



iUOC

# TFC-Desenvolupament d'aplicacions per dispositius mòbils - iUOC

# Agraïments

Aquest treball de fi de carrera és en certa manera la culminació d'un camí que vaig iniciar a finals dels anys vuitanta i que m'ha donat moltes satisfaccions i m'ha permès viure del que més he gaudit fent. En la meua opinió, l'enginyeria informàtica ha de ser per sobre de tot una passió, i així és com l'he viscut jo i espero seguir vivint-la.

En aquest llarg camí m'he trobat amb moltes dificultats i moments en que he pensat en deixar-ho córrer, i és per això que van els meus primers agraïments per la meua dona, que ha estat al meu costat en tot moment, per la meua família que m'ha animat i encoratjat i sobre tot per al meu pare, que sempre havia desitjat viure aquest moment i que, allà on sigui, espero visqui aquest moment com jo. Per a tu papa.

# 1. Resum

El treball de fi de carrera iUOC ha estat un projecte de desenvolupament mòbil al qual s'ha intentat unir tot allò que representa el coneixement adquirit amb els estudis cursats a la UOC amb el que s'ha anat adquirint a la experiència laboral. Per això s'ha triat un projecte en el que el component principal és el desenvolupament mòbil i que es nodreix de les dades d'un servei extern, com acostuma a passar a la majoria de projectes professionals actuals.

Així mateix s'ha intentat que el projecte no for merament un mètode de mostrar les dades obtingudes si no més be un sistema en que es recullen tot una serie de dades, es processen i es presenten de forma més adequada i útil, organitzant els continguts per a permetre una utilització òptima i còmode dels mateixos.

El resultat ha estat una primer aproximació força positiva a una aplicació que pot ser d'ús generalitzat per als estudiants de la UOC i que com a objectiu final té posar-se a disposició de tots els alumnes per a poder tenir un millor accés al estudis de la UOC.

El present document s'ha dividit en sis apartats, dels quals el quart és el referent al desenvolupament de l'aplicació. En aquest quart apartat s'ha estructurat la informació de forma que s'ajusti al procés de avaluació continuada que s'ha seguit durant l'elaboració del projecte.

## 2. Taula de Continguts

<b>3. Taula de figures</b>	<b>7</b>
<b>4. Desenvolupament de l'aplicació iUOC</b>	<b>9</b>
<b>4.1. Capítol 1: Introducció</b>	<b>9</b>
4.1.1. Justificació i context	9
4.1.2. Descripció del projecte	10
4.1.3. Tecnologies utilitzades	11
4.1.4. Objectius	12
4.1.5. Planificació del projecte	13
<i>Cicle de vida</i>	13
<i>Detall de les activitats</i>	13
<i>Disparadors de cost</i>	16
4.1.6. Productes Obtinguts	16
<b>4.2. Capítol 2: Anàlisi</b>	<b>18</b>
4.2.1. Usuaris i Context d'us	18
<i>Perfils d'usuari</i>	19
<i>Anàlisi de tasques</i>	20
4.2.2. Disseny Conceptual	21
<i>Escenaris d'us</i>	21

<i>Fluxos d'interacció</i>	26
4.2.3. Prototipatge	27
<i>Quadre de comandament</i>	28
<i>Pantalla d'activitats</i>	28
<i>Pantalla de Documents</i>	29
4.2.4. Avaluació	30
<i>Informació sobre l'usuari que realitza el test</i>	30
<i>Tasques que els usuaris haurien de realitzar</i>	31
<i>Preguntes referents a les tasques</i>	31
<b>4.3. Capítol 3: Disseny</b>	<b>33</b>
4.3.1. Diagrama de classes	33
4.3.2. Detalls del sistema de accés a OAuth2	34
4.3.3. Detalls de la persistència dels continguts en base de dades	35
<i>User Defaults</i>	35
<i>Core Data</i>	36
4.3.4. Utilització dels patrons de disseny MVC, Singleton i Delegate	37
4.3.5. Tecnologies específiques utilitzades	38
<i>JSON</i>	38
<i>GCD</i>	38
4.3.6. Selecció dels continguts per a la primera versió	39
<b>4.4. Capítol 4: Implementació</b>	<b>40</b>
4.4.1. Eines utilitzades	40

4.4.2. Detalls tècnics dels controls utilitzats per a la presentació	40
<i>HorizontalDayPickerView</i>	40
<i>UITableView</i>	42
4.4.3. Accés a la OpenAPI de la UOC	43
<i>Patró de disseny triat</i>	44
<i>Registre d'usuari</i>	45
<i>Consultes de dades</i>	46
<i>Sistema de notifikacions</i>	48
4.4.4. Incorporació dels dades a la base de dades	50
4.4.5. Presentació del funcionament de l'aplicació iUOC	51
<i>Pantalla inicial</i>	52
<i>Pantalla d'Activitats</i>	53
<i>Pantalla de Documents</i>	56
4.4.6. Futures funcionalitats i millores per a iUOC	61
<b>5.Conclusions</b>	<b>62</b>
<b>6. Glossari</b>	<b>63</b>
<b>7. Bibliografia</b>	<b>66</b>
<b>8.Annexos</b>	<b>67</b>
ANNEX I	67
ANNEX II	67
ANNEX II	67

## 3. Taula de figures

Figura 4.1.1 Cicle de vida aplicat al projecte	13
Figura 4.1.2 Definició de tasques de planificació del projecte	14
Figura 4.1.3 Definició de tasques de anàlisi de requeriments i funcionals	15
Figura 4.1.4 Definició de tasques d'implementació	15
Figura 4.1.5 Definició de tasques de finalització, redacció de la memòria i entrega	16
Figura 4.2.1 Diagrama de seqüència del procés d'OAuth2	22
Figura 4.2.2 Escenari 1: Primera connexió d'usuari des de l'aplicació	23
Figura 4.2.3 Escenari 2: Obtenir les dades de les assignatures	24
Figura 4.2.4 Escenari 3: Consultar la documentació d'una assignatura	25
Figura 4.2.5 Escenari 4: Actualitzar la llista d'activitats	26
Figura 4.2.6 Fluxos d'interacció	27
Figura 4.2.7 Prototipus de la pantalla principal	29
Figura 4.2.8 Prototipus de la pantalla de documents	30
Figura 4.3.1 Diagrama de classes	33
Figura 4.3.2 Stack de Core Data	36
Figura 4.3.3 Esquema del patró MVC	37
Figura 4.4.1 Control UIDatePicker	41
Figura 4.4.2 Control HorizontalDayPickerView	41
Figura 4.4.3 Esquema de creació del copntrol HorizontalDayPickerView	42
Figura 4.4.4 Cel·la adaptada per a la presentació d'activitats	43
Figura 4.4.5 Taula adaptada per a presentar la llista d'activitats	43

Figura 4.4.6 Sistema de notifikacions d'iOS	49
Figura 4.4.7 Pantalla inicial d'iUOC	52
Figura 4.4.8 Pantalla d'autoritzacions d'iUOC	53
Figura 4.4.9 Pantalla d'activitats d'iUOC	54
Figura 4.4.10 Pantalla de detall d'una activitat d'iUOC	55
Figura 4.4.11 Pantalla de documents d'iUOC	56
Figura 4.4.12 Pantalla de lectura de document d'iUOC	57
Figura 4.4.13 Pantalla d'activitats d'iUOC en posició vertical	58
Figura 4.4.14 Pantalla de documents d'iUOC en posició vertical	59
Figura 4.4.15 Pantalla de lectura de document d'iUOC en posició vertical	60



## 4. Desenvolupament de l'aplicació iUOC

El procés seguit per a desenvolupar l'aplicació iUOC ha estat un procés en certa manera particular donat que un sol professional ha estat l'encarregat de realitzar totes les tasques implicades a la concepció, disseny i desenvolupament de l'aplicació. Aquest fet ha provocat en certa manera que les correccions sobre les planificacions inicials, hagin estat més freqüents del que en un equip haguessin estat. Tot i això, s'ha seguit un esquema de plantejament de una solució per a un problema, anàlisi de la solució proposada i finalment desenvolupament de la solució. Aquests son els apartats que es presenten en aquest apartat.

### 4.1. Capítol 1: Introducció

Al present capítol, introducció, es presenten les primeres conclusions, idees i aproximacions que es fan per a proposar una de les moltes possibles solucions a la necessitat que es planteja. En aquesta primera aproximació no es pretén entrar tant en aspectes concrets d'implementació o models concrets com a aspectes com justificació i motivació del projecte, mètodes a seguir per a la implementació o planificació del projecte.

#### 4.1.1. Justificació i context

Durant el transcurs dels estudis a la *UOC*, en el meu cas i penso que en el de molts altres, hi ha molts moments en els que l'estudiant o be no es troba davant un ordinador o l'ordinador del que disposa no és el seu dispositiu personal, és el de la feina o simplement no és adient per a accedir a la *UOC*. Durant aquests moments així com durant els períodes de desplaçament, l'estudiant pot disposar de temps per a estudiar, gestionar les seves activitats a la *UOC* o realitzar diferents tasques que actualment no pot realitzar.

Donat que la difusió de les tauletes, així com les seves capacitats, cada cop és més gran, una aplicació mòbil pot ser de gran utilitat per a aprofitar moments que no es disposa d'un ordinador o de l'espai necessari per a fer servir el material imprès.

Degut a que el projecte es desenvoluparà per a dispositius iOS i a que és un programari per a accedir al material de la *UOC* he decidit jugar amb ambdues paraules i he decidit anomenar a l'aplicació i al projecte per extensió iUOC doncs penso que és una paraula que per a si mateixa pot resultar força aclaridora de quines son les funcionalitats i el suport de la mateixa.

L'aplicació iUOC és una aplicació per a dispositius iPad amb el sistema operatiu iOS 6 o superior que permet consultar el calendari d'activitats de les diferents assignatures de les que l'alumne és matriculat, realitzar anotacions sobre les activitats i consultar el material de les diferents aules de que es forma part.

El fet de limitar el dispositiu a l'iPad es deu a que, tot i ser possible, no he considerat adequada la mesura de la pantalla d'un iPhone per a la lectura de forma intensiva de documentació.

Com a aspectes més destacats de l'iUOC és interessant mencionar que obté les dades de l'**OpenAPI** de la UOC i que per a fer-ho accedeix mitjançant el protocol OAuth2 per a autenticar-se. Així mateix és important destacar que la versió actual necessita que es disposi de connexió a Internet durant la execució, donat que obté la informació de la UOC no emmagatzemant les dades tret de les anotacions realitzades a les activitats per part de l'estudiant.

L'objectiu final és posar a disposició de tota la comunitat d'estudiants, de forma gratuïta, l'aplicació per a facilitar en la mesura del que sigui possible estudiar a la **UOC**.

