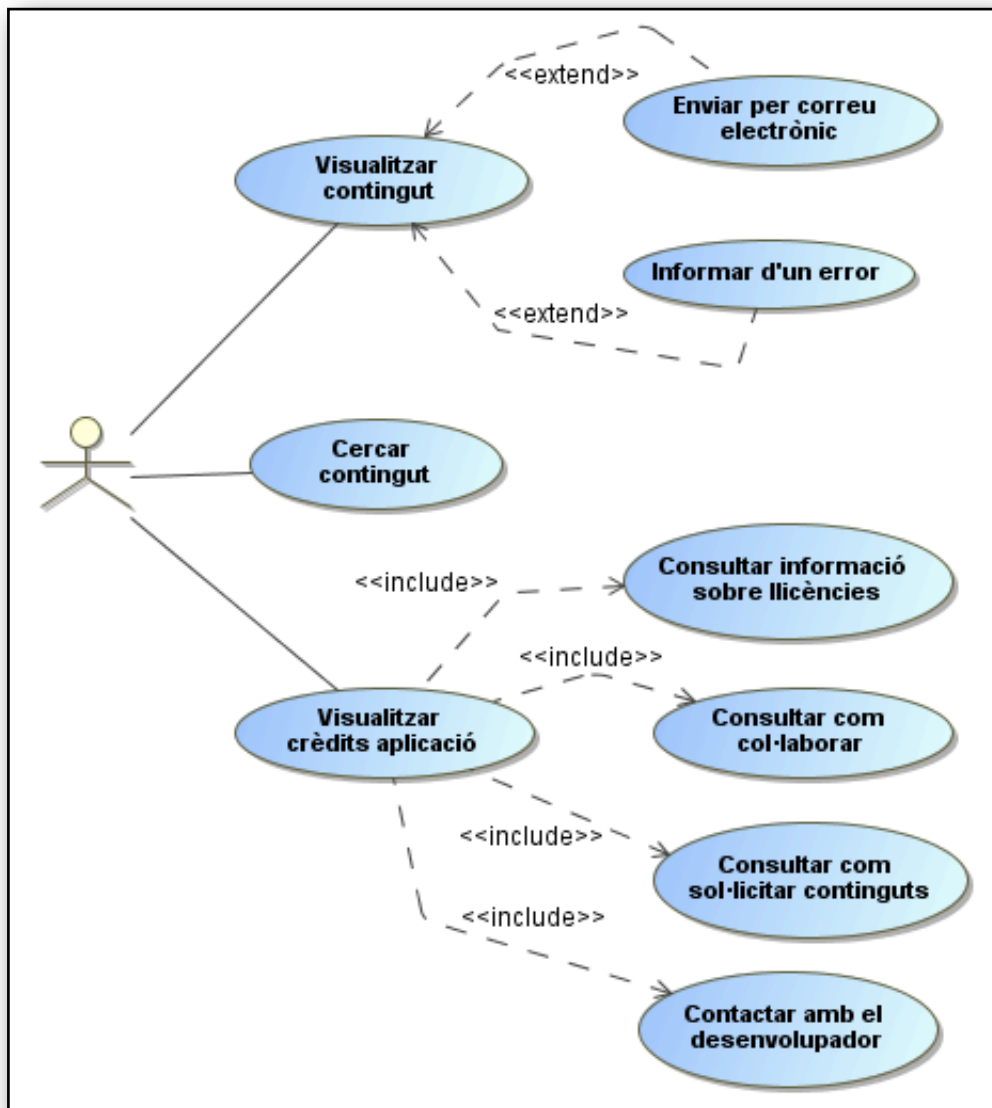
El segon cas d'ús més important és el de cercar continguts. Aquesta funcionalitat permet l'usuari buscar recursos a partir de qualsevol paraula que formi part dels mateixos, per tant és realment flexible. Sense aquesta possibilitat podria resultar complicat trobar el que s'està buscant.

La resta de casos d'ús són tasques purament complementàries a la principal: visualització dels crèdits de l'aplicació i informació sobre llicències, contactar amb l'autor, demanar nous continguts, etc.

---

<sup>6</sup> La usabilitat es pot definir com la facilitat amb què la gent pot usar una eina o un giny per aconseguir un objectiu concret (<http://ca.wikipedia.org/wiki/Usabilitat>).

Figura 4.2.1. Diagrama de casos d'ús.



Cal tenir en compte que el projecte té uns altres casos d'ús relacionats amb la creació de continguts. Aquests altres casos, però, no són pròpiament casos d'ús de l'aplicació UOC Algorithms entesa com l'aplicació que l'usuari té instal·lada en el seu dispositiu iPad. En tot cas es tracta de casos d'ús del propi projecte que van a càrrec del programador i dels creadors de continguts.

Novament ens trobem en aquest punt amb una diferència important en una aplicació per iPad respecte a una aplicació convencional d'escriptori. Mentre que en aquestes darreres és habitual que els usuaris administradors o gestors interactuïn amb la mateixa aplicació que els usuaris, simplement fent ús d'opcions de menú diferents especialment adaptades als seus perfils de creadors i gestors de continguts, en les primeres, en les aplicacions per

iPad, és molt estrany trobar aquest comportament, ja que un iPad no és una bona eina per desenvolupar-hi continguts nous. Per tant, tot i que UOC Algorithms té una vessant de creació de continguts, no es reflexa al diagrama de casos d'ús perquè realment no serà un cas d'ús de la pròpia aplicació, sinó només quelcom que li serà auxiliar.

#### **4.2.2. Consideracions sobre llicències privatives i lliures**

---

Al llarg del desenvolupament de l'aplicació UOC Algorithms s'han considerat diversos tipus de llicències referents a drets d'autor que s'hi podrien aplicar. Per una banda les llicències lliures presenten una sèrie d'avantatges -inclosos els de caire moral-. Per altra banda, la intenció inicial d'explotar comercialment aquesta aplicació feia recomanable considerar altres alternatives més conservadores.

Tenint en compte aquests factors, les llicències que en un primer moment es va considerar convenient adoptar van ser les següents:

- Per la part dels continguts dedicada específicament al codi font en qualsevol llenguatge, llicència lliure de tipus GPL (GNU General Public License). Això d'una banda facilitava que tothom lliurement pogués fer servir qualsevol codi publicat en aquesta aplicació, d'acord amb les conviccions personals i professionals de l'autor del projecte. I d'altra banda evitava que es produïssin col·lisions al llicenciar un codi font que pogués existir prèviament amb llicència lliure sense que se'n tingués coneixement.
- Pel que fa a tota la resta d'apartats dels continguts (explicacions, simulacions de funcionament, anàlisis, etc.), llicència privativa on tots els drets estiguessin reservats per l'autor que hagués elaborat aquests recursos.
- La pròpia aplicació UOC Algorithms, entesa com el codi font en llenguatge Objective-C i tots els recursos complementaris (imatges, documents web, arxius de configuració, etc.), també llicència amb tots els drets reservats, ja que es considerava la millor elecció pensant en la possible comercialització de l'aplicació.

No obstant, aquest plantejament inicial s'ha vist modificat al llarg del desenvolupament del projecte, degut principalment a dos motius.

En primer lloc, el paper del consultor del TFC en aquest punt ha estat fonamental. Des d'un primer moment ha fet esment a la importància que tenen les llicències escollides sobre els continguts docents pensant en la possibilitat que hi hagi col·laboradors que participin en la creació d'aquests recursos didàctics. És evident que el fet de treballar amb

Llicències privatives redueix molt el nombre de recursos que es poden fer servir i pot suposar, al mateix temps, un fre per aconseguir col·laboracions de terceres persones. I un projecte com UOC Algorithms necessita que hi hagi més creadors de continguts que únicament l'autor inicial si realment es pretén que s'arribi a convertir en un repositori de consideració.

En segon lloc, durant els mesos de desenvolupament del projecte s'ha tingut oportunitat d'assistir a diversos esdeveniments relacionats amb la programació d'aplicacions mòbils per diverses plataformes, inclosa la d'Apple. Fruit de l'assistència a aquestes conferències, del *networking* generat amb diverses empreses i professionals del sector, i de l'estudi de les xifres de negoci relacionades amb les aplicacions disponibles a la botiga oficial d'Apple, l'AppStore, s'ha arribat a la conclusió que la viabilitat econòmica del projecte UOC Algorithms tal com s'havia plantejat inicialment no era gens clara.

Així, després d'analitzar detingudament aquests dos punts s'ha arribat a la conclusió que, tant pel tipus d'aplicació que es tracta, com per conviccions personals, la millor alternativa pel que fa a llicències és optar per un model totalment lliure i deixar la possible rendibilitat de l'aplicació en altres tasques auxiliars que se'n puguin derivar, per exemple en forma de prestigi professional o a través de l'elaboració d'aplicacions similars sota comanda amb continguts d'una altra àrea. Aquest nou plantejament és, de fet, més coherent amb la secció inicial d'agraïments mostrada en aquest mateix document.

D'aquesta manera, la decisió final sobre llicències queda molt més simplificada:

- Per tots els continguts de manera íntegra, llicències lliures de tipus GPL (GNU General Public License) i Creative Commons. Això fa que tothom pugui fer l'ús que li sembli més adequat de qualsevol material que trobi a l'aplicació UOC Algorithms.
- La pròpia aplicació UOC Algorithms, entesa com el codi font en llenguatge Objective-C i tots els recursos complementaris (imatges, documents web, arxius de configuració, etc.), també llicència GPL perquè tothom qui ho vulgui pugui analitzar el codi font, col·laborar en la seva programació, corregir errors, aportar idees, etc. Per complir amb aquest requeriment, un cop lliurat el TFC el projecte estarà disponible en un repositori públic de codi font (GitHub).

Aquesta decisió fa que, al mateix temps, la tasca de creació de continguts passi a ser molt més senzilla, ja que ara es pot fer ús de tots els recursos lliures ja existents a la Xarxa i copiar-los, modificar-los, integrar-los i, en general, manipular-los de la manera que es desitgi per acabar creant uns continguts totalment a mida de l'aplicació UOC Algorithms.

Finalment, és convenient remarcar que l'aplicació disposa d'una pantalla informativa on, entre d'altres qüestions, es fa referència a aquestes llicències aplicades. D'aquesta manera l'usuari pot tenir clar que és lliure de fer pràcticament qualsevol ús dels continguts que hi troba sense necessitat de demanar autorització a ningú.

### 4.2.3. Estructuració dels continguts

---

Un dels aspectes que es consideren clau per l'èxit de l'aplicació UOC Algorithms és la presentació uniforme dels continguts. Cal tenir en compte que trobar tota mena de recursos a la Xarxa és una activitat extremadament senzilla atès l'enorme volum d'informació que hi és present. No obstant, presenta el gran inconvenient de no mantenir una uniformitat que permeti comparar i estudiar aquests continguts de manera senzilla. Amb aquest projecte, per tant, es pretén aconseguir una presentació tan homogènia com sigui possible i una qualitat gràfica i de disseny que facilitin el procés d'aprenentatge. No es vol acumular informació, sinó presentar només la que es consideri òptima.

Al mateix temps el fet de mantenir una presentació uniforme enllaça amb un dels requeriments principals del projecte: la persistència en base de dades. És evident que l'aplicació, atesa la seva naturalesa de repositori de continguts, farà servir una base de dades interna per emmagatzemar-los. Per tant, com més similar sigui l'estructura de tots aquests recursos, més senzill serà el procés de guardar-los en taules i, al mateix temps, més fàcil serà la creació de nous materials educatius.

L'estructura que es decideix que tindrà, doncs, cada contingut de l'aplicació UOC Algorithms és la següent:

- **Un títol breu però descriptiu.** És el que es mostra en el menú principal de navegació pels continguts, de manera que cal que sigui concís per tal que l'usuari en tingui prou per saber si es tracta del que està buscant.
- **Una descripció curta.** Tot i que en l'actual versió de l'aplicació no es fa servir, disposar d'aquesta petita explicació sobre el contingut pot resultar útil a l'hora de pensar en exportació de continguts, migració a altres plataformes, etc.
- **Un autor o autors.** Atès que els continguts tenen llicències lliures, les fonts utilitzades per la seva creació poden ser múltiples, així que és bona idea intentar guardar tots els autors que hagin participat, directa o indirectament, en l'elaboració dels recursos.



- **El cos del contingut.** Inclou explicacions teòriques, gràfics, exemples de codi font i, en general, qualsevol mena de recurs tipogràfic o audiovisual que permeti a l'usuari assimilar de la millor manera el concepte que s'explica.
- **Resta d'atributs complementaris.** Per exemple l'àrea i la categoria a la que pertany el contingut, l'idioma, la data de publicació, etc. Tenen com a finalitat estructurar millor la manera com es mostren els recursos educatius a l'usuari.