#### 4.1.2. Descripció del projecte

El present apartat té com a finalitat realitzar l'anàlisi i posterior implementació d'una aplicació per a dispositius mòbils per a la gestió d'activitats i documentació de les assignatures en que estigui matriculat un alumne en un moment donat. El projecte es desenvoluparà en Cocoa, framework per al desenvolupament d'aplicacions per a dispositius iOS, per a dispositius iPad amb la versió 6 de iOS o posterior instal·lada.

Els subsistemes implicats en el projecte son els següents:

**Subsistema de connexió i autoritzacions:** La connexió i validació de identitat per a fer servir la **OpenAPI** de la UOC es fa mitjançant un protocol anomenat OAuth2. Per a fer-la servir serà necessari disposar d'un registre previ de l'aplicació i la obtenció d'una clau única que permetrà l'accés als recursos de la api. Aquest procés es realitza de forma manual accedint a la plana de la OpenAPI<sup>1</sup> que permet sol·licitar les dades per a una aplicació concreta.

**Subsistema de gestió de documents:** Una de les funcionalitats que ha de proporcionar la App és la de permetre obtenir la llista dels documents relacionats amb les aules a les que l'alumne estigui assignat i la descàrrega dels mateixos per a poder realitzar la visualització.

**Subsistema d'Activitats:** El sistema ha de ser capaç de connectar-se i obtenir el llistat de les diferents activitats de les diferents assignatures a les que està matriculat un alumne en un

---

<sup>1</sup> <http://blogs1.uoc.es/developer/documentacio/api/>

moment donat, realitzar el processament per a agrupar-les, mostrar-les i permetre realitzar anotacions vinculades a les activitats.

El desenvolupament es realitzarà amb un ordinador portàtil MacBook Pro 15 y un iPad 32Gb de tercera generació. Així mateix es farà servir el entorn de desenvolupament XCode versió 4.5.

L'usuari final destinatari de l'aplicació és qualsevol estudiant de la UOC que disposi d'un dispositiu iPad capaç de córrer el sistema operatiu iOS 6 o posterior. Per a posar-ho a disposició dels usuaris, l'aplicació s'haurà d'enviar a l'App Store d'Apple i un cop aprovat estarà a disposició de tots els estudiants.

### 4.1.3. Tecnologies utilitzades

Per al desenvolupament del projecte s'ha decidit utilitzar tot un seguit de tecnologies que son extensament utilitzades i en molts casos son estàndards reals o estàndards de facto. En cada cas és positiu per al projecte una petita justificació de per què s'ha decidit utilitzar-les i quines son les avantatges i inconvenients que tenen.

Les tecnologies utilitzades son:

- **MultimarkDown**<sup>2</sup>: Segons la definició de la Wikipedia, Markdown és un llenguatge de marcat lleuger [...] que tracta d'aconseguir la màxima llegibilitat y "publicabilitat" tant en les seves formes de entrada com de sortida, Markdown és un estàndard actualment per a la realització de documentació donada la seva llegibilitat, la seva portabilitat i el fet de que és possible de editar i visualitzar amb qualsevol editor de texts, donat que no deixa de ser un fitxer de text pla. Per a documentació dins del projecte i dins els propis fitxers de codi, es farà servir **Markdown** com sintaxi per a facilitar la llegibilitat. Aquesta sintaxi, però, queda una mica curta en el moment de realitzar tota la documentació del projecte i la memòria, és per això que he decidit fer servir un superset de Markdown anomenat **MultimarkDown** que permet afegir referències, bibliografies, notes a peu de pàgina i moltes altres opcions que son necessàries per a un document d'aquest tipus.
- **OCUnit**<sup>3</sup>: Per a realitzar desenvolupaments que segueixin un esquema d'integració continua es fa requisit imprescindible realitzar tests unitaris del codi desenvolupat, que permetin assegurar la integritat dels requeriments en tot moment. Una pràctica força recomanable per a desenvolupar un codi testeable es fer servir la tècnica de desenvolupament TDD<sup>4</sup>(*Test*

---

<sup>2</sup> Podeu accedir la sintaxi i documentació a <http://fletcher.github.com/peg-multimarkdown/mmd-manual.pdf>

<sup>3</sup> Podeu veure la documentació [http://developer.apple.com/library/ios/#documentation/DeveloperTools/Conceptual/UnitTesting/00-About\\_Unit\\_Testing/about.html](http://developer.apple.com/library/ios/#documentation/DeveloperTools/Conceptual/UnitTesting/00-About_Unit_Testing/about.html) per a més informació

<sup>4</sup> Una bona referència per a iniciar-se en TDD és el llibre de Graham Lee, *\*Test-Driven iOS Development (Developer's Library)\**. Addison-Wesley Professional, 2012.

*Driven Development*, en català *Desenvolupament Orientat als Tests*). Aquesta tècnica serà la emprada principalment durant el desenvolupament del model de dades, doncs és on resulta més intuïtiu i útil segons el meu parer, a més de proporcionar la inestimable avantatge que resulta de la detecció i identificació de l'origen dels errors, inconsistències i incompliments dels requeriments. Desenvolupar utilitzant *TDD* significa seguir un procés segons el qual en primer lloc s'han d'escriure els tests que representen els requeriments funcionals de les classes a desenvolupar. Donat que inicialment aquestes classes ni tant sol existiran els tests donaran error, i és solucionant aquests errors que s'anirà desenvolupant la classe.

- **OAuth2<sup>5</sup>** : Segons la wikipedia OAuth és un estàndard obert per a autoritzacions. Permet als usuaris compartir els seus recursos privats, emmagatzemats a una localització, amb un altre localització sense tenir que gestionar les seves credencials. És una tecnologia que cada cop és més utilitzada degut a la seguretat que proporciona y al control sobre els continguts. Per a accedir als recursos de la UOC mitjançant la API hem de fer servir el protocol OAuth 2 que és una evolució del propi OAuth. Donat que el protocol per a la implementació de OAuth2 a la *OpenAPI* de la UOC no és una implementació 100% estàndard i que al dia de la implementació del present projecte no hi ha cap llibreria d'accés creada per a iOS específica, no és possible utilitzar llibreries de propòsit general per a accedir-hi i s'ha d'implementar unes llibreries específiques per a accedir-hi .

#### 4.1.4. Objectius

Els objectius que es pretén posar en pràctica amb aquest projecte son els següents:

- Posar en pràctica els coneixements obtinguts durant la carrera pel que fa a la planificació, especificació i desenvolupament de projectes de programari.
- Aprendre a afegir un grau superior de rigorositat i teoria als projectes desenvolupats al meu entorn professional realitzant el present projecte.
- Aprofundir en el coneixement de les tecnologies d'accés a recursos en el núvol com ara OAuth2.
- Aplicar patrons de disseny com ara MVC i Singleton entre d'altres així com programació concurrent i gestió complexa de threads.
- Obtenir dades reals del desviament produït entre la planificació del projecte i el resultat final, tant pel que fa a temps com per el que fa a costos i continguts.

---

<sup>5</sup> Per a una definició rigorosa de OAuth 2 podeu accedir a <http://oauth.net/2/>

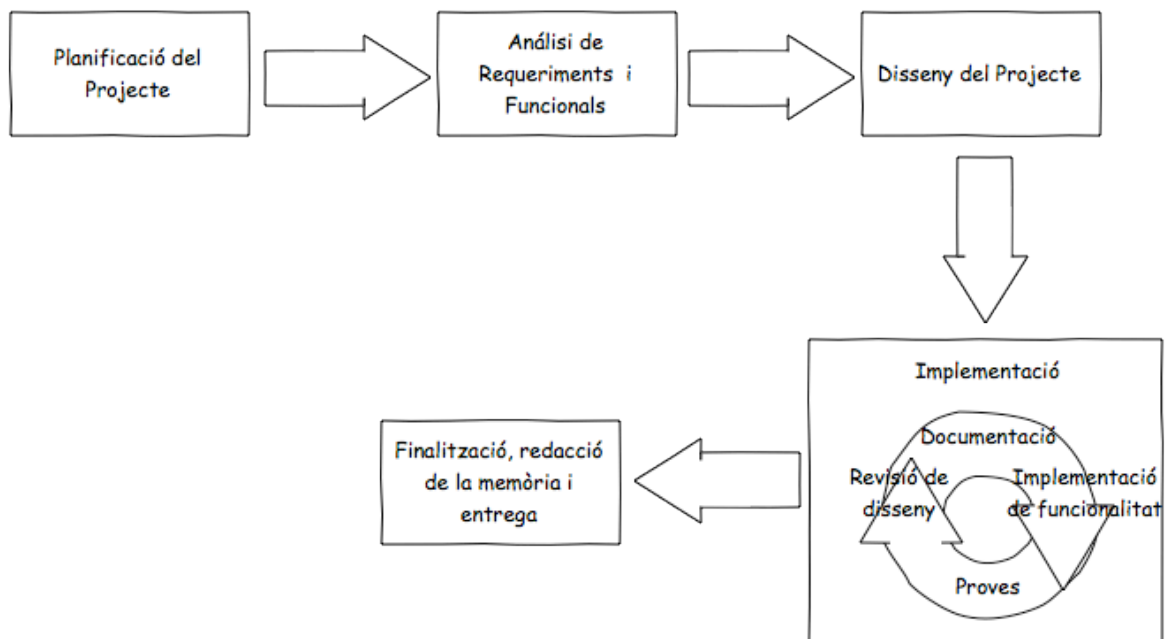
- Crear un projecte que sigui d'utilitat per a tots els estudiants de la UOC, que sigui evolucionat, que permeti afegir funcionalitats futures i que pugui servir de referència de futur per a nous TFC en mobilitat.

#### 4.1.5. Planificació del projecte

##### Cicle de vida

Per a la realització del projecte s'ha triat un cicle de vida que permeti adaptar les funcionalitats a les possibilitats temporals i tècniques del projecte. Per a fer-ho s'ha optat per seguir un model híbrid de cicle de vida clàssic i desenvolupament àgil en el qual s'han de revisar les especificacions i el disseny a cada iteració o desenvolupament de funcionalitat de forma que és possible re definir el disseny parcialment i s'ha de documentar cada canvi de forma que el resultat sigui un reflex fidel del projecte desenvolupat. La figura 4.1.1 presenta de forma esquemàtica com s'estructura aquest cicle de vida.

Figura 4.1.1 Cicle de vida aplicat al projecte



##### Detall de les activitats

La primera etapa és la de Planificació del projecte, a la qual es pretén establir les línies bàsiques del projecte, fer una declaració d'intencions i uns plans de treball, que posteriorment s'aniran concretant i ens permetran avaluar la desviació experimentada entre la planificació i el resultat del projecte.

**Figura 4.1.2 Definició de tasques de planificació del projecte**

Planificació del Projecte	
Tasca	Descripció
Selecció del projecte	Tria entre les diferents opcions de projecte del que s'ha de realitzar
Preparació del projecte	Lectura del material i selecció del programari a utilitzar
Definició del projecte	Selecció de funcionalitats, tecnologies i requeriments necessaris
Planificació del projecte	Descripció del cicle de vida, definició de tasques, temporització de tasques i principals fites del projecte
Redacció del document	Creació i format del document de planificació del projecte

La segona etapa, **Anàlisi de Requeriments i Funcionals**, és on hem d'entrar en detall als diferents requeriments i funcionalitats definides per al projecte i crear un disseny que ha de resultar el més definitiu possible. Hi ha que ser flexible per a modificar el disseny en certs aspectes donat que hi ha un cert grau d'incertesa i desconeixement respecte a la **OpenAPI** de la UOC, però ha de representar, en essència, el resultat final del projecte. Per a realitzar aquesta etapa es farà servir la metodologia de *Disseny Centrat en l'Usuari (DCU)* que s'enfoca principalment en l'usuari, l'usabilitat i la concepció des d'un punt de vista pràctic més que teòric del projecte. Les tasques a realitzar en aquesta fase son:

**Figura 4.1.3 Definició de tasques de anàlisi de requeriments i funcionals**

Anàlisi de Requeriments i Funcionals	
Tasca	Descripció
Anàlisi d'usuaris i context d'ús	Realitzar un anàlisi, mitjançant diferents tècniques de quins son els possibles usuaris de l'aplicació i quin o quins son els possibles contextos d'ús
Disseny conceptual	Disseny dels diferents casos d'ús i diagrames de seqüència
Fluxos d'interacció	Definició dels diferents fluxos d'interacció de l'aplicació des d'un punt de vista de l'usuari.
Prototipatge	Disseny dels prototips de les pantalles de l'aplicació.
Avaluació	Definició de les metodologies per a la obtenció de dades per part dels possibles usuaris de l'aplicació.

Anàlisi de Requeriments i Funcionals	
Redacció del document	Creació i format del document de Anàlisi de Requeriments i Funcionals i Disseny

La etapa de **Implementació** és la que, per la naturalesa de les metodologies que s'utilitzaran, resulta més difícil de valorar temporalment i és on pot haver-hi una major desviació respecte a la planificació, no obstant això, la relació de tasques a realitzar és clara.

**Figura 4.1.4 Definició de tasques d'implementació**

Implementació	
Tasca	Descripció
Subsistema de Connexió	<ul style="list-style-type: none"> <li>• Disseny de la llibreria d'Accés OAuth</li> <li>• Desenvolupament del component de credencials d'usuari</li> <li>• Desenvolupament del component de gestió de la connexió</li> <li>• Desenvolupament de la interfície d'usuari de connexió</li> <li>• Desenvolupament del sistema de descàrrega de dades</li> </ul>
Subsistema de gestió de Documents	<ul style="list-style-type: none"> <li>• Desenvolupament del component de processament de documents</li> <li>• Desenvolupament del component de presentació de documents</li> <li>• Desenvolupament del component de descarrega i visualització de documents</li> <li>• Desenvolupament de la interfície d'usuari de documents</li> </ul>
Subsistema d'Activitats	<ul style="list-style-type: none"> <li>• Desenvolupament del component de processament de activitats</li> <li>• Desenvolupament del component de gestió de calendari</li> <li>• Desenvolupament del component de descàrrega d'activitats</li> <li>• Desenvolupament de la interfície d'usuari d'activitats</li> </ul>
Aplicació	<ul style="list-style-type: none"> <li>• Creació dels projecte principal i els documents associats</li> <li>• Creació del sistema d'integració contínua</li> <li>• Proves finals i tests unitaris</li> </ul>

La darrera etapa del projecte, **Finalització, redacció de la memòria i entrega**, és on es realitzarà una revisió de tota la documentació generada, es revisarà, unirà i integrarà dins el document final per a crear la memòria. Serà en aquesta fase on es podrà realitzar una avaluació de les desviacions produïdes respecte a la planificació inicial. Així mateix es crearà la presentació virtual en format de screencast amb el simulador de dispositiu proporcionat per el sistema de desenvolupament com a dispositiu per a la presentació. Les tasques involucrades son:

**Figura 4.1.5 Definició de tasques de finalització, redacció de la memòria i entrega**

Finalització, redacció de la memòria i entrega	
Tasca	Descripció
Redacció de la memòria final	Revisió i format de documents i elaboració i redacció de la memòria final
Creació de la presentació virtual	Creació i gravació de la presentació virtual
Entrega final	Entrega de tots els documents generats

### Disparadors de cost

De les diferents tasques relacionades anteriorment, hi ha una que es presenta com a disparador de cost donada la incertesa que hi ha sobre la *OpenAPI* donat que no hi ha cap llibreria d'accés creada per a iOS i per tant s'haurà d'implementar una. Aquesta tasca és crítica per la impossibilitat d'avaluar inicialment el cost per el desconeixement de l'interfície i per el fet de que és implicat un desenvolupament extern amb la conseqüent dependència de terceres parts en el proces d'implementació.

### 4.1.6. Productes Obtinguts

Del present TFC s'obtindrà un únic producte que és una aplicació per a dispositius iPad amb iOS 6 o superior a banda de la present memòria del projecte i les presentacions. Es tracta d'una aplicació que permetrà connectar-se i desconnectar-se de la *OpenAPI* de la **UOC** i obtenir d'aquesta les dades corresponents a les diferents assignatures de les que està matriculat un alumne en un moment donat.

Per a la realització de la aplicació s'ha posat especial èmfasi en una interfície d'usuari neta i senzilla, on el principal focus es centrés a la claredat i facilitat d'ús. Per a aconseguir-ho s'ha optat per crear controls no estàndards que permetessin funcionalitats que d'altre manera serien més complexes d'obtenir o d'entendre.



Donat que és una aplicació per a dispositius amb iOS no és possible entregar un executable a l'us com en altres plataformes ja que Apple vincula l'execució de les aplicacions a mesures de seguretat que requereixen registrar els dispositius que poden executar les aplicacions bé des de el entorn de desenvolupament, be des de l'Apple Store. Tot i això l'aplicació es posarà a disposició dels estudiants un cop presentat el TFC.

Malgrat no entregar un executable, es mostraran diferents captures de pantalla i descripcions acurades de les diferents pantalles i funcionalitats del sistema al llarg del present document a l'apartat d'implementació obtinguts de l'execució en el simulador de iPad així com captures reals al dispositiu.

## 4.2. Capítol 2: Anàlisi

### 4.2.1. Usuaris i Context d'ús

Per a determinar els usuaris i context d'ús s'ha optat per utilitzar la tècnica de enquestes en format electrònic. Tenint en compte que és una tècnica quantitativa, s'ha procurat fer arribar al màxim d'usuaris possible obtenint una mostra final de catorze usuaris de diferents entorns i diferents experiències.

Amb la finalitat de determinar el perfil d'usuari, context d'ús i l'anàlisi de tasques s'han estructurat les preguntes de l'enquesta en diferents grups amb la finalitat de obtenir aquesta mostra i poder així determinar quins son els grups, contexts i tasques representades per la mostra.

L'estructura i grup de les preguntes realitzades son els següents:

- **General.** Aquest grup és el destinat a establir el perfil d'usuari i el context.
  - *Ets o has estat alumne de la UOC.* L'objectiu d'aquesta pregunta es determina si s'ha trobat o no en front de les necessitats plantejades de forma real.
  - *Actualment estas cursant estudia en algun altre centre.* Determinant si és alumne actualment podem establir un cert nivell d'implicació donat que pot representar un avantatge per a l'estudiant disposar d'un mètode més per a l'estudi.
  - *En el moment d'estudiar, consideres important poder-ho fer a qualsevol lloc.* Establirem un perfil favorable o no a poder fer servir la eina.
  - *Podries indicar algun avantatge a tenir accés al material d'estudi en una tauleta electrònica.* El fet d'obtenir suggerències respecte a com fer ús de la tauleta indica que o bé en disposa o bé s'ho està plantejant. A més ens permet obtenir una indicació directe del que l'usuari vol.
  - *El fet de disposar d'una aplicació mòbil d'accés a la universitat, en quin grau et motivaria a iniciar els estudis? (1 - poc, 5 - molt).* Una pregunta directa per establir el nivell d'acceptació d'una aplicació o fins i tot del dispositiu.
- **Plà Docent, Documents, Activitats.** Aquest grup és destinat a avaluar quines de les funcionalitats que pensa oferir l'aplicació son més valorades i en quins sentits.
  - Es important disposar de la funcionalitat en qüestió.
  - Es important disposar de la funcionalitat quan no es disposa de connexió.
  - Es important poder realitzar anotacions per gestionar la funcionalitat.
- **Interfície d'usuari.** Aquest grup, més que per a obtenir informació dels desitjos o preferències de l'usuari, és destinat a determinar com de familiaritzat està amb les aplicacions

per a aquest dispositiu i si coneix les darreres tendències, com ara els menús amagats que es fan visibles amb un gest o l'aspecte natural dels fulls de lectura.

El programari triat per a fer l'enquesta és LimeSurvey<sup>6</sup>. Les raons per triar-ho han estat bàsicament la seva senzillesa d'ús, el fet de que es tracta de programari lliure, que es pot publicar a una plana web sense massa esforç ni configuració i que disposa de diferents eines d'exportació que ens resulten valuoses per a fer el posterior anàlisi.

Per a la creació del formulari és necessari realitzar la instal·lació de LimeSurvey a un allotjament web, en aquest cas s'ha optat per allotjar-ho al domini [www.bytesPotion.com](http://www.bytesPotion.com) degut a que és un domini disponible. L'instal·lació s'ha realitzat utilitzant el software SimpleScripts<sup>7</sup> que permet realitzar instal·lacions de diferents productes de forma senzilla.

El format resultant del formulari es pot consultar com a ANNEX I.

El resultat de les enquestes en forma de percentatges i gràfic es pot consultar a l'ANNEX II.

L'anàlisi del resultat de les enquestes ofereix els resultats següents.

### **Perfils d'usuari**

La mostra s'ha realitzar entre un nombre d'usuaris que han estat alumnes de la UOC i que no ho han estat aproximadament igual, la qual cosa ens dona una mostra equilibrada en aquest sentit, de la mateixa forma que el 50% dels usuaris son actualment alumnes de la UOC. Tots els usuaris consideren important poder estudiar en qualsevol lloc.

Les respostes obtingudes quan es demana enumerar avantatges de poder disposar del material d'estudi en una tauleta electrònica, repeteixen principalment les paraules *flexibilitat*, *mobilitat* i *disponibilitat* com a trets més importants.

Respecte a la pregunta de si disposar de una aplicació mòbil et motivaria a estudiar a la UOC, obtenim que un 57% es sentiria motivat contra un 21% que no s'hi sentiria.

Ens trobem amb un conjunt d'usuaris tan estudiants com no de la UOC, que coneixen i consideren important poder disposar d'una eina per a estudiar en entorns no fixes i que valoren aspectes com la flexibilitat, mobilitat i disponibilitat al qual el fet de tenir una aplicació mòbil per a estudiar li representaria un al·licient en molts dels casos per a posar-se a estudiar.