#### 4.2.4. Consideració inicial sobre la interfície gràfica

---

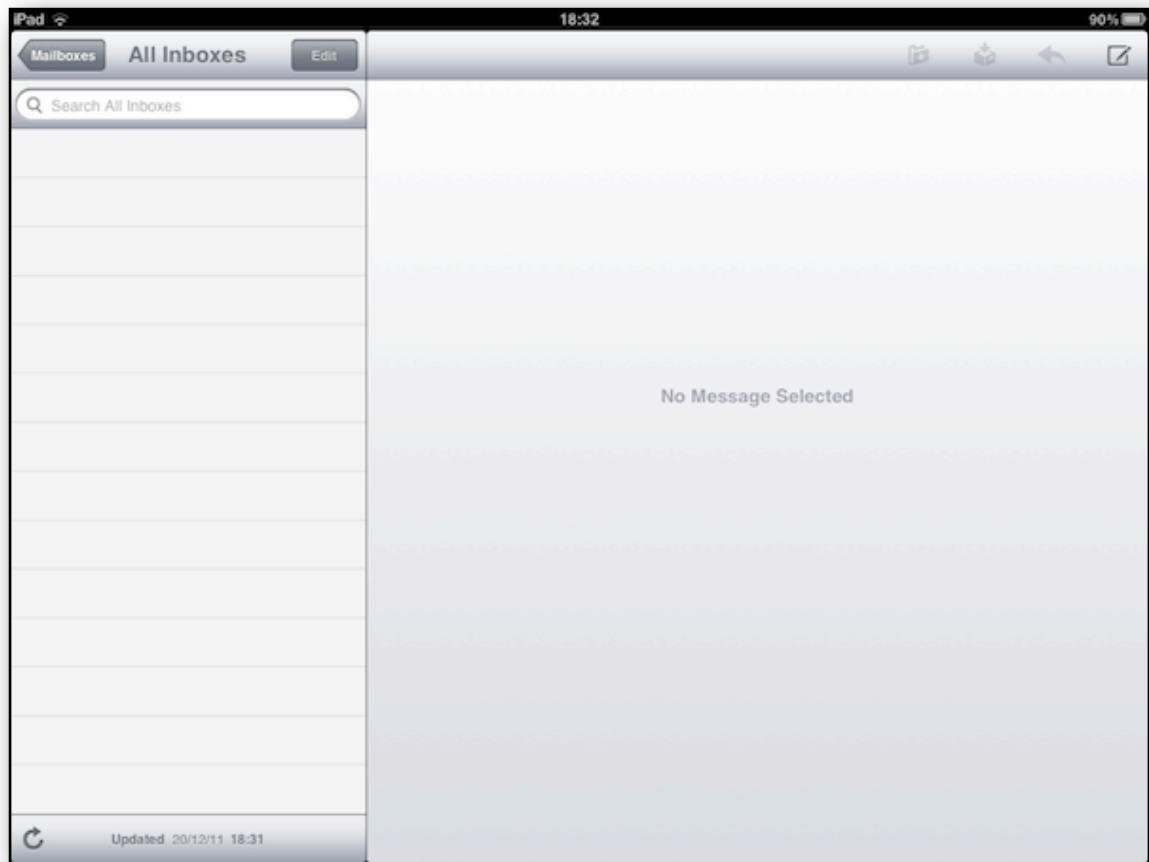
Dintre de les categories d'aplicacions que Apple estableix, UOC Algorithms és sens dubte una aplicació educativa pura. Per tant, i a diferència del que succeeix amb altres aplicacions per iPad que s'anomenen immersives i que busquen que l'usuari s'introdueixi totalment dins de l'aplicació, amb UOC Algorithms es pretén mostrar una interfície tant neta i clara com sigui possible, ja que el que realment ha de tenir importància són els continguts.

D'aquesta manera, per dur a terme el projecte s'han pres com a model algunes de les principals aplicacions educatives i de productivitat que ja hi ha publicades a l'App Store, inclosa alguna d'oficial d'Apple, com per exemple l'aplicació de correu Mail que es mostra en la figura 4.2.2. Fent-ho així es mostra a l'usuari una interfície que ja li és familiar i, per tant, es redueix pràcticament a zero el temps necessari per entendre el funcionament de l'aplicació.

En aquest tipus d'aplicacions s'aprecia una estructura habitual mestre - detall. En la part de l'esquerra hi apareix l'índex de continguts, agrupats per temàtiques o nivells, amb els títols mostrats de manera clara i amb un cercador a la part superior. En la part de la dreta, la principal, s'hi mostra el contingut i una barra superior en la que és possible triar entre diferents accions auxiliars.

Exactament de la mateixa manera es desitja que funcioni UOC Algorithms. Però no s'ha d'entendre aquest punt com una manca d'originalitat, sinó com l'interès en seguir les directrius i recomanacions que Apple marca als desenvolupadors d'aplicacions per tal d'aconseguir, d'aquesta manera, que tot el conjunt d'aplicacions disponibles per iPad que tenen una naturalesa comuna també tinguin una aparença visual similar. Així doncs, UOC Algorithms complirà els requeriments d'Apple.

**Figura 4.2.2. Aplicació Mail oficial d'Apple per dispositiu iPad.**



### 4.3. Capítol 3: Disseny

En aquest capítol s'entra en temes de disseny d'aplicacions i es veuen diagrames que ajuden a mostrar com s'ha organitzat. Per una banda tenim un diagrama UML amb les classes que formen el model de l'aplicació i que, per tant, també seran les que tindran persistència. Per una altra banda es mostra el detall de les taules de la base de dades interna. També es planteja la conveniència d'utilitzar el patró de disseny MVC (Model View Controller).

#### 4.3.1. Diagrama de classes model - diagrama ER

---

Com en tota aplicació que inclou una interfície gràfica d'usuari, UOC Algorithms està desenvolupada a partir d'un conjunt de classes que inclou classes de negoci, classes de control i classes frontera.

Les classes frontera es corresponen al que en el patró de disseny MVC juga el paper de les vistes. Bàsicament consisteixen en controls gràfics que es mostren a l'usuari per tal d'interactuar amb l'aplicació. Les classes de control, al seu torn, es corresponen amb el que en el model MVC fa de controlador: de manera resumida, totes les classes que gestionen la interacció entre el model i les vistes. Tant unes com altres tenen un paper secundari, deixant les classes de negoci com les de més importància per entendre el funcionament del programari. Finalment, aquestes classes de negoci, que són les que en el patró MVC es corresponen amb el model, són les que reflecteixen com s'estructuren les dades de UOC Algorithms.

En una aplicació per iPad creada en l'entorn de desenvolupament Xcode i fent servir el framework Core Data, la relació entre aquestes classes de negoci i les taules de la base de dades és molt estreta, de tal manera que es pot considerar que una cosa i l'altra són equivalents.

Per desenvolupar-les es parteix del disseny d'una sèrie d'entitats, a les que es pot considerar d'alguna manera les taules de la base de dades, i se'n deriven de manera directa les classes, que es creen com a subclasses de la classe `NSObject`<sup>7</sup>.

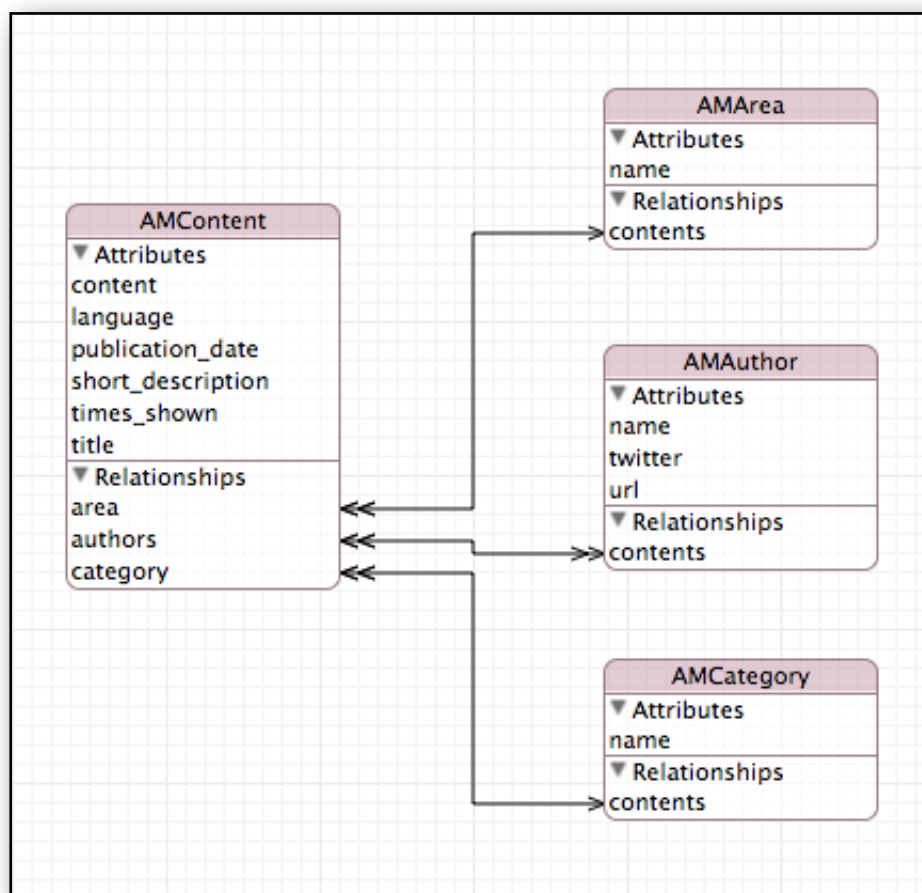
En la figura 4.3.1 següent es mostra el que és un diagrama a mig camí entre un clàssic diagrama UML de classes i un diagrama ER de base de dades. En aquesta representació cal entendre les línies amb una punta a una banda i dues a l'altra com a relacions 1 a molts, mentre que les línies amb dues puntes a cada banda equivalen a relacions molts a molts.

Com es pot veure a la figura, UOC Algorithms utilitza una estructura de taules molt senzilla per mantenir un registre dels continguts: es vol guardar l'àrea i categoria a la que pertany cada recurs, els seus autors i una sèrie de dades bàsiques.

---

<sup>7</sup> Al llarg d'aquest document apareixen en diverses ocasions els noms d'algunes classes típiques del desenvolupament d'aplicacions per iOS o concretes del desenvolupament de l'aplicació UOC Algorithms. Totes les que comencen pel prefix NS- (per Next Step) o UI- (per User Interface) són de les primeres. Les que, en canvi, comencen per AM- (per les inicials de l'autor d'aquest TFC) corresponen al segon grup.

Figura 4.3.1. Diagrama de classes model amb Core Data.



### 4.3.2. Detalls de la persistència dels continguts en base de dades

Tenint en compte el que s'ha comentat en l'apartat anterior, és evident que la base de dades on s'emmagatzemen totes les dades de l'aplicació és realment molt senzilla. La pròpia taula de continguts es pot concebre de dues maneres diferents: com a primera opció, el camp *content* pot emmagatzemar tot el contingut directament en mode text; com a segona alternativa, aquest camp pot guardar únicament un nom d'arxiu, que serà el document HTML que tindrà tot el contingut amb el format adequat.

Tot i que la segona alternativa és la més senzilla d'implementar, presenta el gran inconvenient que dificulta molt les recerques. Així, si un usuari vol buscar un contingut a partir de paraules a l'atzar que cregui que hi poden aparèixer, resulta lent haver de recórrer un gran nombre d'arxius HTML mirant-ne el contingut interior. Una altra alternativa seria emmagatzemar una sèrie de paraules clau per cada recurs i que les cerques només es poguessin fer sobre aquestes paraules.

Per contra, la primera opció, tot i ser més complexa des del punt de vista tècnic, facilita enormement l'acció de buscar i, al mateix temps, evita haver de guardar una munió de documents web al paquet<sup>8</sup> de l'aplicació UOC Algorithms. És per aquest motiu que s'ha triat aquesta manera de fer persistents els continguts.

Per tant, les quatre taules que recullen tota la persistència de l'aplicació UOC Algorithms són les següents:

**Figura 4.3.2. Taula AMArea a Core Data.**

AMArea		
Camp	Tipus	Descripció
<b>id</b>	<b>Numèric</b>	<b>Identificador únic de l'àrea a què pertany un contingut</b>
name	Text	Nom de l'àrea

**Figura 4.3.3. Taula AMAuthor a Core Data.**

AMAuthor		
Camp	Tipus	Descripció
<b>id</b>	<b>Numèric</b>	<b>Identificador únic de l'autor d'un contingut</b>
name	Text	Nom i cognoms de l'autor
url	Text	Adreça web (p.ex. el seu bloc)
twitter	Text	Nom d'usuari a Twitter

<sup>8</sup> S'acostuma a utilitzar el mot anglès 'bundle' per fer referència a aquest paquet.

**Figura 4.3.4. Taula AMCategory a Core Data.**

AMCategory		
Camp	Tipus	Descripció
<b>id</b>	<b>Numèric</b>	<b>Identificador únic de la categoria a què pertany un contingut</b>
name	Text	Nom de la categoria

**Figura 4.3.5. Taula AMContent a Core Data.**

AMContent		
Camp	Tipus	Descripció
<b>id</b>	<b>Numèric</b>	<b>Identificador únic del contingut</b>
title	Text	Títol del contingut
short_description	Text	Descripció curta
content	Text	Contingut complet
publication_date	Data	Data de publicació
times_shown	Numèric	Nombre de vegades que s'ha mostrat
language	Text	Idioma
<b>id_autor *</b>	<b>Numèric</b>	<b>Autor que l'ha escrit</b>
<b>id_area</b>	<b>Numèric</b>	<b>Àrea a què pertany</b>
<b>id_categoria</b>	<b>Numèric</b>	<b>Categoria a què pertany</b>

No obstant, cal tenir en compte un detall realment important pel que fa a aquestes taules: la tecnologia Core Data emprada en el desenvolupament de UOC Algorithms no ens permet crear les taules seguint un model ER convencional. Per exemple, en les quatre taules exposades tots els camps que apareixen en negreta no s'han incorporat a les taules en forma de camps, sinó que ha estat Core Data qui s'ha encarregat de gestionar-los. Així, la relació molts a molts existent entre AMContent i AMAuthor que, en principi, s'hauria de recollir amb una altra taula auxiliar AMContentAuthor la gestiona Core Data de manera automàtica.

Es podria haver optat per prescindir del framework Core Data, crear manualment les taules en la base de dades desitjada (p.ex. SQLite, que és la mateixa que fa servir internament Core Data) i després, via codi font amb llibreries escrites en llenguatge C o Objective-C connectar-hi per introduir-hi i extreure'n informació. Però es tracta d'una tasca prescindible atès que Core Data funciona realment de manera adequada a les necessitats del projecte.

Tot això no vol dir, és clar, que Core Data no treballi internament amb taules en el sentit ortodox. Sí que ho fa. Si es consulta l'estructura de les taules directament a l'arxiu UOCAlgorithms.sqlite, que es correspon amb la base de dades pròpiament dita, s'hi troben les següents definicions de taules:

- CREATE TABLE **ZAMAREA** ( Z\_PK INTEGER PRIMARY KEY, Z\_ENT INTEGER, Z\_OPT INTEGER, ZNAME VARCHAR )
- CREATE TABLE **ZMAUTHOR** ( Z\_PK INTEGER PRIMARY KEY, Z\_ENT INTEGER, Z\_OPT INTEGER, ZURL VARCHAR, ZNAME VARCHAR, ZTWITTER VARCHAR )
- CREATE TABLE **Z\_2CONTENTS** ( Z\_2AUTHORS INTEGER, Z\_4CONTENTS INTEGER, PRIMARY KEY (Z\_2AUTHORS, Z\_4CONTENTS) )
- CREATE TABLE **ZAMCATEGORY** ( Z\_PK INTEGER PRIMARY KEY, Z\_ENT INTEGER, Z\_OPT INTEGER, ZNAME VARCHAR )
- CREATE TABLE **ZAMCONTENT** ( Z\_PK INTEGER PRIMARY KEY, Z\_ENT INTEGER, Z\_OPT INTEGER, ZTIMES\_SHOWN INTEGER, ZAREA INTEGER, ZCATEGORY INTEGER, ZPUBLICATION\_DATE TIMESTAMP, ZTITLE VARCHAR, ZCONTENT VARCHAR, ZLANGUAGE VARCHAR, ZSHORT\_DESCRIPTION VARCHAR )

S'hi observen les quatre taules que s'han detallat anteriorment i una taula extra, Z\_2CONTENTS, que és la que duu a terme la relació molts a molts entre AMAuthor i AMContent, tal com s'ha comentat amb anterioritat. Realment, però, al llarg de tot el desenvolupament de l'aplicació no es tenen en compte aquestes taules. Aquest detall forma part d'un baix nivell del qual Core Data s'encarrega directament, exposant un mètode de treball amb objectes de les classes NSFetchRequest, NSPredicate i NSEntityDescription, de molt més alt nivell i amb les que, per tant, resulta més còmode treballar.