<sup>6</sup> <http://www.limesurvey.org>

<sup>7</sup> <http://www.simplescripts.com>

## Anàlisi de tasques

L'anàlisi de les diferents funcionalitats de forma individual per cadascuna d'elles és el següent.

El **Plà Docent** és important per a la pràctica totalitat dels usuaris, que considera en la seva majoria que ha de ser consultable quan no es disposa de connexió i que en el mateix percentatge considera que és important poder fer-hi anotacions. Respecte a les consideracions particulars obtenim sorprenentment dues respostes totalment oposades. Una indicant que és important disposar de tota la informació i l'altra que ens indica que només el consulta a l'inici del curs, per tant no podem obtenir informació rellevant d'aquest camp.

La **Documentació de l'assignatura** és igualment important per a la immensa majoria, encara que sorprenentment de forma menys contundent que el plà docent, en canvi pel que fa a la necessitat de disposar de la informació de l'assignatura quan no es té connexió es repeteix el patró que em observat en l'anterior pregunta i a la resposta equivalent del plà docent.

Novament observem un fet curiós doncs el nombre de respostes molt positives a la pregunta de la necessitat o no de fer anotacions a la documentació, tot i que molt clarament positiva, ho és menys, de nou, que a l'equivalent del plà docent.

Respecte a la necessitat o no de poder exportar els documents amb les anotacions, només la meitat dels enquestats ho consideren realment important i respecte als mitjans més sol·licitats per a l'exportació resulten ser per correu i a Dropbox, essent però el correu el més demandat tot i que pot ser per el desconeixement del servei Dropbox per part d'alguns usuaris.

Hi ha una única resposta en altres mitjans que correspon a "xarxes socials" i que es descarta per improcedent.

Com a suggerències en aquest cas es poden destacar tres aspectes, que la documentació estigui actualitzada, es pugui realitzar anotacions i correccions i la seva disponibilitat.

Pel que fa a les **Activitats de les assignatures**, hi ha una tendència molt clara a sol·licitar que es faci servir el calendari del sistema, que les notificacions d'avis de les activitats siguin diàries i que es pugui anotar el nivell de progrés de les activitats, la qual cosa ho fa en certa mesura incompatible amb fer servir el calendari del sistema.

Com a altres consideracions es sol·licita poder realitzar les activitats a la tauleta, però aquesta funcionalitat queda completament fora de l'abast del present projecte.

Pel que fa a l'interfície d'usuari proposada, les respostes no son conclouents en cap cas doncs estan totes les respostes equilibrades de forma que no podem extraure cap conclusió.

Podem concloure doncs que es considera important disposar totes les funcionalitats proposades, disposar de la possibilitat de consultar-les quan no es disposa de connexió i de poder fer anotacions i exportar-les principalment per correu. Les activitats es prefereix que formin part del calendari del sistema i poder marcar el progrés de les mateixes.

En tots els casos és destacable la importància que es dona a disposar de dades actualitzades en tot moment.

### 4.2.2. Disseny Conceptual

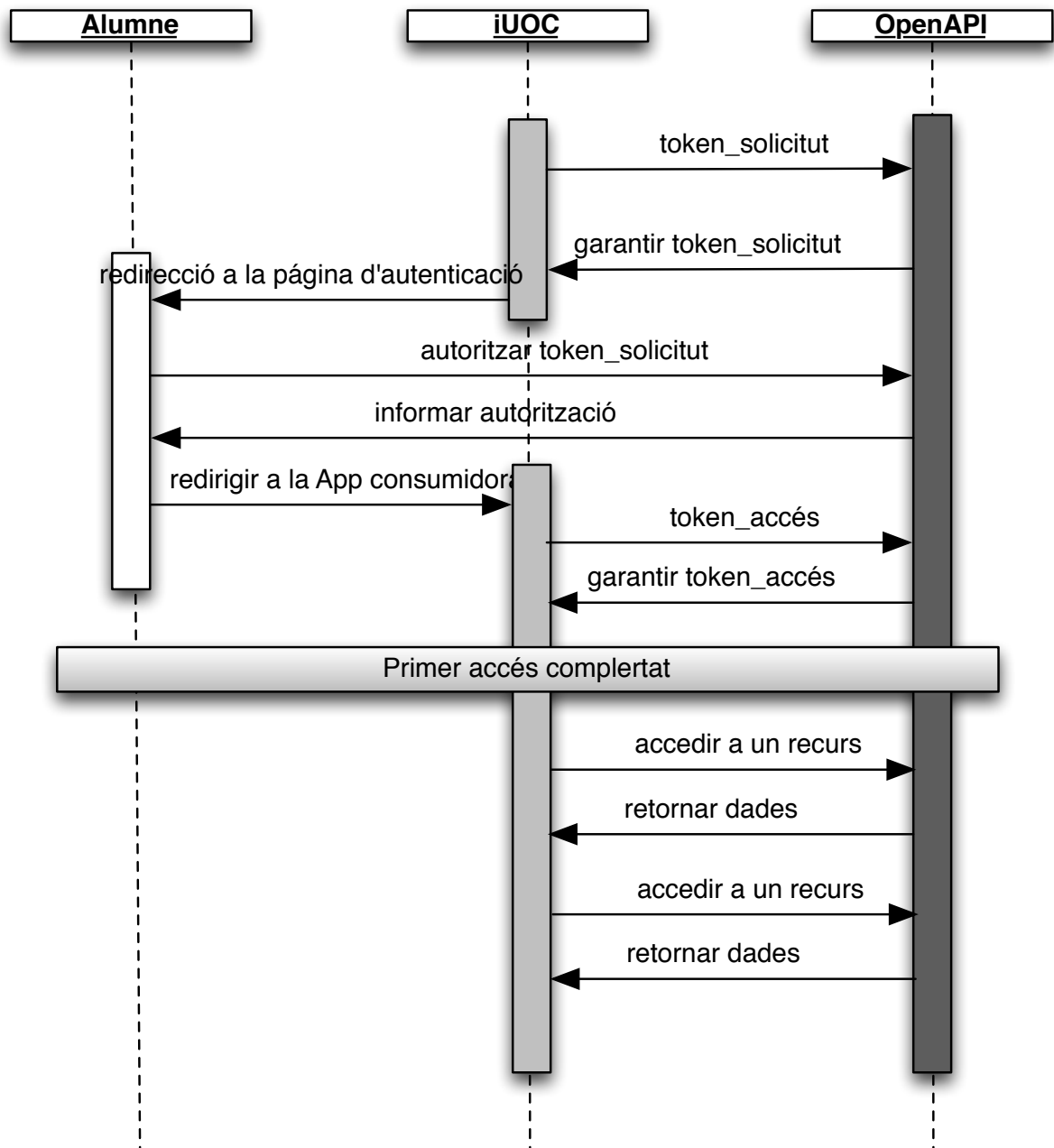
Per al disseny conceptual es pren la determinació de no considerar el perfil d'usuari donat que en tots els casos es tracta del perfil Alumne de la UOC i per tant considere reiteratiu i innecessari afegir una dada que no aportarà cap informació nova en tots els escenaris.

#### Escenaris d'ús

Els escenaris d'ús, descriuen les diferents interaccions de l'usuari amb el sistema. Es presentaran doncs els diferents casos d'ús o escenaris des d'un punt de vista de interacció de l'usuari doncs el DCU fa èmfasi en aquest mode d'aproximació al disseny.

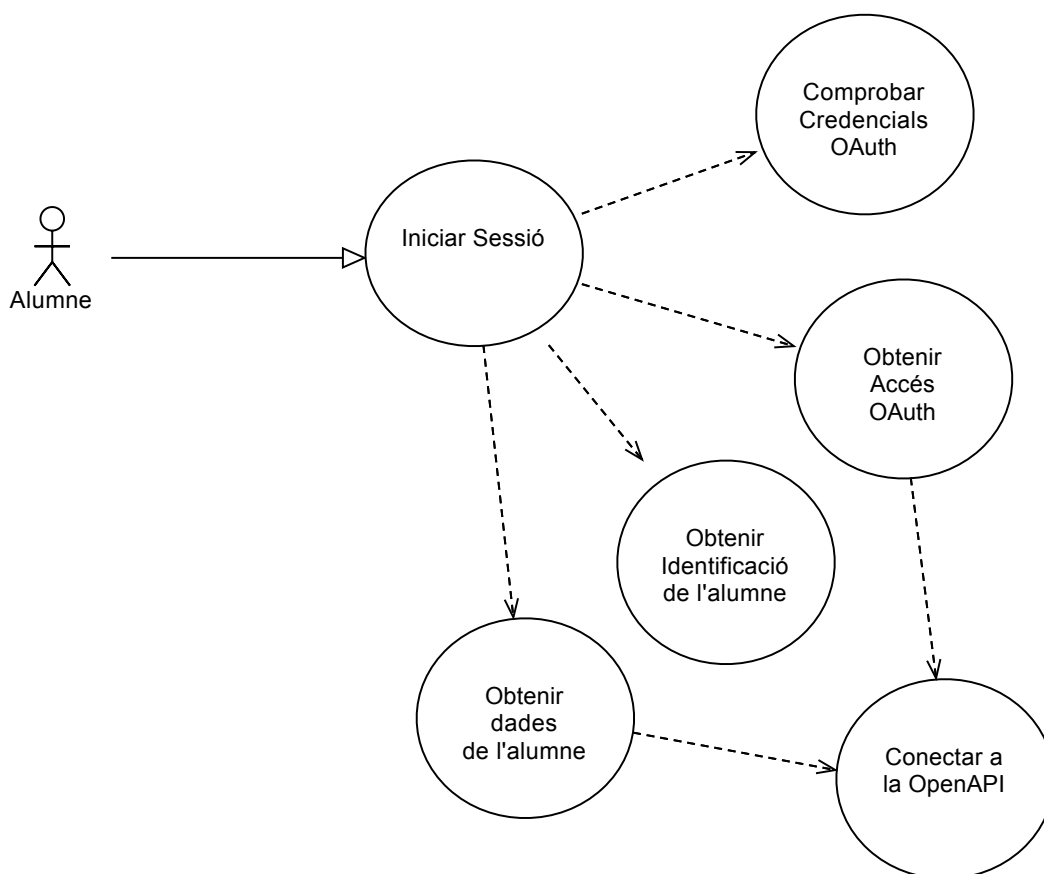
Hi ha un cas específic, però, que si es considera interessant expressar en forma diagrama de seqüència del cas d'ús per a fer més clar el accés als recursos de la **OpenAPI**. Aquest diagrama és el de autorització de OAuth 2 que es realitzarà per autorització un cop i a partir d'aquí amb el token garantit només s'accedirà als recursos. Més endavant s'especificarà amb molt més detall tot allò relacionat amb OAuth2 i com es realitzarà la implementació.

Figura 4.2.1 Diagrama de seqüència del procés d'OAuth2



**Figura 4.2.2 Escenari 1: Primera connexió d'usuari des de l'aplicació**

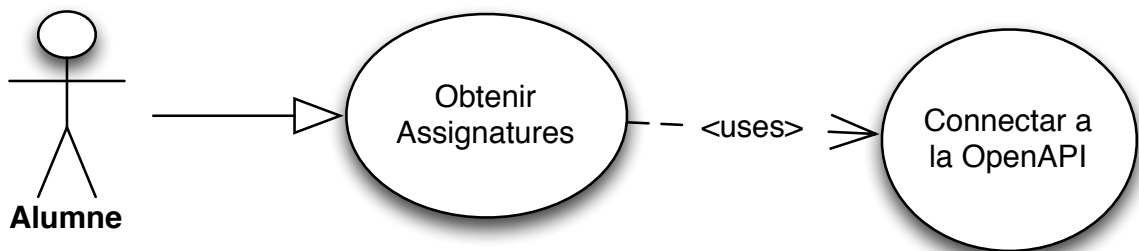
Escenari 1	
Descripció	Primera connexió de l'usuari des de l'aplicació
Context	L'usuari no s'ha connectat mai i inicia per primera vegada l'aplicació i per tant desencadena el procés de autorització de l'aplicació amb les credencials d'usuari.



El primer cop que l'alumne estableix la connexió, el servei haurà de demanar l'accés OAuth a la **OpenAPI** de la **UOC**. Per a fer-ho es disposa d'una clau d'identificació de aplicació i un token per a l'aplicació. Amb aquestes dades es re dirigeix a l'alumne cap a la pàgina d'autenticació de la **OpenAPI** s'identificarà amb el seu nom d'usuari i la seva contrasenya i atorgarà uns privilegis a la aplicació per a accedir als serveis amb la seva identificació. La **OpenAPI** retornarà aleshores un token i un secret que es faran servir per a accedir en posteriors ocasions a la consulta de serveis.

**Figura 4.2.3 Escenari 2: Obtenir les dades de les assignatures**

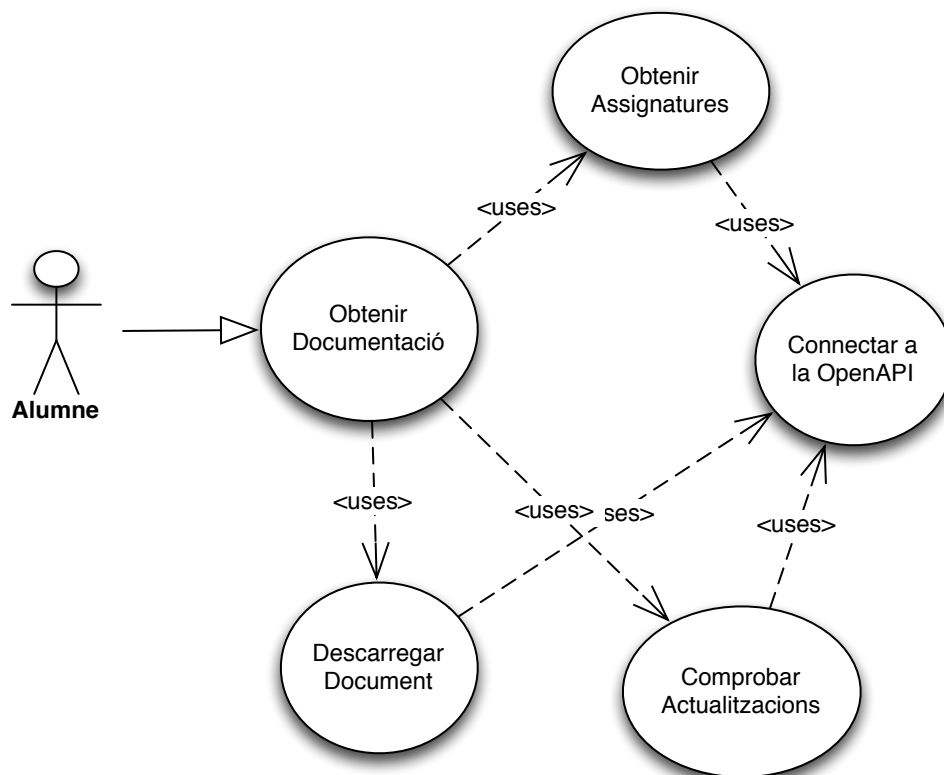
Escenari 2	
Descripció	Obtenir les assignatures de les que està matriculat l'alumne
Context	Donat que les connexions poden ser en diferents semestres, el sistema ha de ser capaç d'obtenir les assignatures de les que un alumne s'ha matriculat.





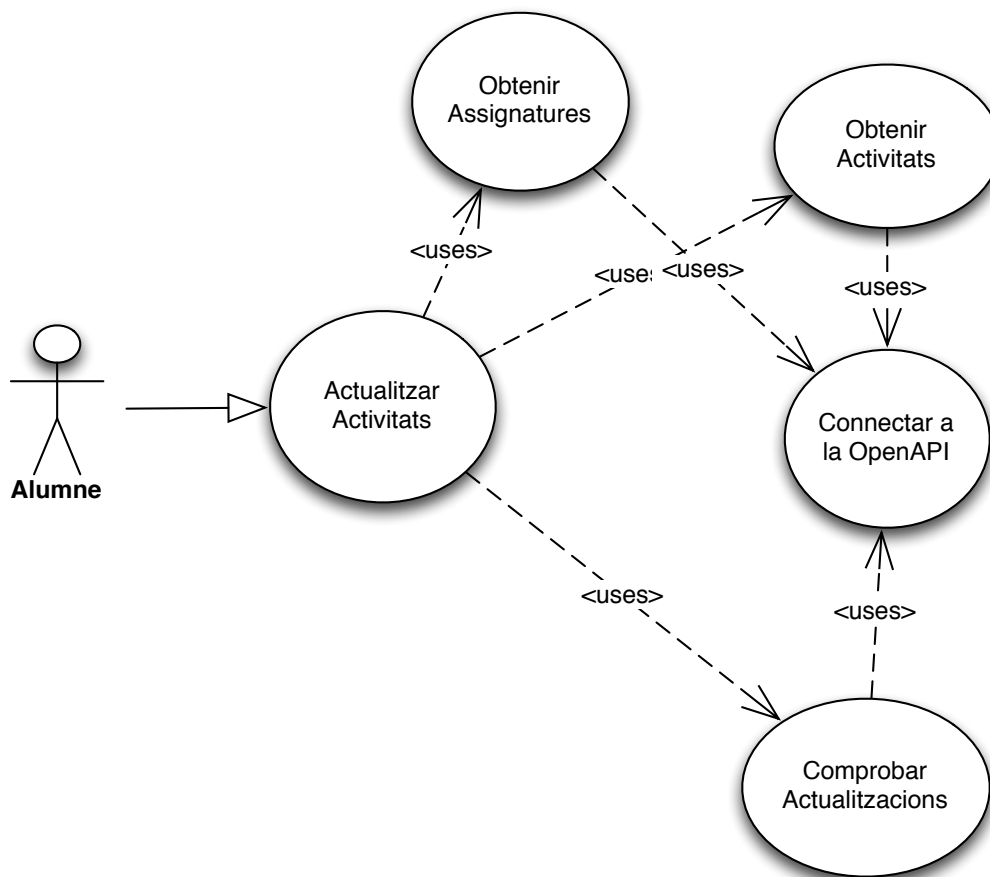
**Figura 4.2.4 Escenari 3: Consultar la documentació d'una assignatura**

Escenari 3	
Descripció	Consultar la documentació d'una assignatura
Context	Donada una assignatura, el sistema ha de ser capaç d'obtenir la seva documentació, gestionar les actualitzacions i permetre la seva visualització.



**Figura 4.2.5 Escenari 4: Actualitzar la llista d'activitats**

Escenari 4	
Descripció	Actualitzar la llista d'activitats
Context	Les activitats de les assignatures son potser l'element més interrelacionat amb el sistema i alhora l'element amb mes possibilitats de canvi. és per això que el sistema ha de permetre mantenir les activitats actualitzades.

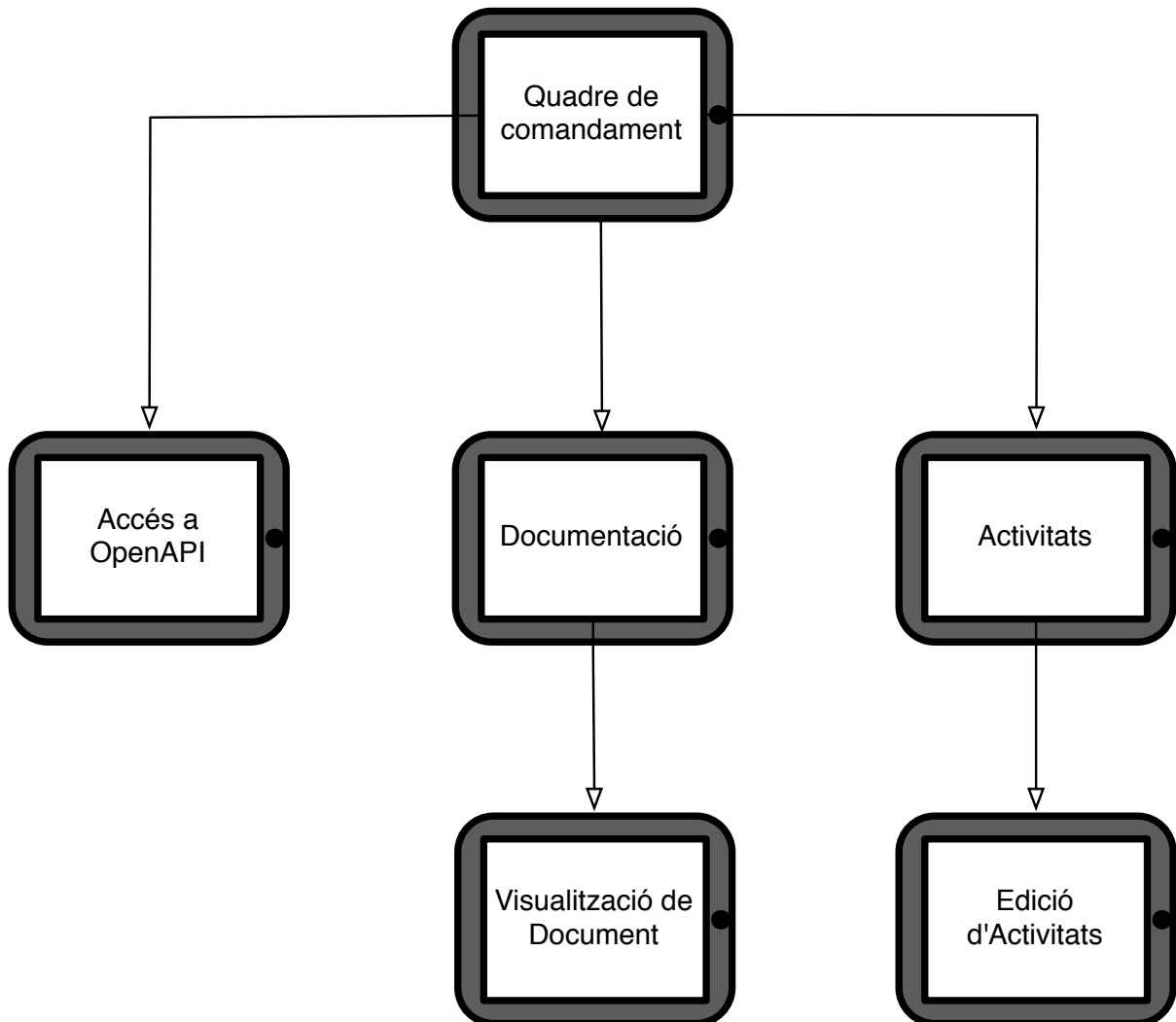


### Fluxos d'interacció

Els fluxos d'interacció de l'aplicació son els expressats en l'esquema. L'element principal és el Quadre de comandaments, que permetrà una visió global del sistema així com l'accés a les diferents funcionalitats del mateix.