### 4.3.3. Utilització de formes normals (FN) a la base de dades

---

En la teoria de bases de dades relacionals, es parla de formes normals (FN) per fer referència al nivell en què la informació està dividida en taules de tal manera que no n'hi hagi de repetida en cadascuna d'elles. A major nivell de forma normal, menor és el risc que a la base de dades hi hagi inconsistències i anomalies, per bé que, a canvi, la

informació està més disgregada i, per tant, quan es vol recuperar de manera completa és necessari fer unions entre diferents taules. Dur a terme aquestes unions té un cost major o menor en funció de les dades que es volen recuperar i del nivell de formes normals que s'ha fet servir: primera forma normal (1FN, poc recomanada), tercera forma normal (3FN, la més habitual) o fins a sisena forma normal (6FN, poc habitual).

En els sistemes gestors de bases de dades convencionals es recomana treballar, com a mínim, en tercera forma normal (3FN), perquè presenta una molt bona relació entre l'estructuració adequada de la informació i el cost que té a posteriori recuperar-la de manera conjunta.

En dispositius mòbils com l'iPad, en canvi, cal anar amb compte amb els dissenys de bases de dades que es duen a terme. Per una banda, cal tenir en compte que la potència del processador d'aquests aparells no és la mateixa que la d'un ordinador convencional. Per una altra, és evident que el gestor de bases de dades que fan servir (en el cas de l'aplicació UOC Algorithms, SQLite) és molt menys versàtil que els productes comercials habituals de servidor com els de les companyies Oracle i Microsoft, entre d'altres. És per això que a l'hora de dissenyar la persistència en base de dades d'una aplicació per iPad sovint cal renunciar una mica a aquest procés d'estructuració de la informació en formes normals per tal de guanyar en rapidesa a l'hora de recuperar la informació.

En general es recomana pensar de quina manera es mostrarà la informació a l'usuari en les diferents pantalles de l'aplicació i, partint d'aquest punt, guardar-la en taules de tal manera que la traducció de taules en pantalles sigui gairebé immediata. Sovint això suposa quedar-se a mig camí entre la segona i la tercera formes normals (2FN i 3FN).

No obstant, la base de dades que fa servir UOC Algorithms és molt senzilla, de manera que no hi ha problemes en aquest sentit i es pot respectar la tercera forma normal (3FN). S'indica aquesta qüestió pensant en el probable increment de complexitat que pugui sorgir en posteriors versions de l'aplicació si l'increment de funcionalitats requereix un disseny de base de dades més complex.

#### **4.3.4. Utilització del model MVC**

---

En desenvolupament de programari s'utilitzen diversos patrons de disseny que permeten, pel fet de fer-los servir adequadament, obtenir un producte final de major qualitat, ja que s'aprofita que hi ha una sèrie d'escenaris coneguts en els que es poden aplicar unes tècniques de programació que s'han provat amb èxit anteriorment.

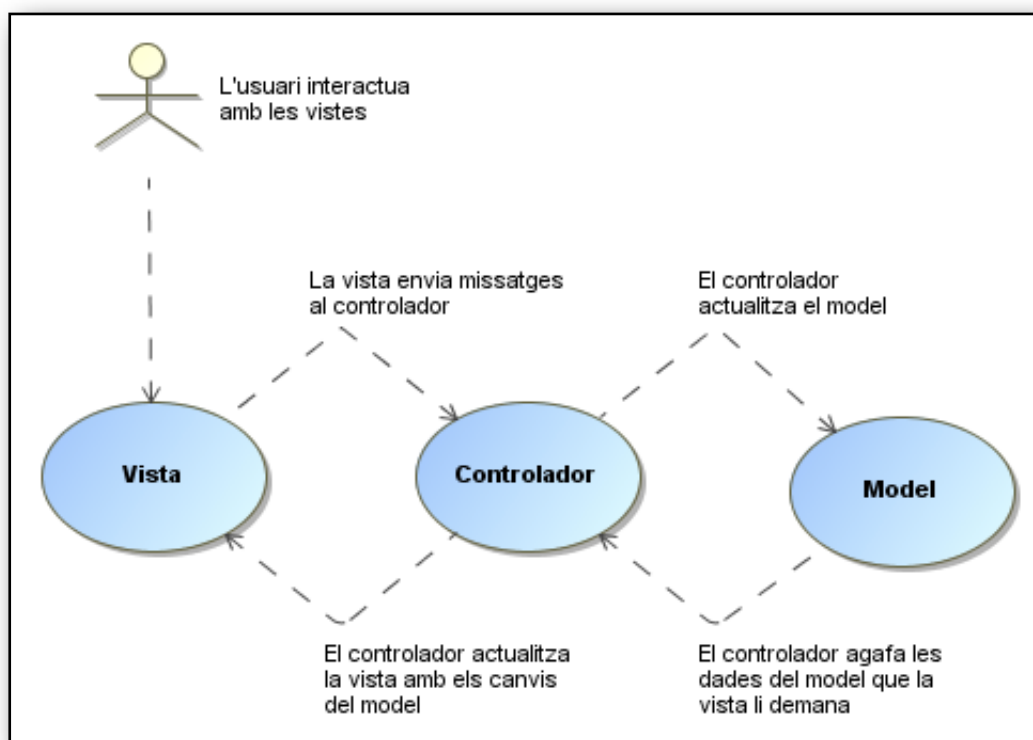


D'entre tots els patrons de disseny existents, n'hi ha un que és, sens dubte, el més anomenat i utilitzat en el desenvolupament d'aplicacions per plataformes Apple: el Model - Vista - Controlador, o MVC.

Aquest patró es pot explicar de manera breu dient que qualsevol objecte que formi part d'una aplicació serà o bé un objecte Model, o bé un objecte Vista o bé un objecte Controlador. Només una possibilitat en cada cas.

Els objectes Vista són aquells que l'usuari visualitza per pantalla, per exemple un botó o una imatge. Els objectes Model són els que guarden i gestionen les dades de l'aplicació. Tant els objectes Vista com els Model ignoren absolutament l'existència dels altres<sup>9</sup>. Això fa que perquè l'aplicació actuï com un tot calgui un tipus d'objectes integradors, que són els Controlador.

**Figura 4.3.6. Esquema del patró de disseny Model - Vista - Controlador (MVC)<sup>10</sup>.**



<sup>9</sup> De fet, hi ha diferents interpretacions i aplicacions del patró de disseny MVC pel que fa al grau de coneixement que les vistes tenen del model. En algunes, les vistes sí que tenen coneixement del model. No obstant, en l'aplicació més habitual del patró MVC treballant amb Cocoa Touch, model i vistes són del tot independents i es desconeixen totalment.

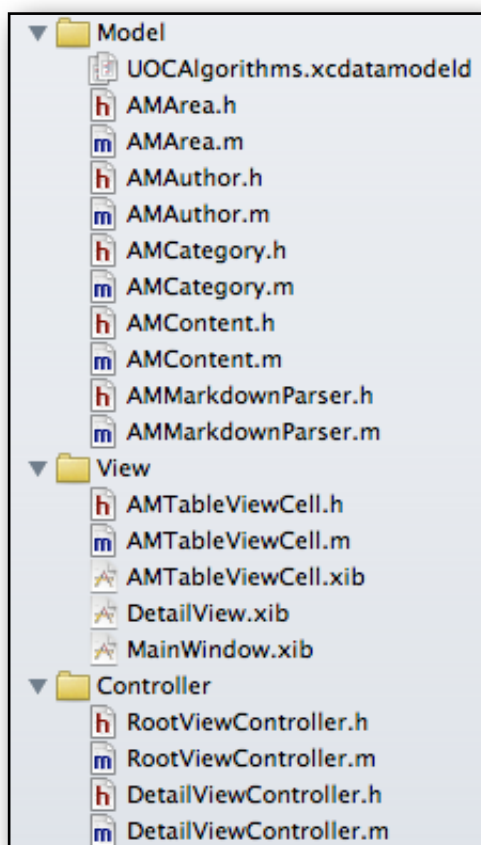
<sup>10</sup> Inspirat en l'esquema del llibre "iOS Programming, The Big Nerd Ranch Guide" dels autors Jon Conway i Aaron Hillegass, la informació del qual es troba a la bibliografia.

En el cas del desenvolupament d'aplicacions per iPad aquest model es porta a l'extrem, facilitat pel fet que la combinació del llenguatge de programació Objective-C i el framework Cocoa Touch ho potencien al màxim.

UOC Algorithms aplica estrictament aquest patró de disseny, així que en cap cas existeix un objecte Model que conegui l'existència d'un objecte Vista (ni a l'inrevés). Tots dos es comuniquen exclusivament amb els objectes Controlador. Per aquest motiu s'ha comentat amb anterioritat en aquest document que la programació dels objectes Controlador és la que resulta més sensible a l'aparició d'errors.

En la figura següent es mostra el contingut dels tres blocs del model MVC perfectament separats en la versió final de UOC Algorithms:

**Figura 4.3.7. Xcode gestionant grups separats per les classes MVC.**



### 4.3.5. Selecció dels continguts per la primera versió

---

Si bé a mig termini, i especialment a llarg, es pretén que l'aplicació UOC Algorithms disposi d'un repositori realment voluminós de continguts educatius, la intenció d'aquest projecte no és tant la creació inicial d'aquests recursos com la del marc en que es mostren a l'usuari en un dispositiu iPad i, de manera complementària, la definició de l'estructura d'aquests per facilitar el procés de crear-ne de nous.

No obstant, és evident que per tal de mostrar el funcionament de tot el sistema calen alguns continguts inicials que permetin avaluar quin és, i quin serà, el funcionament de l'aplicació. Així mateix, triant aquests continguts inicials de manera que pertanyin a diverses àrees dins dels estudis d'informàtica s'aconsegueix que quedi més clara la manera com s'organitzen els recursos educatius.

En conseqüència, la selecció de continguts per aquesta primera versió de l'aplicació és la següent:

- **Funcionament detallat dels punters de memòria en llenguatge C.** Inclou un text amb l'explicació del concepte i diversos fragments breus de codi que ajuden a entendre'l. Aquest contingut cau en l'àrea de fonaments de programació.
- **Algorisme de la distància de Levenshtein.** Inclou, com en el cas anterior, una explicació inicial, un apartat extens amb el codi font corresponent a aquest algorisme i un petit exemple del seu funcionament. Aquest contingut cau també en l'àrea de fonaments de programació.
- **Patró de disseny Singleton.** Consta també d'explicacions basades en text, apartats amb codi font i petita representació gràfica. Aquest contingut forma part de l'àrea de patrons de disseny.
- **Funcionament exhaustiu dels tipus abstractes de dades Pila i Llista.** Com en els casos anteriors, i com passarà en el futur per la majoria de continguts, disposen d'explicacions a base de text, fragments de codi font i alguns gràfics. Aquests continguts cauen en l'àrea d'estructures de dades.

Amb aquests cinc continguts inicials pertanyents a tres àrees diferents és possible donar una bona idea de quin és i serà el funcionament de UOC Algorithms a mesura que el repositori de recursos educatius sigui més extens.

## 4.4. Capítol 4: Implementació

En aquest capítol s'aprofundeix, per una banda, en aspectes tècnics de la implementació de tot el que s'ha vist en els capítols anteriors. I per altra banda, atès que l'aplicació final UOC Algorithms no es pot posar a provar en un ordinador convencional pels requeriments que té, es mostren una sèrie de captures de pantalla per facilitar la comprensió del seu funcionament.

### 4.4.1. Eines utilitzades

---

El desenvolupament d'una aplicació per iPad requereix l'ús d'una sèrie d'eines específiques tant a nivell de maquinari com de programari. A banda, cal emprar també altres recursos que, malgrat no ser d'obligatòria utilització, sí que són de gran ajuda pel programador. En el cas concret d'aquest projecte, les eines que s'han emprat són les següents:

- Ordinador MacBook Pro amb sistema operatiu OS X 10.7 (Lion).
- Entorn de desenvolupament Xcode.
- Programari Instruments que complementa Xcode.
- Programari Komodo Edit per edició de codi font extern a Xcode.
- Programari Pixelmator per dur a terme la creació i retoc d'imatges.
- Programari MediaWiki per gestionar un sistema wiki amb la documentació del projecte a mesura que es va desenvolupant. El material inclòs en aquest sistema serà de gran ajuda per la realització de la memòria final del projecte.
- Programari GanttProject per fer la planificació de fases i tasques i el seu seguiment.
- Sistema de control de versions distribuït Git.
- Eines ofimàtiques estàndards: Pages i Keynote.

### 4.4.2. Detalls tècnics de l'estructura dels continguts

---

En una aplicació per iPad es disposa de diversos sistemes per mostrar contingut estàtic a l'usuari. Per exemple hi ha la possibilitat de fer-ho a través d'objectes UILabel (etiquetes) o a través de cel·les UITableViewCell en un format tabular corresponent a una vista UITableView. No obstant, aquests dos sistemes mostren dificultats per presentar un text formatat amb recursos gràfics i possibles insercions d'imatges.

És per això que el recurs tècnic que es fa servir per mostrar els continguts és, en tots els casos, un objecte de tipus UIWebView treballant conjuntament amb un altre objecte de

tipus UIScrollView. Aquest conjunt permet disposar, per una banda, d'una àrea en la que l'usuari es pot desplaçar còmodament amunt i avall per veure les diferents parts del contingut. I per l'altra banda, dóna la possibilitat de crear textos amb formats que incloguin color i altres recursos tipogràfics, ja que l'objecte UIWebView treballa perfectament tant amb HTML5 com amb CSS3.