Hi ha pantalles que son comuns donat que l'única diferència és l'origen del document i la tipologia lògica per tant, però que seran tractats de la mateixa forma.

**Figura 4.2.6 Fluxos d'interacció**



### 4.2.3. Prototipatge

Els prototipus de les diferents pantalles del flux d'interacció consistents en una primera idea de com organitzar visualment les pantalles de l'aplicació han estat realitzats sobre paper i mitjançant un scanner digitalitzats en imatges.

## **Quadre de comandament**

El quadre de comandament és la pantalla principal on es presenta tota la informació. Per a definir aquesta pantalla hi ha dues zones diferenciades, una és la zona comú, que es presenta a la part superior, on es podran trobar els controls per a accedir als diferents continguts, connectar i desconnectar i refrescar les dades presentades.

Donat que s'ha intentat donar molta importància a la senzillesa del disseny, es prototipen dues pantalles. Una que presenta el quadre de comandaments i la pantalla d'activitats a la zona específica i un altre que presenta el quadre de comandaments i la pantalla de documents a la zona específica.

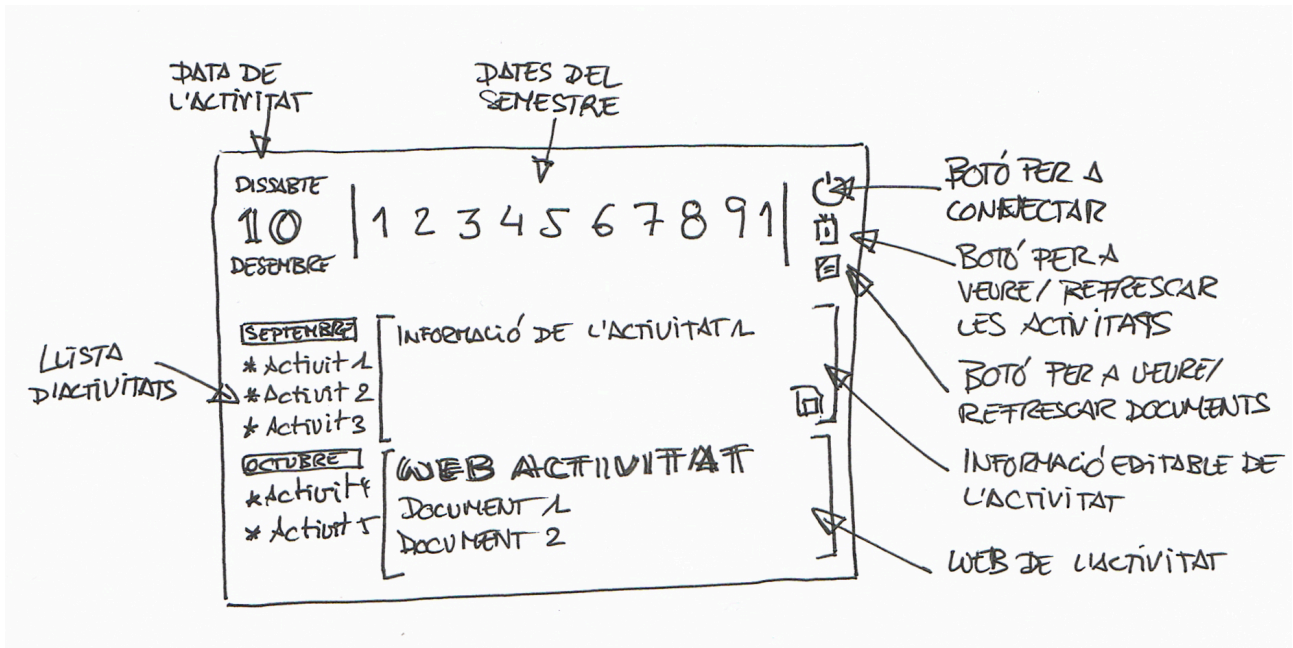
## **Pantalla d'activitats**

Com es pot observar la pantalla d'activitats presenta a la zona comú del quadre de comandaments la data de l'activitat seleccionada ( en cas de que cap activitat estigui seleccionada presentarà la data actual ), un control específic on es mostren els dies del semestre i que permetrà un desplaçament horitzontal i presentarà amb una senyal els dies que tinguin activitats. Finalment, a la dreta de la zona del quadre de comandaments es presenten els botons de connectar i desconnectar, mostrar i refrescar activitats i mostrar i refrescar documents. Aquests darrers botons son els que faran que el contingut de la zona específica canviï presentant una o altre informació.

A la part de la zona específica en el cas de les activitats es presenten tres zones definides. En primer lloc una llista ocupant tota la zona esquerra amb les diferents activitats organitzades per mesos. Aquesta llista permetrà un desplaçament vertical i serà sensible al tocs per a mostrar a les altres zones la informació relativa a la activitat seleccionada.

Les altres dues zones, ocupant les dues tercers parts de la pantalla a la esquerra, mostraran la informació de l'activitat seleccionada, que serà editable i emmagatzenable mitjançant un botó, i la plana web associada a la activitat.

Figura 4.2.7 Prototipus de la pantalla principal

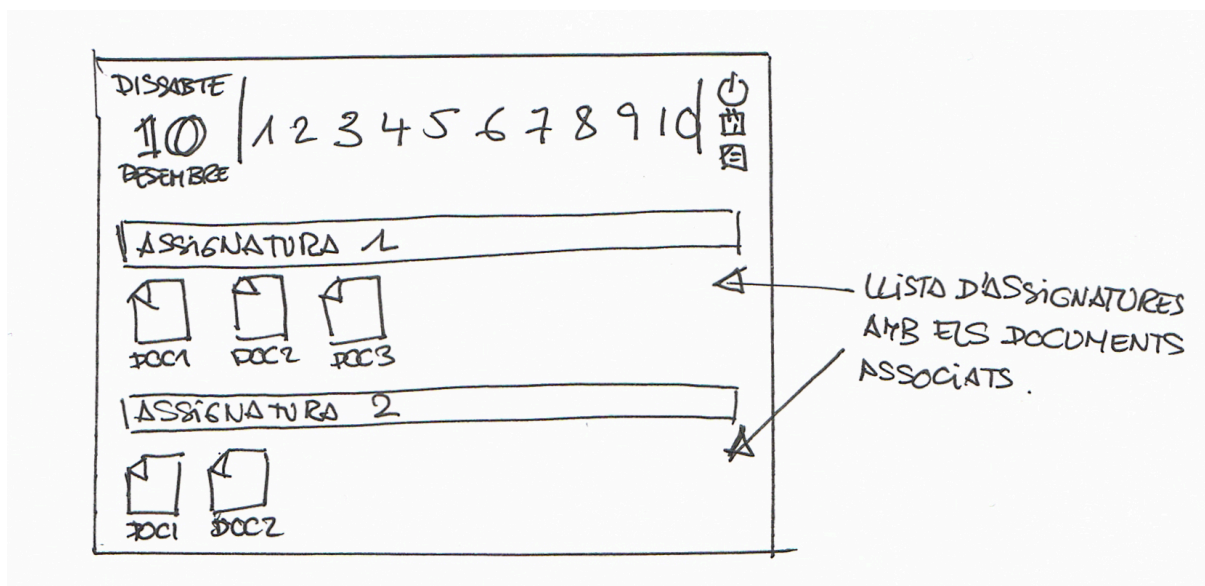


### Pantalla de Documents

La pantalla de documents només es referencia la part específica doncs la part comú del quadre de comandaments no varia. A aquesta zona es presentarà ocupant tot l'ample de la pantalla una llista de les assignatures a les que és matriculat l'alumne i dins cada una de les assignatures icones corresponents a cada un dels documents associats a l'aula de l'assignatura a la que l'alumne està inscrit. Aquestes icones seran sensibles als tocs i permetran accedir a una pantalla a la que es presentarà el document seleccionat per a la seva lectura.

S'ha decidit no prototipar la pantalla de visualització de documents per la seva simplicitat, doncs únicament es tractarà del document a pantalla completa.

Figura 4.2.8 Prototipus de la pantalla de documents



#### 4.2.4. Avaluació

Donat que el present projecte inclou un important component de connexió a un servei de dades, és important destacar que els tests que es poden realitzar en la fase d'avaluació seran realitzats sobre l'interfície d'usuari i la navegació per la mateixa però no sobre la connexió donat que és un sistema complex d'implementar i qualsevol prototipatge no seria evident per l'usuari i probablement no reflectiria la realitat.

#### Informació sobre l'usuari que realitza el test

Partint dels resultats obtinguts en la primera fase d'investigació, podem determinar dues característiques que agrupen els tipus d'usuari en quatre possibles categories.

1. *Ets o has estat alumne de la UOC.* Aquesta pregunta ens permetrà determinar el nivell de coneixement que l'usuari té de la metodologia d'estudi de la UOC i possiblement el nivell de realisme basat en l'experiència de les opinions expressades.
2. *Estas actualment cursant estudis.* El fet de trobar-se actualment realitzant estudis aporta un valor essencial a l'opinió doncs està basada en les experiències del moment.

Això ens proporcionaria quatre tipus d'usuari.

- Mai ha estudiat a la UOC i no estudia actualment.
- Mai ha estudiat a la UOC però estudia actualment.

- Ha estudiat a la UOC però no estudia actualment.
- Està estudiant a la UOC actualment.

### **Tasques que els usuaris haurien de realitzar**

Existeixen diferents programaris per a realitzar els tests dels prototipus com per exemple AppCooker<sup>8</sup> o BluePrint<sup>9</sup> per iPad o LucidChart o Prototyper per a escriptori. Tot i que són bones eines per a realitzar tests d'usabilitat, hem de considerar les limitacions que tenen cadascun d'ells. En el cas del que s'executen a les iPad la principal limitació és que el usuari de test ha de disposar d'un dispositiu físic al que provar el prototipus i en el cas del d'escriptori la limitació es dona pel fet que no són situacions prou semblants a la realitat i per tant el context queda alterat.

L'escenari que es proposa plantejar és realitzar els tests mitjançant AppCooker, per a obtenir un resultat el més real possible de com de senzilla i intuïtiva resulta la navegació entre pantalles, si l'estructura de navegació que és la que de forma natural es realitzaria o requereix realitzar tasques que no són de lògica natural.

Com s'ha comentat en l'anterior punt, existeixen restriccions de funcionalitats de les que no es poden fer tests com ara la connexió, exportació o inclusió de notes als documents.

### **Preguntes referents a les tasques**

Les preguntes que es plantejarien als usuaris són bàsicament preguntes de facilitat d'ús donat que és el que al cap i a la fi s'està posant en prova.

1. Consideres que el quadre de comandament proporciona una visió prou general de l'estat de les activitats?
2. Consideres que és senzill navegar per les diferents activitats i els diferents mesos del semestre en el quadre de comandament?
3. Consideres intuïtiu la forma d'accedir als diferents documents de l'assignatura?
4. Consideres adequada la forma en que es presenta la lectura de documents?
5. Creus que és intuïtiu i senzill la forma en que es realitzen les anotacions?
6. Consideres que l'exportació de documents mereix una pantalla pròpia o que hauria de ser una funcionalitat en la pantalla de documents?

---

<sup>8</sup> <http://www.appcooker.com/>

<sup>9</sup> <http://www.groosoft.com/blueprint/>

A més de plantejar les preguntes amb un si o no, és interessant facilitar la possibilitat de, en cas de que la resposta sigui no, l'usuari proposi la forma que creu correcte de presentar la funcionalitat o corregir l'errada.



## 4.3. Capítol 3: Disseny

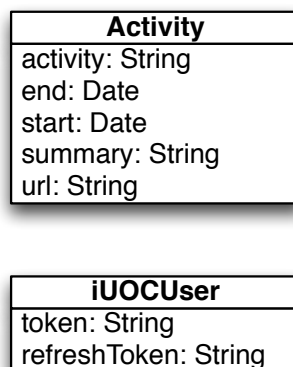
Aquest capítol es centra en el disseny de les diferents parts que integren el sistema, començant per el diagrama de classes, on ens centrarem en l'adaptació de les classes que s'ha realitzat de la **OpenAPI** de la *UOC* per al tractament de dades. Seguidament ens centrarem a la implementació de la llibreria d'accés OAuth2, que és la part més complicada del sistema donat que s'ha realitzat una implementació des de zero. Al següent apartat s'especificarà les decisions de disseny preses per a la persistència, que en aquest cas es realitzarà per a dos conceptes que son les dades de connexió de l'usuari i les dades de les activitats per a permetre les anotacions. El quart apartat és on es presentaran les decisions respecte als patrons de disseny utilitzats i les justificacions pertinents i, finalment, al darrer apartat es presentara el contingut de dades sobre el que s'està realitzant les proves.

### 4.3.1. Diagrama de classes

Les classes utilitzades per a la implementació del sistema son, bàsicament, les extretes de les especificacions de la **OpenAPI** de la *UOC* que és el proveïdor de dades i per tant marcarà el disseny de les classes de dades que farem servir.

Tot i que tractem dues entitats en el sistema que son Activitats i Documents (event i resource a **OpenAPI**), s'ha decidit implementar únicament una de les dues entitats que és Activitats doncs de Documents únicament es fa una lectura dels recursos i es construeixen dues llistes enllaçades de dades a partir de les dades obtingudes. En canvi de Activitats tenim que realitzar la persistència i per tant s'implementa la entitat que després es convertirà en un objecte `NSManagedObject`<sup>10</sup> per a poder emmagatzemar les dades en SQLite que és un dels sistemes de persistència que ofereix iOS.

**Figura 4.3.1 Diagrama de classes**



<sup>10</sup> [https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/CoreDataFramework/Classes/NSManagedObject\\_Class/Reference/NSManagedObject.html](https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/CoreDataFramework/Classes/NSManagedObject_Class/Reference/NSManagedObject.html)

Donat que per a l'accés a les dades fem servir el protocol OAuth2, hem d'emmagatzemar dues dades també, que son el token d'accés obtingut i el token de refresc per a quan expiri la validesa del token. Donat que aquestes dades son molt poc voluminoses i que només hem de tenir una instància d'aquesta classe, la decisió ha estat implementar la persistència fent servir el sistema de configuració d'opcions d'usuari `NSUserDefaults`<sup>11</sup> que permet emmagatzemar de forma molt senzilla opcions de configuració d'usuari.

### 4.3.2. Detalls del sistema de accés a OAuth2

Considerant l'entorn actual en el qual desenvolupem les aplicacions, no aprofitar l'avantatge que ens ofereix l'accés a les dades per part dels diferents proveïdors de dades i serveis representaria perdre moltes oportunitats de crear clients o dotar les aplicacions de moltes i molt potents funcionalitats. Per aquesta mateixa raó, el fet que hi hagi aquesta gran varietat de serveis fa que l'usuari mitjà s'hagi de registrar múltiples vegades de forma que cada cop que es registra en un servei ha de proporcionar les seves dades al servei i recordar un usuari i una paraula de pas. Així mateix el fet de oferir accés per part d'aquests serveis a clients obre un forat de seguretat en haver la necessitat de que aquests clients del servei hagin d'emmagatzemar les dades de l'usuari com ara el nom d'usuari o el password. Aquest fet fa que es concedeixi un accés sense restriccions i total al client, a més de no poder cancel·lar l'accés d'aquest client si no és des del mateix client.

Per a solucionar tots aquestes problemes apareix OAuth. OAuth presenta un sistema en el qual per contra del tradicional *grant* amb el parell usuari-contrasenya, el que es fa és permetre l'accés a recursos a través d'un token que permet al client, prèviament identificat, accedir al recursos que l'usuari ha seleccionat. Això permet que el client no conegui el nom d'usuari ni la contrasenya, que es pugui cancel·lar l'accés als recursos des del client i que un canvi en les dades d'accés al servei per part de l'usuari no cancel·li l'accés al servei per part del client.

Els passos per a utilitzar el sistema son els següents:

1. Registrar la aplicació client al sistema de forma que l'usuari pugui comprovar que qui demana l'accés és qui diu dir que es. Aquest registre proporcionarà dos valors anomenats *Client\_Id* i *Client\_Secret* que seran utilitzats per part del client per a identificar-se.
2. Quan l'aplicació client desitja accedir als recursos oferts per part del servei, ha de connectar amb una plana web que ofereix el servei, on es demana que l'usuari s'identifiqui. Aquestes

---

<sup>11</sup> [https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/CoreDataFramework/Classes/NSManagedObject\\_Class/Reference/NSManagedObject.html](https://developer.apple.com/library/mac/#documentation/Cocoa/Reference/CoreDataFramework/Classes/NSManagedObject_Class/Reference/NSManagedObject.html)

dades son sol·licitades pel servei i obtingudes al servei, per tant el client no emmagatzema aquestes dades.

3. Si les dades d'usuari son vàlides, el servei presentarà una segona plana web a la qual mostrarà una llista de serveis que poden ser autoritzats i als quals l'usuari haurà d'indicar quins son els serveis que permet utilitzar.
4. Un cop indicats els serveis, el servei retornarà a una url de resposta predetermina, que en cas de l'accés des d'un dispositiu mòbil serà una url no valida que permetrà al dispositiu detectar que s'ha finalitzat la validació i recollir de la resposta les dades per a accedir al servei.
5. El servei proporcionarà al client tres dades.
  - 5.1. *Token* - Valor que indicarà al servei quin és el codi d'autorització.
  - 5.2. *Validesa del Token* - Temps per el qual és vàlid el token a partir del moment en que ha estat generat.
  - 5.3. *Refresh Token* - Token especial que servirà per a obtenir un nou token vàlid per a un nou període.
6. Per a accedir als recursos, es realitzarà la crida a la url del servei rest que ofereix els recursos, passant com a paràmetres a la capçalera de la crida el *Client\_Id*, el *Client\_Secret* i el *Token* de forma que s'identifica el client i l'autorització.

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

Per a la persistència del sistema d'han utilitzat dos dels sistemes de persistència que ofereix iOS. Per una banda s'ha fet servir els "*User Defaults*" per a emmagatzemar les dades de connexió de l'usuari a la **OpenAPI** de la UOC i per l'altre s'ha fet servir *Core Data* per a l'emmagatzemament de les dades relatives a les activitats.

A continuació es presenten els trets principals dels dos sistemes.

#### User Defaults

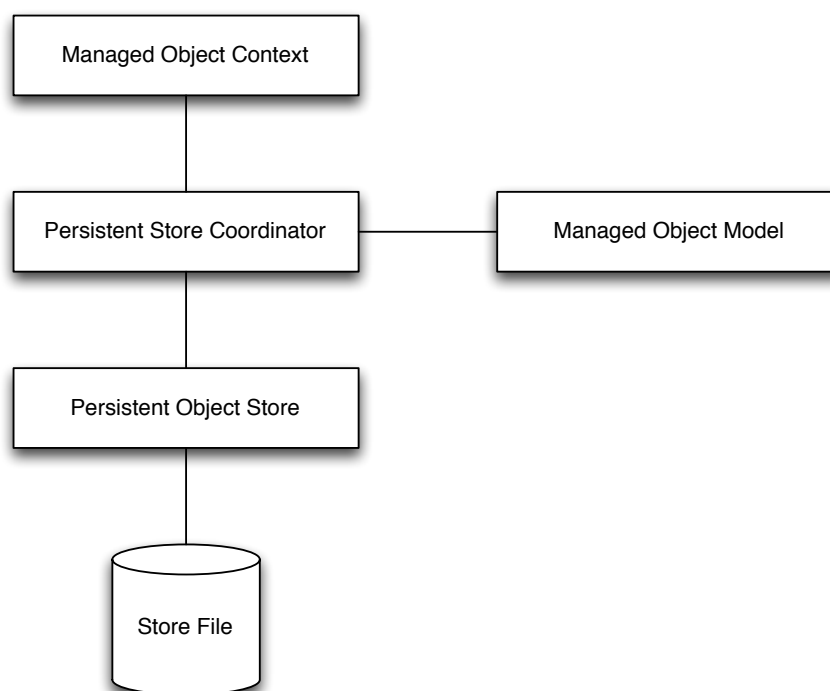
Segons la definició d'Apple *'la classe NSUserDefaults proporciona una interfície de programa per a interactuar amb les configuracions per defecte del sistema. Les configuracions per defecte del sistema permeten a una aplicació adaptar el seu comportament per a complir les preferències de l'usuari. Per exemple, pots permetre als usuaris les unitats de mesura que la aplicació mostrarà o com de sovint els teus documents han de ser emmagatzemats de forma automàtica. Les aplicacions emmagatzemen aquestes preferències assignant valors a un conjunt de paràmetres a la base de dades de preferències d'usuari. Ens referim als paràmetres com a preferències donat que son normalment utilitzats per a determinar l'estat per defecte de*

*l'aplicació o la forma en que aquesta actua per defecte*'. Donada aquesta definició i la senzillesa de l'ús de la classe, s'ha decidit que aquest el el millor sistema de persistència per a les dades d'accés al servei.