Convé no confondre el fet d'utilitzar un control UIWebView per mostrar els continguts amb el fet de crear una aplicació web, aspecte que s'ha comentat amb anterioritat en l'apartat **4.1.3. Enfocament i mètode seguit**. UOC Algorithms és una aplicació nativa per iPad, per tant pot fer ús de totes les prestacions que ofereix el dispositiu sense les limitacions que tenen les aplicacions web. Però, al mateix temps, s'aprofita de la versatilitat que ofereix el format web per mostrar els continguts de la manera més atractiva possible a l'usuari.

### **4.4.3. Introducció al framework Cocoa Touch**

---

En el desenvolupament d'aplicacions per qualsevol plataforma és habitual disposar d'una API bàsica que subministra accés a tota una sèrie de recursos tant de programari com de maquinari. Si es programa en Java, per exemple, s'utilitza la pròpia API de Java, que inclou classes bàsiques com String, FileInputStream o JTextField. Per al desenvolupament per ordinadors de tipus Mac en llenguatge Objective-C es disposa de l'anomenada Mac OS X Cocoa API, que satisfà les mateixes necessitats. Quan, en canvi, es tria programar aplicacions per iPad, l'API que hi ha disponible és una variant de Cocoa enfocada al treball amb dispositius mòbils tàctils: Cocoa Touch.

Cocoa Touch és una interfície de programació desenvolupada principalment en Objective-C i que proveeix una capa d'abstracció sobre el sistema operatiu de l'iPad, iOS, de tal manera que el programador té accés a tot un conjunt de característiques, tant de programari com de maquinari, del dispositiu de manera més senzilla que si hi hagués d'accedir directament.

Un iPad, per exemple, disposa d'un giroscopi integrat que permet conèixer en tot moment l'orientació en l'espai de l'aparell. UOC Algorithms utilitza aquesta característica per tal de rotar automàticament entre una posició vertical i una horitzontal i, així, mostrar els continguts sempre ben orientats a l'usuari en funció de com estigui emprant el dispositiu. Aquesta característica es mostra en les figures 4.4.1 i 4.4.2. Sense el marc d'abstracció que ofereix Cocoa Touch caldria accedir directament al giroscopi i interpretar-ne les dades. Amb Cocoa Touch aquesta tasca es converteix en trivial al disposar d'una interfície senzilla, neta i totalment orientada a objectes.

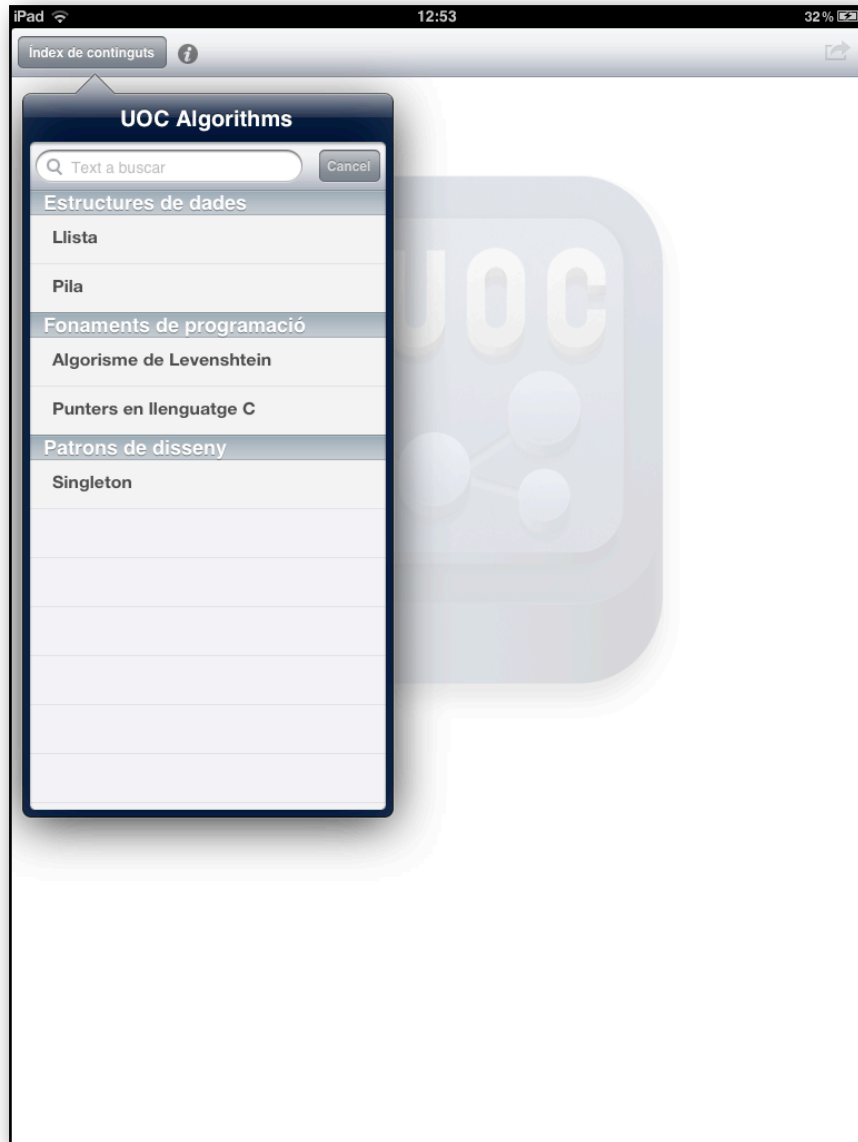
**Figura 4.4.1. UOC Algorithms amb l'iPad en posició horitzontal.**



Així, Cocoa Touch aporta facilitat a l'hora de desenvolupar, permetent accedir a tota mena de recursos (com ara quadres de text, navegador web, acceleròmetre, connexió a la Xarxa, etc.) del dispositiu d'una manera sempre uniforme, independentment que el recurs en qüestió tingui, de manera interna, un grau de complexitat major o menor.

Per al desenvolupament del projecte UOC Algorithms s'ha utilitzat aquesta API de manera constant. De fet, els tres grans trets que caracteritzen aquest projecte a nivell tècnic són, sense cap mena de dubte, la utilització de Cocoa Touch, la gestió de la persistència amb Core Data i l'aplicació, arreu, del patró de disseny MVC.

**Figura 4.4.2. UOC Algorithms amb l'iPad en posició vertical.**



#### 4.4.4. Format i ús de marques per la creació de continguts

Tal com s'ha comentat a l'apartat **4.3.2. Detalls de la persistència dels continguts en base de dades**, una vegada decidit que els continguts es mostren a l'aplicació utilitzant un control `UIWebView`, hi ha dues grans alternatives pel que fa a com crear els continguts des d'un punt de vista tècnic: crear cadascun dels diferents apartats de cada contingut com una document HTML i incloure tots aquests documents HTML en el paquet de l'aplicació, o bé crear els continguts utilitzant text pla i establint un conveni de

marques per tal de mostrar recursos gràfics com negretes, cursives, mides de lletra superiors, colors, etc.

Totes dues alternatives són vàlides i funcionen, i totes dues faran ús intensiu de les tecnologies HTML5 i CSS3. Però tenint en compte aspectes tècnics relacionats, sobretot, amb la facilitat de dur a terme cerques sobre els continguts, s'ha triat la segona alternativa. Això ha fet necessari establir un llenguatge de marques que després s'ha de tenir en compte a l'hora de crear els continguts.

Per establir aquest llenguatge de marques novament es presenten dues alternatives: crear-ne un des de zero totalment adaptat a les necessitats particulars de UOC Algorithms, o buscar-ne un d'estàndard existent a la Xarxa i que sigui tan àmpliament utilitzat com sigui possible. La primera alternativa facilita la creació de les marques i convenis que es vulguin adoptar, però requereix crear manualment tota la funcionalitat per convertir després els textos que facin servir aquestes marques a format HTML. La segona opció, en canvi, obliga a adoptar un conveni de marques ja existent i, per tant, aprendre'n el funcionament, però com a punt positiu ofereix eines ja creades per convertir a format HTML.

Tenint en compte aquestes consideracions, s'ha decidit adoptar un dels llenguatges de marques més populars a la Xarxa en el moment del desenvolupament d'aquest projecte: Markdown.

Markdown és un projecte personal de John Gruber<sup>11</sup> consistent en un conjunt de marques que s'incorporen al text sense restar-li llegibilitat i que, amb l'aplicació posterior d'una eina que interpreta aquestes marques, permet obtenir documents en format HTML. Es tracta d'un projecte lliure que, per tant, pot ser utilitzat en qualsevol altre projecte sense restriccions. Un arxiu escrit utilitzant Markdown tindria un aspecte similar a aquest:

**Figura 4.4.3. Exemple de text utilitzant el format Markdown.**

```
TÍTOL PRINCIPAL
=====
Subtítol també important
-----
Contingut d'un paràgraf.
```

<sup>11</sup> Es pot trobar l'especificació completa de la sintaxi que permet Markdown, a banda de tota la informació del projecte i d'eines relacionades, a l'adreça web <http://daringfireball.net/projects/markdown>.

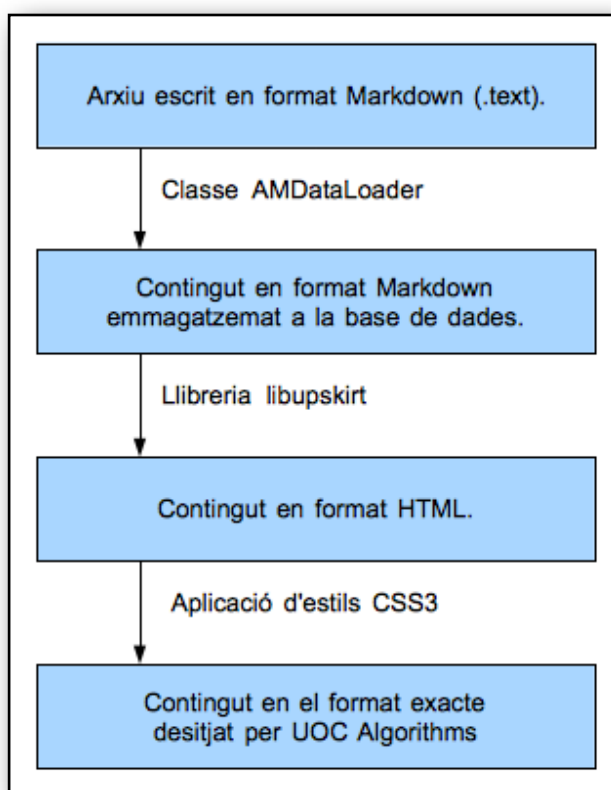


Es pot apreciar que el text resulta del tot llegible i les marques, lluny de fer difícil la lectura o escriptura dels arxius, gairebé la faciliten. En l'exemple de la figura anterior les marques utilitzades són simplement les línies de subratllats.

Un cop es disposa d'un arxiu elaborat amb aquest format hi ha disponibles diverses eines per transformar-lo en un document HTML. D'entre totes, per l'aplicació UOC Algorithms s'ha triat la implementació anomenada libupskirt, feta per Natasha Kerensikova<sup>12</sup>, ja que és de les poques implementacions desenvolupades íntegrament en llenguatge C (per tant, totalment integrable en un projecte desenvolupat en Objective-C) i que no afegeixen funcions addicionals que l'aplicació UOC Algorithms realment no necessita, permetent així que el nivell de complexitat del projecte romanguí baix. A més, com en el cas de Markdown, es tracta també d'una implementació de codi lliure que permet fer-ne l'ús que aquest projecte TFC requereix.

Per tant, el flux de treball des de que es crea un fitxer amb un contingut en format Markdown fins que finalment es mostra a l'usuari a través d'un control UIWebView a l'aplicació UOC Algorithms és el següent:

**Figura 4.4.4. Flux de treball amb arxius en format Markdown.**



<sup>12</sup> Es pot trobar la llibreria amb les explicacions corresponents a l'adreça web <http://fossil.instinctive.eu/libupskirt/home>.

#### 4.4.5. Incorporació dels continguts a la base de dades

---

L'aplicació UOC Algorithms utilitza, com s'ha comentat en apartats i capítols anteriors, una persistència basada en una base de dades SQLite a la que, no obstant, no s'accedeix directament, sinó que s'empra el framework Core Data.

Al mateix temps, cal tenir en compte que l'aplicació fa un ús de la base de dades de només lectura. Certament l'usuari mai introdueix informació nova. L'únic camp que seria susceptible de ser actualitzat és el que guarda el nombre de vegades que un contingut s'ha mostrat als usuaris, però aquest atribut en aquesta primera versió de l'aplicació no està operatiu, i s'hi ha afegit només a mode de línia de treball futura.

Així doncs, és evident que en el moment d'instal·lar l'aplicació en un dispositiu és necessari que disposi ja d'una base de dades plena de continguts que actuï com a repositori d'informació.

Per afegir aquesta informació hi ha, fonamentalment, dues possibilitats:

- Crear un script SQL amb instruccions de tipus INSERT i executar-lo directament contra l'arxiu UOCAlgorithms.sqlite, que és el que conté tota la base de dades.
- Aprofitar que l'aplicació UOC Algorithms ja disposa de tot el model de dades per comunicar amb la base de dades per fer que sigui la mateixa aplicació -o una petita variant- la que afegixi el contingut a la base de dades.

La primera opció seria la més senzilla, però genera un problema: Core Data amaga l'estructura de la base de dades. Tot i això sempre és possible accedir a l'arxiu UOCAlgorithms.sqlite de manera directa amb l'eina sqlite3 i esbrinar quina estructura segueix la base de dades, però això suposa actuar a esquenes del que Core Data espera i fa que, potencialment, en el futur aquesta manera de procedir pugui deixar de funcionar si una nova implementació d'aquest framework canvia l'esquema de les taules.

Si, per exemple, es compara l'entitat que fa referència als autors, AMAuthor, en Core Data (figura 4.3.1) i en SQLite en la figura següent, s'observen diferències tècniques:

**Figura 4.4.5. Taula AMAuthor a SQLite.**

```
CREATE TABLE ZMAUTHOR ( Z_PK INTEGER PRIMARY KEY, Z_ENT  
INTEGER, Z_OPT INTEGER, ZURL VARCHAR, ZNAME VARCHAR,  
ZTWITTER VARCHAR );
```

En concret, els atributs marcats en vermell són atributs propis que Core Data ha afegit a la taula. El primer d'ells és de fàcil interpretació: no és altra cosa que l'identificador numèric que actua com a clau primària. Els altres dos, en canvi, són marques pròpies del framework que, a més, no estan documentades. A partir de l'observació de diferents bases de dades és possible esbrinar quina funció tenen i, a partir d'això, crear scripts SQL que funcionin adequadament. No obstant, tal com s'ha comentat, aquesta manera de procedir podria deixar de ser vàlida en el moment en què una nova versió de Core Data gestionés les taules de manera diferent.

Per tant és molt millor idea fer servir la segona alternativa: si UOC Algorithms està preparada per llegir la base de dades a través de Core Data, no ha de ser gaire més complicat fer que també la pugui escriure. Aquesta és, doncs, l'aproximació que s'ha pres en aquest projecte.

Ara bé, no convé que l'aplicació UOC Algorithms real, la que els usuaris tindran als seus aparells, tingui la sobrecàrrega d'haver d'omplir la base de dades. És per això que s'ha creat una petita variant del projecte que en el moment de compilar-lo permet triar si es vol generar la versió per distribució o una versió per desenvolupament, que és la que permet omplir la base de dades amb tots els continguts.

A nivell tècnic, per omplir la base de dades cal, per una banda, la classe AMDataLoader que s'ha desenvolupat expressament per dur a terme aquesta operació. I, per l'altra banda, són necessaris els arxius amb la informació que ha d'anar a les taules. Aquests arxius tenen un format de text pla. Per exemple per la taula d'autors l'arxiu pot tenir aquest contingut:

**Figura 4.4.6. Contingut de l'arxiu de text per la taula AMAuthor.**

```
Albert Mata;@almata;http://www.albertmata.net  
Scott Chacon;@chacon;http://schacon.github.com
```

En tots aquests arxius de text pla, cada línia equival a un registre a la base de dades. I els diferents camps es posen simplement separats per caràcters punt i coma. La classe AMDataLoader, amb el mètode loadDatabaseUsingManagedObjectContext:, i el framework Core Data s'encarreguen d'abocar tota aquesta informació a les taules de la base de dades SQLite de la manera adequada.

Finalment, pels propis recursos educatius cal crear, a banda d'un arxiu com l'anterior amb totes les capçaleres de cada contingut (títol, autor, data de publicació, idioma, etc.), un

arxiu també de text pla però en format Markdown (extensió .text) on hi hagi totes les explicacions pertinents. El nom d'aquest arxiu s'incorpora en l'arxiu de capçaleres en la línia corresponent a aquell contingut.

Amb aquest sistema s'aconsegueixen omplir les taules de SQLite sempre que es desitgi, de manera que resulta senzill publicar una actualització de l'aplicació i que dugui incorporada una base de dades amb més informació que la versió anterior.

#### **4.4.6. Presentació del funcionament de l'aplicació UOC Algorithms**

---

Finalment, i després de tenir en consideració tots aquests aspectes més tècnics i de dur a terme una tasca de desenvolupament important, s'ha obtingut el producte final UOC Algorithms. En aquest apartat es mostra el funcionament bàsic d'aquesta aplicació, atès que, al tractar-se de programari específic per iPad, no és possible comprovar-ne l'ús en un ordinador sense una configuració específica.

El primer que es veu a l'iniciar l'aplicació és directament la interfície amb què l'usuari interactuarà tota l'estona, tal com es pot apreciar a la figura 4.4.7. A l'esquerra s'hi mostra l'arbre de continguts, convenientment agrupats per categories, i un cercador a la part superior. A la dreta hi apareix l'àrea principal de la interfície, on es mostraran tots els recursos didàctics. En un primer moment només s'hi mostra el logotip de UOC Algorithms, suficientment difuminat perquè no agafi més protagonisme del compte.

S'hi aprecien també dos petits botons que ofereixen funcionalitats auxiliars: per una banda un boto d'informació (a la part superior esquerra de l'àrea dreta principal) que mostrarà els crèdits de l'aplicació; per l'altra, un botó que permetrà dur a terme accions amb el contingut mostrat en pantalla (a la part superior dreta de la imatge), però que en aquesta primera visió apareix inhabilitat.

Un cop l'usuari selecciona qualsevol dels continguts en l'arbre de l'esquerra, aquest es mostra amb tot detall a la part dreta de la pantalla. La representació pot incloure explicacions basades en text, imatges, blocs de codi font i fins i tot fragments de vídeo. Aquest darrer punt donarà la possibilitat d'afegir demostracions pràctiques de determinades tècniques.

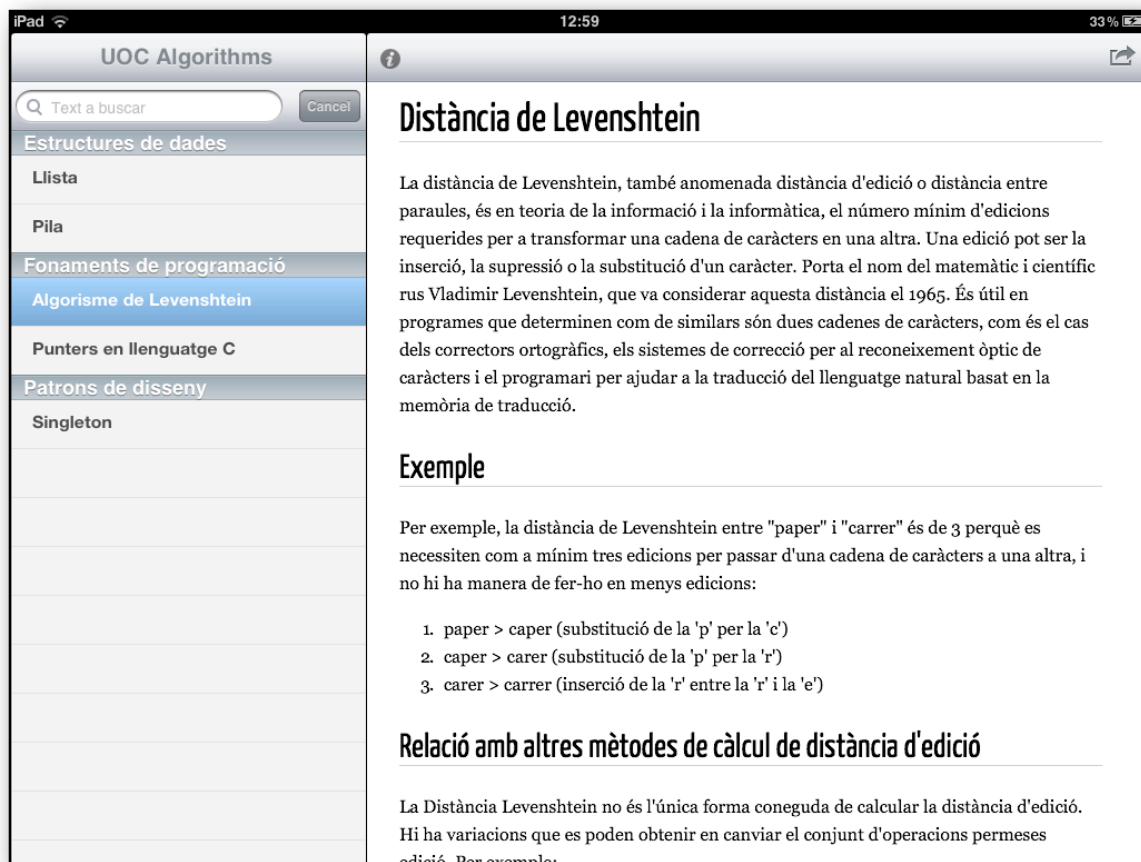
**Figura 4.4.7. UOC Algorithms en el moment d'iniciar l'aplicació.**



A més, en l'arbre de continguts es deixa marcat el que s'ha seleccionat, per facilitar a l'usuari retroalimentació de en quin punt concret es troba. Fer-ho d'aquesta manera està totalment en línia amb les recomanacions d'Apple pel disseny d'aplicacions amb aquest model mestre - detall.

En la figura 4.4.8 es pot apreciar també com s'ha procurat que la part més important des del punt de vista visual siguin els continguts didàctics. La resta de la interfície utilitza tonalitats de colors més apagades i amb menys contrast, per tal que l'àrea principal prengui tot el protagonisme que li correspon.

Figura 4.4.8. UOC Algorithms quan s'ha seleccionat un contingut.

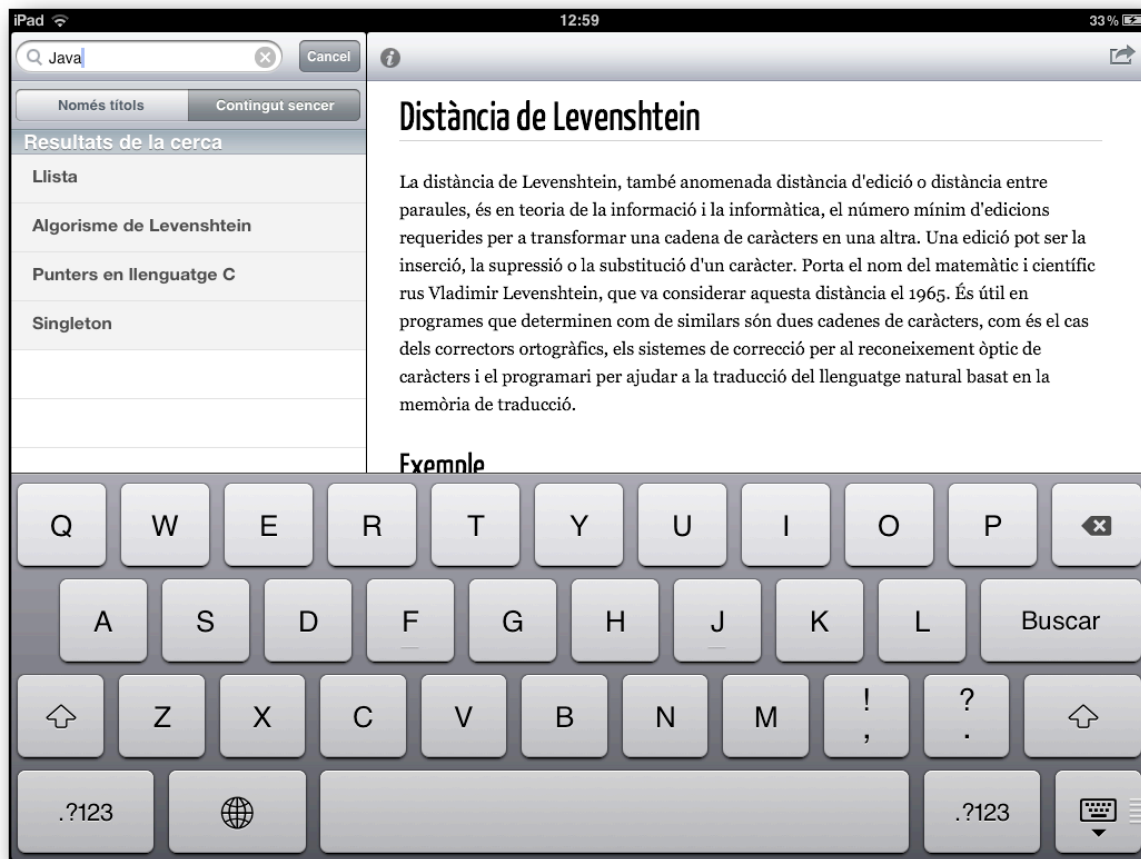


Una de les funcionalitats bàsiques de l'aplicació és el cercador. En aquesta versió inicial hi ha pocs continguts perquè només es pretén mostrar el funcionament i crear el marc de treball, però a mesura que el repositori d'informació creixi serà fonamental disposar d'un cercador que faciliti l'accés als recursos.

El cercador que s'ha implementat, tal com es pot apreciar a la figura 4.4.9, permet fer cerques tant en els títols dels continguts com en el seu interior. Així, si es volen filtrar els recursos que corresponguin a algorismes es pot escriure el mot 'algorismes' (o una part del mateix) i els resultats es filtraran automàticament per mostrar només aquells que tenen aquest terme en el seu títol. Si, per contra, es volen veure tots els recursos que inclouen en algun punt -títol o no- la paraula 'Java' el nombre de resultats serà més extens i menys precís, però pot ser útil a l'hora de cercar un contingut basant-se en alguna paraula clau.

Quan l'usuari ha acabat de buscar, disposa d'un botó per cancel·lar que torna a mostrar l'arbre sencer de continguts, exactament igual com es mostrava abans de començar la cerca.

**Figura 4.4.9. UOC Algorithms amb l'opció de cerca activada.**



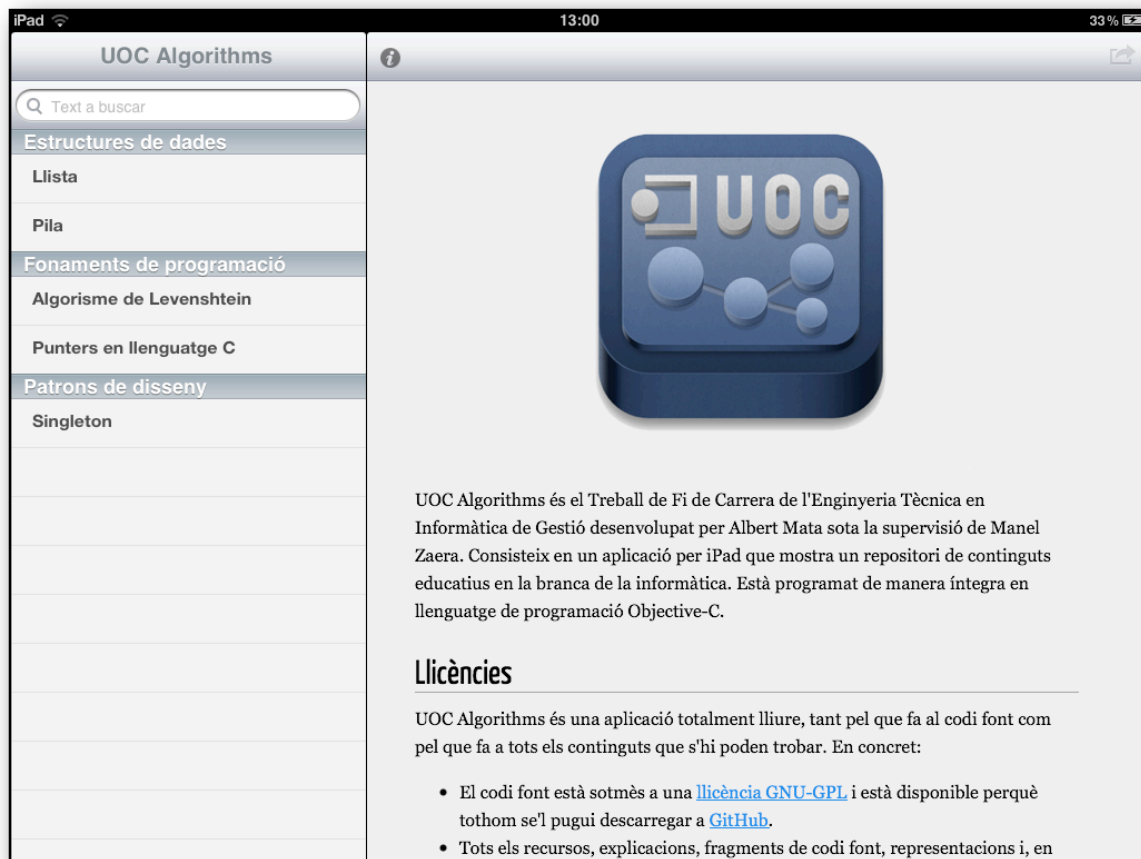
Una altra de les opcions que hi ha disponibles és la de mostrar els crèdits de l'aplicació, tal com es pot veure en la figura 4.4.10 a continuació. Aquesta opció és especialment útil perquè no només mostra informació sobre l'autor de l'aplicació, sinó que també exposa tot el que fa referència a les llicències, a les col·laboracions i sol·licituds i a la manera de contactar amb el desenvolupador.

Pel que fa a les llicències, s'explica que UOC Algorithms és una aplicació 100% lliure, tant a nivell de continguts com del propi codi font, i s'enllaça amb el repositori públic on estarà disponible el codi de l'aplicació (GitHub).

Pel que fa a col·laboracions i sol·licituds, es comenta de quina manera es poden fer arribar a l'autor i com seran ateses arribat el cas.

Finalment, es facilita tant una adreça de correu electrònic com un compte a la coneguda xarxa social Twitter per tal que qui ho desitgi pugui posar-se en contacte amb qui ha creat l'aplicació.

Figura 4.4.10. UOC Algorithms mostrant els crèdits de l'aplicació.



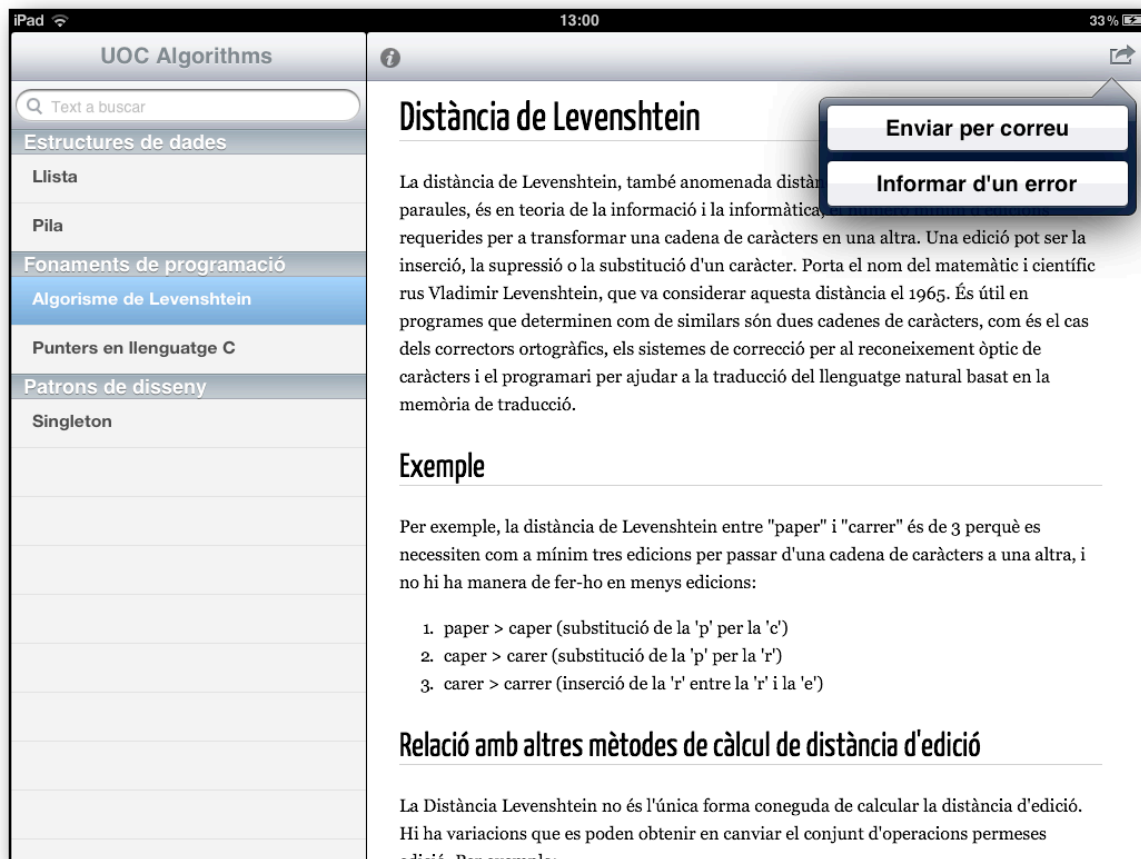
Sempre que s'està mostrant un contingut en pantalla, hi ha disponible un botó a la part superior dreta que permet dur a terme dues accions auxiliars: enviar per correu i informar d'un error. Això es mostra amb detall a la figura 4.4.11.

Amb la primera es permet a l'usuari enviar per correu electrònic el contingut que està visualitzant. Per fer-ho, l'aplicació UOC Algorithms interactua amb l'aplicació Mail, nativa de l'iPad, i configura un nou correu electrònic amb l'assumpte i el cos corresponents, de manera que només calgui afegir un destinatari i enviar-lo.

Amb la segona es segueix el mateix procediment d'obrir l'aplicació de correu, però en aquesta ocasió el destinatari ja queda fixat (serà l'autor de l'aplicació) i es demana a l'usuari que expliqui quin és l'error que ha detectat en el contingut. Quan ho hagi fet, s'enviarà un correu indicant en quin contingut s'ha detectat l'error i de quin problema es tracta.



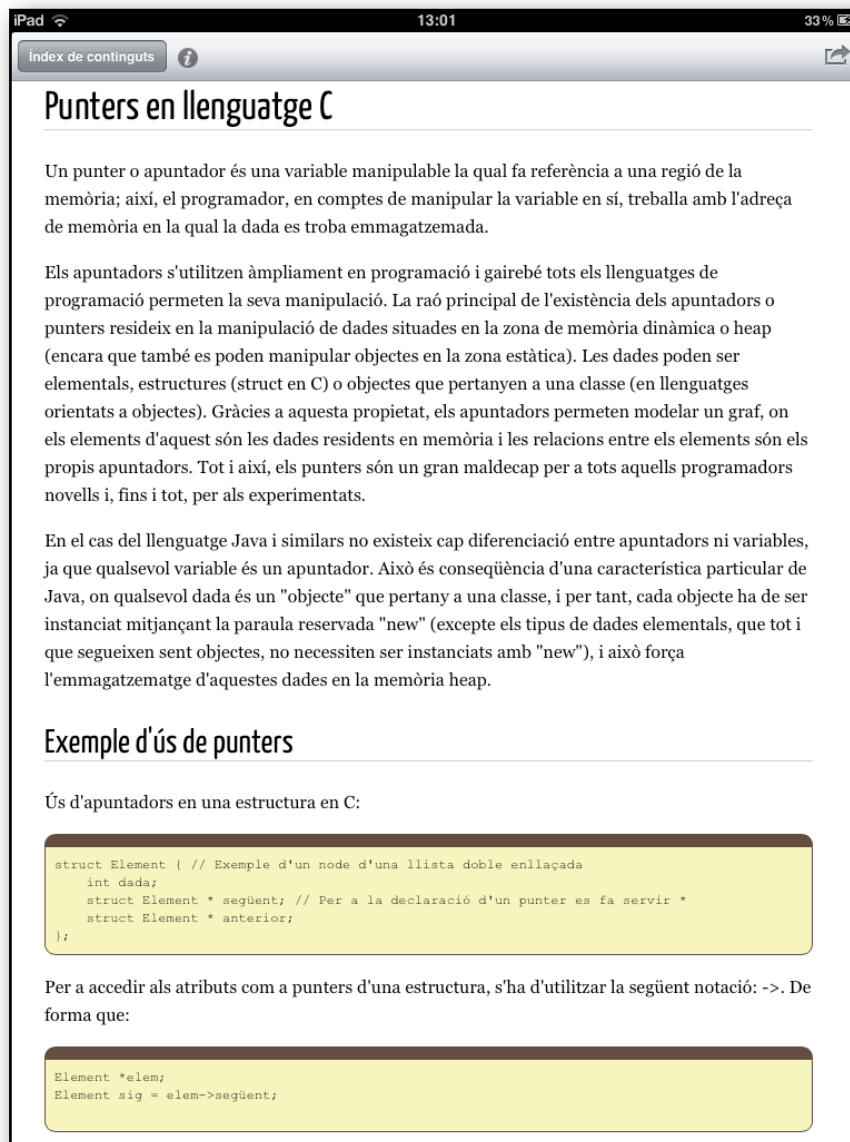
Figura 4.4.11. UOC Algorithms mostrant les accions auxiliars.



Finalment, cal destacar un cop més que l'aplicació funciona de manera automàtica tant en posició horitzontal com en posició vertical. En les imatges anteriors s'ha mostrat sempre la primera, però en la figura 4.4.12 següent es pot veure un contingut quan l'iPad està agafat de manera vertical.

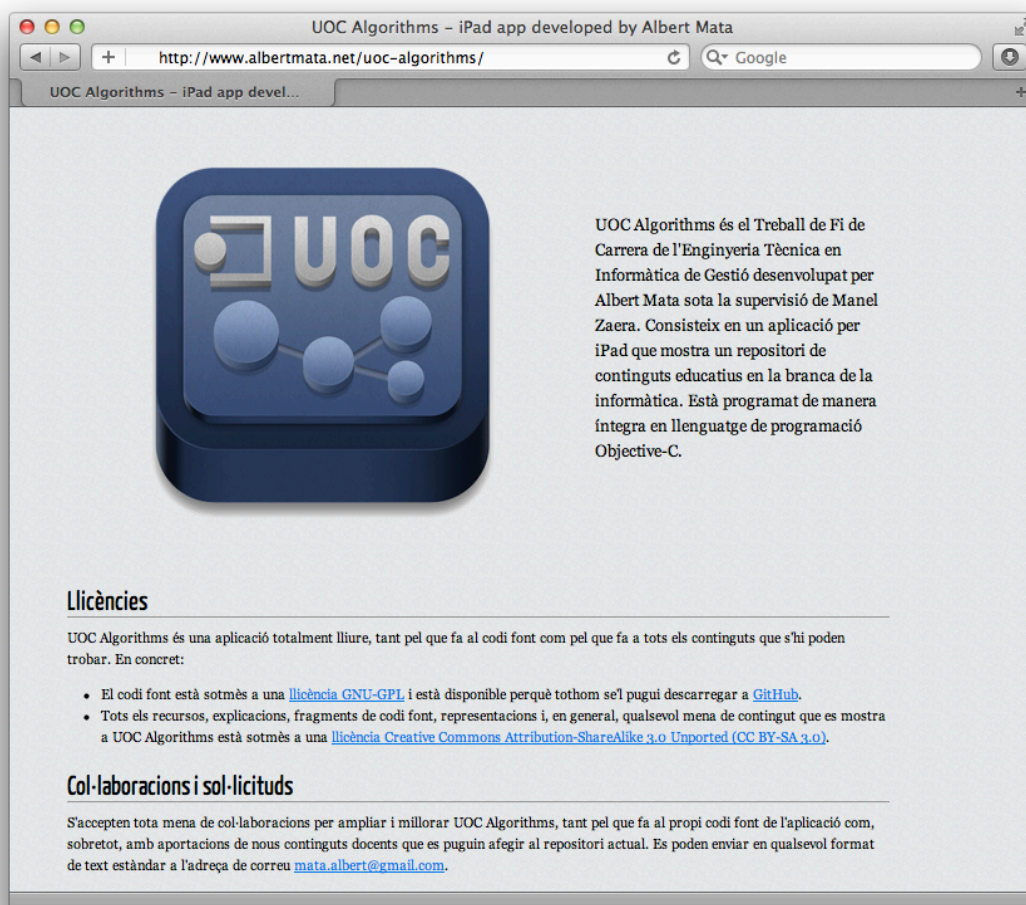
En aquesta variant l'arbre de continguts de l'esquerra desapareix per deixar tot l'espai disponible per poder-hi mostrar el detall del recurs seleccionat. No obstant, apareix un nou botó a la part superior esquerra que permet mostrar-lo per seguir utilitzant l'aplicació amb tota normalitat.

**Figura 4.4.12. UOC Algorithms amb l'iPad en posició vertical.**



Tal com s'ha comentat al parlar de la planificació d'aquest projecte, per poder publicar una aplicació a l'AppStore cal disposar també d'un lloc web que, encara que sigui senzill, permeti als usuaris conèixer qui hi ha darrera d'una aplicació i com hi poden contactar. En el cas del projecte UOC Algorithms, aquesta funció la fa el lloc web disponible a l'adreça <http://www.albertmata.net/uoc-algorithms> que es mostra en la imatge següent.

**Figura 4.4.13. Pàgina web de l'aplicació UOC Algorithms.**



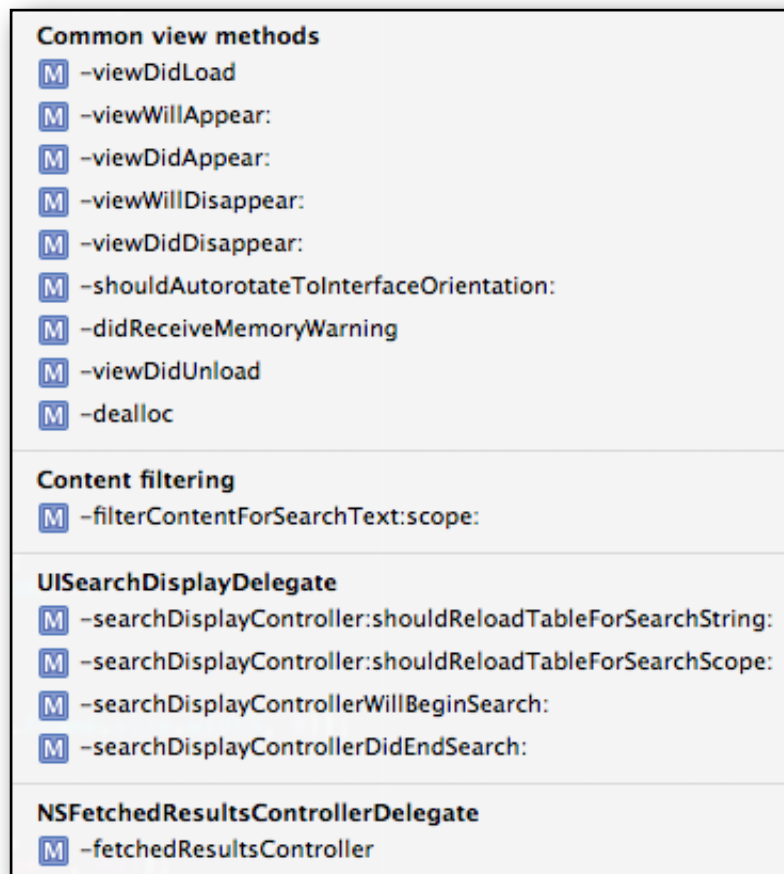
## 4.5. Capítol 5: Documentació

En tot projecte de programari la documentació del codi és una de les tasques fonamentals que cal dur a terme. Documentar s'acostuma a concebre com una manera de facilitar el seguiment del codi a altres programadors que hi tinguin accés, però més enllà d'això cal considerar-ho com un procés de comunicació amb un mateix.

Al llarg del procés de desenvolupament del projecte UOC Algorithms s'ha seguit una metodologia rigorosa de documentació i d'ordenació del codi font basada en dos aspectes principals.

Per una banda, s'han utilitzat les marques (#pragma mark) en Xcode per tal d'organitzar el codi font en blocs de mètodes que tinguin una naturalesa comuna. Això permet navegar internament pel codi de manera còmoda, com s'aprecia en la imatge següent:

**Figura 4.5.1. Exemple d'utilització d'etiquetes #pragma mark en Xcode.**



Per l'altra banda, s'han posat blocs de comentaris a tots i cadascun dels mètodes creats nous i que, per tant, són propis de l'aplicació. Aquests comentaris s'han escrit en anglès pensant en la possibilitat d'arribar al major nombre possible de col·laboradors en el futur desenvolupament de l'aplicació, atès que, com s'ha comentat, aquesta és de codi obert. En la imatge 4.5.2 es mostra un exemple d'aquests blocs de comentaris.

Amb l'aplicació d'aquests dos criteris s'aconsegueix un codi font ordenat i fàcil de seguir. S'ha decidit no generar un paquet extern de documentació de l'estil Javadoc, que en el cas de l'entorn Xcode i el llenguatge de programació Objective-C es podria fer amb eines externes com Doxygen<sup>13</sup>, perquè es considera que UOC Algorithms no és un producte acabat, sinó que estarà en permanent evolució, i es pretén evitar que hi hagi asincronia entre la darrera versió del codi font i versions anteriors de la documentació. No obstant, qualsevol programador se la pot generar per ell mateix si així ho desitja.

<sup>13</sup> Es pot trobar tota la informació sobre aquesta eina al lloc web <http://www.stack.nl/~dimitri/doxygen>.

**Figura 4.5.2. Exemple de mètode documentat.**

```
/*
 * Updates results after user has introduced some new text on search field.
 */
- (BOOL)searchDisplayController:(UISearchDisplayController *)controller
shouldReloadTableForSearchString:(NSString *)searchString
{
    [self filterContentForSearchText:searchString scope:
    [[self.searchDisplayController.searchBar scopeButtonTitles] objectAtIndex:
    [self.searchDisplayController.searchBar selectedScopeButtonIndex]];

    return YES;
}
```

## 4.6. Capítol 6: Consideracions econòmiques

Tal com s'ha comentat i argumentat al llarg d'aquest document, el plantejament econòmic i comercial del projecte ha canviat d'una primera versió en què es plantejava una llicència comercial a l'actual enfocament totalment lliure, tant a nivell de codi font com a nivell de continguts.

Aquests canvis afecten molt notablement la valoració econòmica del projecte UOC Algorithms. Si inicialment hauria calgut dur a terme estudis de costos per contingut, rendibilitat esperada, nombre de descàrregues a la botiga oficial AppStore, possibilitat de remuneració o no als creadors dels continguts, costos de programació de les actualitzacions, etc., amb el nou plantejament totes aquestes consideracions han perdut gran part de rellevància.

Tot el que fa referència al cost de desenvolupament del codi font de l'aplicació i de les futures actualitzacions es pot considerar no quantificable econòmicament, en tant que el projecte estarà disponible en un repositori públic (probablement GitHub, tot i que podria variar en el futur) obert i disponible perquè tothom qui ho vulgui pugui suggerir noves funcionalitats i participar en el seu desenvolupament. Per tant, no se n'espera un retorn econòmic directe.

A nivell de continguts, el fet que el programari utilitzi una llicència totalment lliure permet no només que tothom qui ho desitgi pugui fer ús de qualsevol de les explicacions i elements mostrats, sinó també que la pròpia aplicació es proveeixi d'un gran nombre de recursos lliures disponibles a la Xarxa. Com a conseqüència, els costos de la generació de continguts es redueixen de manera dràstica i deixen de ser un problema potencial.

El projecte UOC Algorithms amb el plantejament actual té tres aspectes principals a tenir en compte pel que fa a valoració econòmica:

- El cost anual que suposa el programa de desenvolupadors Apple que dóna dret a tenir l'aplicació publicada a l'AppStore disponible perquè tothom se la pugui descarregar. Es tracta d'un cost baix que assumeix l'autor del projecte en tant que, per necessitats professionals pròpies, també requereix formar part d'aquest programa i tenir aquesta llicència.
- El cost de mantenir, actualitzar i ampliar el codi font de l'aplicació. Com s'ha comentat, en tant que es tractarà d'un projecte de codi obert es comptarà amb la possible col·laboració desinteressada d'altres usuaris. En cas que no es produeixi, no tractant-se d'un producte comercial la pressió per actualitzar a noves versions serà menor i, per tant, no problemàtica.
- El cost d'ampliar el repositori de continguts de l'aplicació. Atès que les llicències lliures emprades faciliten l'accés a tots els repositoris d'informació també lliure existents a la Xarxa, aquest cost es reduirà a la tasca de recerca i donar format als continguts que es vulguin afegir.

En resum, cal tenir clar que UOC Algorithms està plantejat com un projecte no comercial i que, per tant, es pretén que generi uns costos mínims, tant a nivell econòmic com temporal. Aquests costos mínims, al seu temps, poden tenir un cert retorn a nivell de prestigi i visibilitat professional per l'autor.

## 4.7. Capítol 7: Conclusions

El desenvolupament d'aquest TFC i totes les seves activitats complementàries han suposat un procés d'aprenentatge eminentment pràctic que, malgrat exigir una dedicació horària més gran de la inicialment prevista, ha resultat realment molt profitós. Per una banda, ha estat possible aplicar de manera directa molts dels conceptes vistos al llarg dels estudis d'Enginyeria Tècnica en Informàtica de Gestió, concretament els més relacionats amb la programació informàtica i el desenvolupament de programari. Per l'altra, ha calgut aprendre noves eines i tecnologies pràcticament des de zero, cosa que ha fet imprescindible actuar amb rigor i eficiència si es volien respectar els terminis.

Durant aquest període s'han afrontat tres situacions, clarament relacionades amb la vessant tècnica, en les quals s'han hagut d'aplicar les habilitats adquirides durant els estudis d'informàtica a la UOC.

En primer lloc, ha estat necessari valorar diferents alternatives possibles en diversos punts. Per exemple a l'hora de decidir quina mena de tecnologies es volien fer servir. Això ha comportat dur a terme una petita anàlisi pràctica de cadascuna de les possibilitats. No obstant, cal tenir en compte que quan s'està davant d'un projecte real amb un temps acotat, com és el cas del TFC, no és possible invertir una gran quantitat de temps en l'estudi profund de totes les eines, de manera que cal cercar comparatives dutes a terme anteriorment per altres professionals i, junt amb l'opinió pròpia que es pugui derivar d'uns dies de proves en primera persona, decidir quina opció serà l'escollida.

Fent-ho d'aquesta manera s'ha decidit l'ús de la combinació de tecnologies Objective-C, Xcode, Cocoa Touch i Core Data com a principals durant la fase de desenvolupament del producte. Vist a posteriori, es considera que ha estat una bona elecció: el rendiment de l'aplicació és molt bo i ha estat possible treure el màxim profit de les característiques d'un iPad, cosa que amb una altra tria hauria estat més difícil o impossible. A més, s'ha adquirit un domini d'aquestes tecnologies que pot resultar molt útil a nivell professional.

En segon lloc, ha calgut dur a terme una immersió total en algunes tècniques que pràcticament no s'havien vist en el pla d'estudis, però que el fet de tenir una bona base acadèmica ha facilitat. És el cas, per exemple, del patró de disseny MVC, àmpliament comentat en el capítol 4. Amb l'ús d'aquest patró s'ha aconseguit una separació total entre classes de negoci i de presentació, cosa que facilita en gran mesura tant el desenvolupament com, sobretot, el posterior manteniment del projecte. Per tant, es considera que tot el temps dedicat a l'estudi i aplicació d'aquest patró de disseny ha estat ben invertit.

En darrer lloc, el fet de treballar en una nova plataforma -iOS i, en general, Mac OS X- no vista amb anterioritat ha fet necessari un esforç inicial de lectura i assimilació de moltes directrius que Apple estableix en els seus programes per a desenvolupadors. El fet de seguir-les garanteix que el producte que es genera està en línia amb els requeriments de qualitat d'Apple i que, per exemple, l'aplicació resultant es podrà posar a disposició de tots els usuaris d'aquestes plataformes. Es valora, doncs, molt positivament l'experiència d'haver entrat en una nova plataforma de desenvolupament i haver assolit els objectius marcats. Però, novament, aquest punt hauria estat molt més complicat de no disposar de la base de coneixements que suposa haver cursat els estudis d'informàtica a la UOC.

Fora ja de la vessant més tècnica, tota la part d'estudi i anàlisi de les diferents llicències que es podrien haver aplicat al projecte (privatives per una banda, i lliures amb GPL, BSD i Creative Commons per l'altra) i de les implicacions que comporta l'elecció particular de cadascuna d'elles ha resultat ser un exercici que, malgrat no tenir-lo previst inicialment, ha enriquit l'experiència personal del desenvolupament del TFC.



Cal tenir en compte, a més, que la primera vegada que es desenvolupa una aplicació per una nova plataforma és habitual voler-hi afegir massa funcionalitats simplement pel fet de conèixer, en tant que usuaris d'aquella tecnologia, les possibilitats que ofereix. Aquest factor s'ha donat al llarg del desenvolupament d'aquest projecte: inicialment s'havien previst una sèrie d'opcions, per exemple per mostrar els continguts dividits en diversos apartats, que responien més al fet de saber que allò era tècnicament possible que no pas a què aquella fos la millor manera de fer-ho pensant en l'usuari.

Aquesta manera de procedir es correspon, de fet, amb el que en enginyeria del programari es coneix com un antipatró de disseny. Ha estat necessari, per tant, fer una anàlisi crítica i reiterada al llarg del desenvolupament del projecte per tal d'intentar evitar al màxim aquests comportaments. Fruit d'aquestes iteracions han desaparegut funcionalitats inicialment previstes i n'han aparegut altres que han canviat lleugerament la naturalesa de l'aplicació, però que en milloren la usabilitat.

Amb tot, la planificació inicial s'ha acabat mostrant força acurada. És evident que hi hagut canvis i que ha calgut actuar amb flexibilitat en alguns moments, però el fet d'haver dut a terme una primera planificació preveient l'aparició de problemes i deixant petits períodes de marge ha permès que el grau d'acompliment final hagi estat total sobre els objectius marcats. Fins i tot s'han assolit alguns objectius personals no previstos inicialment, com per exemple haver desenvolupat el projecte íntegrament utilitzant un sistema de control de versions distribuït (Git, en aquest cas) amb dos repositoris: un a la pròpia màquina de desenvolupament i un altre al servei Dropbox. Aquest sistema ha funcionat realment bé i s'adoptarà com a mètode de treball en propers projectes.

Finalment, i en la línia de plantejar la viabilitat futura de l'aplicació UOC Algorithms, cal tenir ben present que el que s'ha desenvolupat només és una eina inicial, en cap cas un producte final tancat. Serà necessària, doncs, una feina constant d'incorporació de nous continguts docents durant els propers mesos si es desitja, com és el cas, que l'aplicació es converteixi en quelcom realment útil pels estudiants d'informàtica.

#### **4.7.1. Línies de treball futures**

---

Malgrat totes aquestes conclusions, és irreal pensar que aquest procés està finalitzat. Tot projecte de programari té un cicle de vida relativament llarg en el qual es van afegint funcionalitats i es van alliberant versions que les incorporen. El projecte UOC Algorithms no és cap excepció a aquest comportament, però sí que tenia marcada, des del primer moment, una data d'entrega important en què calia satisfer tota una sèrie d'objectius. Aquest TFC es correspon amb la primera versió actual del programari i, d'alguna manera,



es pot considerar que l'aplicació que s'entrega és un producte tancat i acabat, llest per ser avaluat.

No obstant, una vegada fora del marc del TFC el projecte continua i es tenen ja clares algunes de les línies de treball que es volen seguir, que són les següents:

- Incorporar als continguts la possibilitat de fer servir la tecnologia LaTeX per tal que es puguin crear recursos més rics des del punt de vista tipogràfic.
- Utilitzar alguna llibreria per remarcar de manera automàtica la sintaxi del codi font, per exemple SyntaxHighlighter<sup>14</sup> o algun producte derivat.
- Buscar les millors fonts de continguts lliures per tal d'omplir el repositori. Per la primera versió s'ha fet servir de manera pràcticament exclusiva la Viquipèdia<sup>15</sup>, però s'ha detectat que la qualitat dels seus continguts en la branca informàtica està molt lluny de la qualitat que es pretén assolir amb UOC Algorithms. Caldrà, per tant, fer una tasca important en aquest sentit.
- Permetre que les imatges dels continguts es puguin afegir quan es decideix enviar un recurs per correu electrònic des de la pròpia aplicació. Actualment això no és possible. I, de fet, no ho serà de manera fàcil i immediata, ja que la limitació és pròpia del protocol de correu electrònic. Caldrà pensar en com es pot millorar aquesta funcionalitat, potser plantejant una exportació prèvia dels continguts a un format portable, com el PDF.
- Internacionalitzar l'aplicació perquè estigui disponible en diversos idiomes a més del català: castellà i anglès seran els primers que s'hi inclouran.
- Adaptar UOC Algorithms per a altres dispositius de tipus tauleta d'altres plataformes (principalment Android, però sense perdre tampoc de vista els passos duts a terme per Microsoft en aquest terreny).

---

<sup>14</sup> Aquesta llibreria és la base de moltes altres que ofereixen funcionalitats similars, i està disponible de manera lliure a l'adreça web <http://alexgorbatchev.com/SyntaxHighlighter>.

<sup>15</sup> La Viquipèdia (<http://ca.wikipedia.org>) és la versió catalana de la coneguda Wikipedia.

## 5. Glossari

---

**Appcelerator** Tecnologia que permet el desenvolupament d'aplicacions per dispositius mòbils de diferents plataformes sense haver de programar de manera específica i individual per cadascuna d'elles.

**Apple** Empresa nord-americana de tecnologia líder en el sector que comercialitza, entre d'altres, els dispositius iPad i iPhone i els ordinadors Mac.

**AppStore** Botiga oficial on es comercialitzen totes les aplicacions que es desenvolupen per dispositius mòbils de la companyia Apple. Tot i que hi ha altres maneres, al llindar de la legalitat, d'obtenir aplicacions, l'única manera oficial de fer-ho és en aquesta botiga.

**Cocoa Touch** Interfície de programació d'aplicacions propietat d'Apple que serveix per facilitar el desenvolupament d'aplicacions per iPad i iPhone. Està formada per un conjunt de classes, funcions, procediments i mètodes principalment fent servir una tecnologia orientada a objectes.

**Core Data** Llibreria inclosa en l'entorn de desenvolupament d'aplicacions d'Apple que serveix per gestionar les relacions entre objectes i la seva representació en fitxers, permetent al desenvolupador no haver d'entrar en detalls relacionats amb bases de dades.

**CSS3** Tecnologia que permet separar l'estructura d'un document de la seva presentació. Amb l'ús d'aquesta tecnologia, s'encapsulen les qüestions relatives al format d'un document web per tal de mantenir-les totes juntes en un fitxer extern separat del document web principal.

**framework** Marc de treball que s'utilitza en el desenvolupament de programari i que inclou tot un conjunt d'eines, llibreries i recursos pel programador que li faciliten la seva tasca.

**Git** Sistema de control de versions distribuït que permet gestionar de manera eficient totes les versions del codi font d'un programari que existeixen al llarg del seu cicle de vida.

**GitHub** Lloc web on es poden emmagatzemar versions del codi font d'un programari controlades a través del sistema de control de versions distribuït Git. Es tracta, fonamentalment, d'un repositori públic de projectes de codi obert.

**HTML5** Darrera versió del llenguatge d'etiquetes HTML que permet donar a un determinat document un format web estàndard, de tal manera que es pugui visualitzar emprant qualsevol navegador existent al mercat.

**internacionalització** Procés que permet que un programari estigui disponible en diversos idiomes, pensant especialment en els textos i avisos que s'hi mostren. Sovint es representa amb l'abreviatura i18n.

**IDE** Programari utilitzat en la creació d'aplicacions consistent en un entorn de desenvolupament integrat que acostuma a incloure, com a mínim, facilitats per escriure codi font, compilar-lo i depurar-ne els errors.

**iOS** Sistema operatiu que utilitzen els dispositius mòbils desenvolupats per l'empresa Apple, com per exemple l'iPad.

**iPad** Dispositiu mòbil de tipus tauleta i amb tecnologia tàctil desenvolupat per l'empresa Apple.

**JavaScript** Llenguatge de programació àmpliament utilitzat de manera conjunta amb HTML5 i CSS3 en el desenvolupament de documents web.

**LaTeX** Sistema de composició de textos pensat especialment per a la creació de documents que incloguin fórmules matemàtiques.

**Llicència BSD** Tipus de llicència aplicable al programari que el fa totalment lliure per qualsevol tipus d'utilització, permetent fins i tot que el codi amb aquesta llicència formi part d'altres programaris no lliures. Aquest darrer tret la diferencia, per exemple, d'una llicència GPL.

**Llicència Creative Commons** Llicència aplicable a qualsevol tipus d'obra que garanteix els drets d'autor al mateix temps que permet que es pugui fer un ús lliure de l'obra. Hi ha diverses variants de la llicència que estableixen quins són exactament els usos lliures acceptats en cada cas.

**Llicència GPL** Tipus de llicència aplicable al programari que permet distribuir-lo, modificar-lo i utilitzar-lo de manera lliure, però no fer-ne un ús comercial dins d'un altre programari que no sigui lliure. Aquest darrer tret la diferencia d'una llicència BSD.

**Mac OS X** Sistema operatiu que utilitzen els ordinadors desenvolupats per l'empresa Apple.

**MVC** Patró de disseny que consisteix en separar de manera clara en una aplicació les classes de negoci i dades (Model), la interfície gràfica que es mostra a l'usuari (Vista) i la lògica de control que enllaça una part amb l'altra (Controlador).

**Objective-C** Llenguatge de programació orientat a objectes que s'utilitza pel desenvolupament d'aplicacions pels diferents sistemes operatius de l'empresa Apple. Es tracta d'un llenguatge creat sobre la base del llenguatge C, per la qual cosa permet fer ús de qualsevol codi font preexistent en aquest.

**PDF** De l'anglès *Portable Document Format*, és un format d'emmagatzematge de documents pensat perquè la seva informació es mostri de manera invariable en totes les màquines.

**persistència** Característica que permet que les dades que formen part d'un programari no desapareguin quan aquest deixa d'executar-se, sinó que s'emmagatzemin de tal manera que puguin ser recuperades la propera vegada que es fa servir el programari.

**PhoneGap** Tecnologia que permet el desenvolupament d'aplicacions per dispositius mòbils de diferents plataformes sense haver de programar de manera específica i individual per cadascuna d'elles.

**repositori** Conjunt d'informació de diversos tipus que es guarda de manera centralitzada per facilitar-ne l'accés.

**SQL** Llenguatge de consultes àmpliament utilitzat per operar amb bases de dades.

**SQLite** Sistema gestor de bases de dades que destaca pels seus requeriments mínims de recursos i per la senzilla instal·lació i funcionament en qualsevol maquinari, aspecte que el fa òptim per la seva utilització com a base de dades en dispositius mòbils.

**Xcode** IDE específic de la plataforma Mac que facilita el desenvolupament d'aplicacions per dispositius de l'empresa Apple.

## 6. Bibliografia

---

**Apple Inc.** (diversos autors) (2011). *Core Data Programming Guide* (1a. ed.). Cupertino: Apple.

**Apple Inc.** (diversos autors) (2011). *Core Data Tutorial for iOS* (1a. ed.). Cupertino: Apple.

**Apple Inc.** (diversos autors) (2011). *iOS Human Interface Guidelines* (1a. ed.). Cupertino: Apple.

**Apple Inc.** (diversos autors) (2011). *Predicate Programming Guide* (1a. ed.). Cupertino: Apple.

**Apple Inc.** (diversos autors) (2011). *Table View Programming Guide for iOS* (1a. ed.). Cupertino: Apple.

**Chacon, Scott** (2011). *Pro Git* (1a. ed.). Berkeley: Apress.

**Conway, Joe; Hillegass, Aaron** (2011). *iOS Programming: The Big Nerd Ranch Guide* (2a. ed.). Atlanta: Big Nerd Ranch.

**Dalrymple, Mark; Knaster, Scott** (2009). *Learn Objective-C on the Mac* (1a. ed.). Berkeley: Apress.

**Gruber, John** (2004, 17 de desembre). "Markdown". *Daring Fireball* [article en línia]. [Data de consulta: 10 de desembre de 2011].  
<<http://daringfireball.net/projects/markdown/>>

**Harwani, Bintu** (2011). *Core Data iOS Essentials* (1a. ed.). Olton: Packt Publishing.

**Hillegass, Aaron** (2008). *Cocoa Programming for MAC OS X* (3a. ed.). Boston: Addison Wesley Professional.

**Kaffeetsatz, Yanone** (2004). "Free Font: Yanone Kaffeetsatz". *Yanone.de* [article en línia]. [Data de consulta: 13 de desembre de 2011].  
<<http://www.yanone.de/typedesign/kaffeetsatz/>>

**Kawasaki, Guy** (2005, 30 de desembre). "The 10/20/30 Rule of PowerPoint". *How to Change the World* [article en línia]. [Data de consulta: 29 de desembre de 2011].  
<[http://blog.guykawasaki.com/2005/12/the\\_102030\\_rule.html](http://blog.guykawasaki.com/2005/12/the_102030_rule.html)>

**Kerensikova, Natasha** (2011, 22 de maig). "Libupskirt". *Natasha Kerensikova's Fossil Repositories* [article en línia]. [Data de consulta: 12 de desembre de 2011].  
<<http://fossil.instinctive.eu/libupskirt/home>>

**Kochan, Stephen** (2011). *Programming in Objective-C* (3a. ed.). Boston: Addison Wesley Professional.

**Muchow, John** (2008, 26 de novembre). "Date Formatter Examples – Take 3: Date from String". *iOS developer:tips* [article en línia]. [Data de consulta: 23 de desembre de 2011].  
<<http://iphonedevloperstips.com/cocoa/date-formatters-examples-take-3.html>>

**Wenderlich, Ray** (2010, 13 d'abril). "Core Data Tutorial: Getting Started". *Ray Wenderlich - Tutorials for iPhone / iOS Developers and Gamers* [article en línia]. [Data de consulta: 22 de novembre de 2011].  
<<http://www.raywenderlich.com/934/core-data-tutorial-getting-started>>

**Wenderlich, Ray** (2010, 15 d'abril). "Core Data Tutorial: How To Preload/Import Existing Data". *Ray Wenderlich - Tutorials for iPhone / iOS Developers and Gamers* [article en línia]. [Data de consulta: 22 de novembre de 2011].  
<<http://www.raywenderlich.com/980/core-data-tutorial-how-to-preloadimport-existing-data>>

**Wenderlich, Ray** (2010, 17 d'abril). "Core Data Tutorial: How To Use NSFetchedResultsController". *Ray Wenderlich - Tutorials for iPhone / iOS Developers and Gamers* [article en línia]. [Data de consulta: 22 de novembre de 2011].  
<<http://www.raywenderlich.com/999/core-data-tutorial-how-to-use-nsfetchedresultscontroller>>

**Wentk, Richard** (2011). *Xcode 4* (1a. ed.). Indianapolis: Wiley Publishing.

## 7. Annexos

---

Tots els annexos inclosos en aquest document poden trobar-se amb facilitat a diversos llocs de la Xarxa. No obstant, per raons de completesa s'adjunten també aquí.

### 7.1. Annex 1: Llicència d'ús de la llibreria Libupskirt 1.1

Copyright (c) 2009, Natacha Porté<sup>16</sup>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

### 7.2. Annex 2: Llicència d'ús del programari Markdown<sup>17</sup>

Markdown is free software, available under the terms of a BSD-style open source license.

LICENSE

Copyright © 2004, John Gruber

<http://daringfireball.net/>

All rights reserved.

---

<sup>16</sup> Natacha Porté i Natasha Kerensikova són la mateixa persona. En aquest document s'han respectat els noms que utilitza habitualment: el primer en la llicència i el segon en els continguts que publica al seu lloc web.

<sup>17</sup> Realment UOC Algorithms no utilitza el programari Markdown, sinó únicament el format de marques conegut com Markdown. Aquest format no té cap llicència, el que sí que en té és la implementació en Perl que l'autor de Markdown n'ha fet, però UOC Algorithms no en fa ús. De tota manera, per mantenir els crèdits de Markdown i deixar constància que també es tracta de programari lliure s'adjunta aquí la llicència.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name "Markdown" nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

### 7.3. Annex 3: Llicència d'ús de la font programari Yanone Kaffeesatz<sup>18</sup>

The SIL Open Font License (OFL) is a free and open source license specifically designed for fonts and related software based on our experience in font design and linguistic software engineering.

The main purpose is to enable a true open typographic community to spring up and grow. The OFL provides a legal framework and infrastructure for worldwide development, sharing and improvement of fonts and related software in a collaborative manner. It enables font authors to release their work under a common license that allows bundling, modification and redistribution. It encourages shared value, is not limited to any specific

---

<sup>18</sup> L'aplicació fa ús de la font tipogràfica Yanone Kaffeesatz per presentar els continguts. Aquesta font es distribueix de manera gratuïta amb una llicència lliure de tipus SIL Open Font License. Tot i que aquí es mostra la part més significativa d'aquesta llicència, al següent lloc web hi ha disponible el contingut íntegre de la mateixa: [http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&id=OFL](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL)



computing platform or environment, and can be used by other organisations or individuals.

The OFL meets the specific needs of typographic design and engineering as well as the gold standards of the FLOSS (Free/Libre and Open Source Software) community, namely the cultural values and guidelines from the FSF <sup>1</sup>, the Debian Free Software Guidelines<sup>2</sup>, as well as the Open Source Definition<sup>3</sup>. It draws inspiration from some concepts and elements found in other free and open licenses, but we believe our improvements in the specific area of fonts will make the licensing model work better than other models currently in use.