## Core Data

El sistema de persistència de *Core Data* és força més complexe que no pas *NSUserDefaults*. Bàsicament *Core Data* és un sistema que en permet aïllar la forma de representació de les dades del mètode d'emmagatzemament utilitzat. Per a fer-ho *Core Data* proporciona un conjunt d'objectes anomenat '*Core Data Stack*' que permet que podem triar el tipus d'emmagatzemament independentment del tipus de dades i treballar directament amb objectes sense tenir en compte la resta del procés. L'esquema de '*Core Data Stack*' és el següent:

**Figura 4.3.2 Stack de Core Data**



- **Managed Object Context** - És una col·lecció de *Managed Objects*. Cada *Managed Object* és una entitat de la que volem tenir persistència. És una classe que derivem de *NSManagedObject* i que adaptem a les nostres necessitats.
- **Persistent Store Coordinator** - Coordina les diferents *stores* que tenim. És a dir, els diferents tipus d'emmagatzemament que podem estar utilitzant en cada moment.
- **Persistent Object Store** - És un conjunt de dades en un sistema d'emmagatzemament concret com pot ser SQLite, Memòria, XML, etc ...

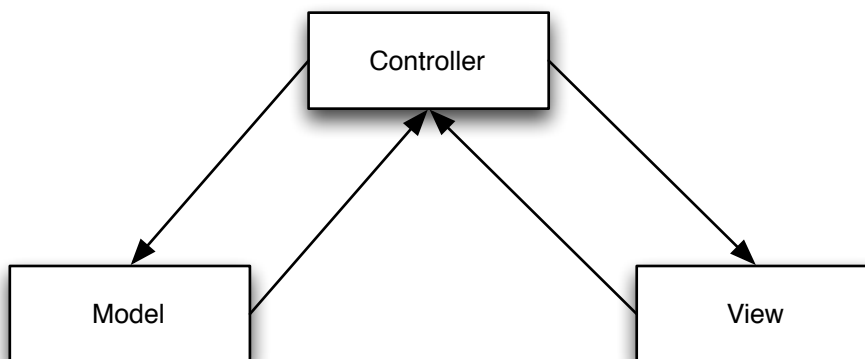
- **Managed Object Model** - Model en que definim la representació de les dades que volem utilitzar i de les que es derivaran les entitats de *managed object* i de les diferents *stores* que podem estar utilitzant.

#### 4.3.4. Utilització dels patrons de disseny MVC, Singleton i Delegate

Per a la implementació de l'aplicació iUOC s'ha optat per utilitzar els patrons de disseny MVC, Singleton i Delegate. La justificació de la utilització de cadascun d'ells és la següent:

- **MVC** - El patró Model-View-Controller és el patró de disseny bàsic a utilitzar en el framework cocoa, que és el utilitzat per al desenvolupament de l'aplicació *iUOC*. Aquest model de disseny presenta el següent esquema.

Figura 4.3.3 Esquema del patró MVC



El fet de aïllar el model de dades, del controlador on residirà la lògica de negoci, de la vista que és on es presentaran les dades i s'obtidran les accions i entrades d'informació, permet separar, independitzar i reutilitzar els components per a altres desenvolupaments, així en cas de que volguéssim portar un desenvolupament que està basat en iOS a un desenvolupament per OSX només hauríem d'implementar la vista, doncs la resta podria ser compatible. La realitat però és que la forma en que està implementat el MVC en Cocoa té una cohesió més elevada del que hauria de ser entre la vista i el controlador, essent realment un M-VC.

- **Singleton** - El patró de disseny Singleton es basa en que només hi ha una instància d'una determinada classe en tot moment. Aquesta instància és creada en fer la primera crida a la classe i retornada cada cop que es sol·licita per part de qualsevol component del sistema. Aquest patró és especialment adient per a casos en que es vol evitar col·lisions de crides, controlar els accessos des de un mateix punt o limitar d'alguna forma la creació d'objectes que son costosos. Aquest és el cas de l'accés a la **OpenAPI** de la *UOC* i és per això que

s'ha decidit implementar el patró de disseny singleton per a la classe OpenAPI. Donat que l'entorn de execució de l'aplicació és multi-fil i que les crides es fan des de diferents fils, hi ha que tenir força en compte aquest fet i assegurar-se que quan es fan les crides, es facin des del fil que es facin sempre retorni la mateixa instància i que aquesta sincronitzi les crides que es facin.

- **Delegate** - El patró delegate és aquell que descarrega certes tasques a realitzar en un objecte delegat i, un cop finalitzades les tasques, reprèn el control per a continuar les tasques. Aquest és un patró força comú en Cocoa i és ideal per a fer servir en connexions de dades, on la tasca de connexió i tractament de les dades podria bloquejar l'execució de la interfície d'usuari. La forma de fer-ho servir en Cocoa és creant el que s'anomena protocol. Un protocol és un conjunt de mètodes que la classe que compleix el protocol ha d'implementar. D'aquesta forma quan la classe delegada acaba la tasca encomanada pot cridar el mètode del delegat per indicar la finalització o fins i tot per a passar les dades obtingudes si és necessari.

#### 4.3.5. Tecnologies específiques utilitzades

D'entre de les diferents tecnologies utilitzades cal destacar dues que resulten imprescindibles per a la bona gestió de l'aplicació.

##### JSON

Segons la definició de wikipedia, "JSON és un estàndard obert basat en text dissenyat per a intercanvi de dades llegible per humans. Deriva del llenguatge script JavaScript, per a representar estructures de dades simples i llistes associatives, anomenades objectes. Malgrat la seva relació amb el JavaScript, té implementacions per a gran part dels llenguatges de programació."

En aquest cas es necessària l'utilització d'aquesta tecnologia doncs els resultats proporcionats per el servei OpenAPI de la UOC es serveixen en JSON. Per a utilitzar-ho farem servir el parser que proporciona el framework cocoa mitjançant la classe *NSJSONSerialization* que transformarà les dades obtingudes en una instància de la classe *NSDictionary*.

##### GCD

Grand Central Dispatch és una tecnologia desenvolupada per Apple que te com a objectiu optimitzar el suport ofert a les aplicacions per a aprofitar els processadors de més d'un nucli.

Aquest conjunt de tecnologies permet als desenvolupadors crear amb mes facilitat aplicacions que obtenen el màxim rendiment de sistemes amb múltiples nuclis donat que els diferents fils d'execució estan gestionat per GCD de forma transparent sense que l'aplicació hagi de fer una

implementació específica, la qual cosa resulta en un rendiment superior a l'hora de fer servir programació multi fil.

Es possible doncs especificar diferents cues de execució, per a poder llençar execucions seqüencialment en una fil, donar prioritats a les cues d'execució o forçar l'execució d'un bloc a un fil especial anomenat '*main\_thread*' que és el fil d'execució principal i on han de ser executades totes aquelles accions que tinguin relació amb l'interfície d'usuari.

#### 4.3.6. Selecció dels continguts per a la primera versió

Les dades utilitzades en la primera versió de l'aplicació son les proporcionades per la **OpenAPI** de la *UOC*. Aquestes dades consisteixen en tres activitats, que es situen en dies consecutius a partir del dia en curs, la descripció que tenen és '*Summary de l'event amb id x*' i la url que proporcionen és incorrecta, per tant, amb la finalitat de mostrar un resultat el més real possible, la url s'ha "hardcodejat" amb la url de la plana inicial de la *UOC*.

Així mateix es proporcionen dues Assignatures i dues aules a les que és matriculat l'alumne:

- Introduction to Java Programming Aula 1
- Introduction to C Programming Aula 1

Amb dos documents cadascuna que apunten a un mateix document a la xarxa. Aquest document és un fitxer en format pdf que es farà servir per a presentar el document en pantalla per a la seva lectura.

## 4.4. Capítol 4: Implementació

El contingut del capítol 4: Implementació seran tots aquells detalls tècnics dels continguts que s'ha anat presentant en capítols anteriors així com la explicació de forma visual de quin és el funcionament de l'aplicació *iUOC*, donat que no és possible executar-la en un dispositiu que no sigui un iPad o un Macintosh amb l'entorn de desenvolupament instal·lat.

### 4.4.1. Eines utilitzades

Per al desenvolupament d'una aplicació per a iPad és necessari fer servir tot un seguit d'eines específiques, tant de programari com de maquinari. Així mateix s'han fet servir tot un seguit d'eines, que si bé no son estrictament necessàries si han estat de gran ajuda per a completar el desenvolupament de l'aplicació en tots els seus aspectes. Aquestes eines son:

- Macbook Pro 15' amb el sistema operatiu OSX 10.8.2 ( Mountain Lion)
- Entorn de desenvolupament XCode Versió 4.5.2 (4G2008a) amb iOS SDK 6.0
- Instruments. Programari per al control de memory leaks, dead locks, zombies, etc ... inclòs en la suite de XCode.
- PixelMator. Programari de dibuix per al disseny de la icona i els elements gràfics de l'aplicació.
- Omnigraffle. Programari per al disseny de diagrames.
- TextMate. Programari de edició de codi font per a realitzar les proves inicials de servidor no OpenAPI.
- GIT. Sistema de control de versions.

### 4.4.2. Detalls tècnics dels controls utilitzats per a la presentació

L'entorn de desenvolupament *Cocoa* ofereix diferents mètodes per a la presentació dels continguts dels quals a la present aplicació s'han fet servir *UILabel* (etiquetes), *UITableView* (taules), *UIButton* (botons), *UIWebView* ( vista de web ) i *UIScrollView* ( vistes desplaçables ). Tots aquests controls presenten aspectes estandarditzats, per la qual cosa s'ha optat per realitzar diferents personalitzacions del controls per a oferir una imatge uniforme i no estàndard de l'aplicació. Les principals adaptacions de les vistes son les següents:

#### **HorizontalDayPickerView**

Es un control creat expressament per a l'aplicació en substitució al *UIDatePicker* que ofereix el sistema. La decisió es realitza per dos raons bàsicament. En primer lloc degut a que l'aspecte físic que presenta el *UIDatePicker* del sistema no encaixa amb la imatge general del sistema i



en segon lloc per que no permet incloure ( no de forma senzilla ) marques que indiquin d'alguna forma que hi ha activitats a un dia concret. Els aspectes de ambdós controls son:

**Figura 4.4.1 Control UIDatePicker**

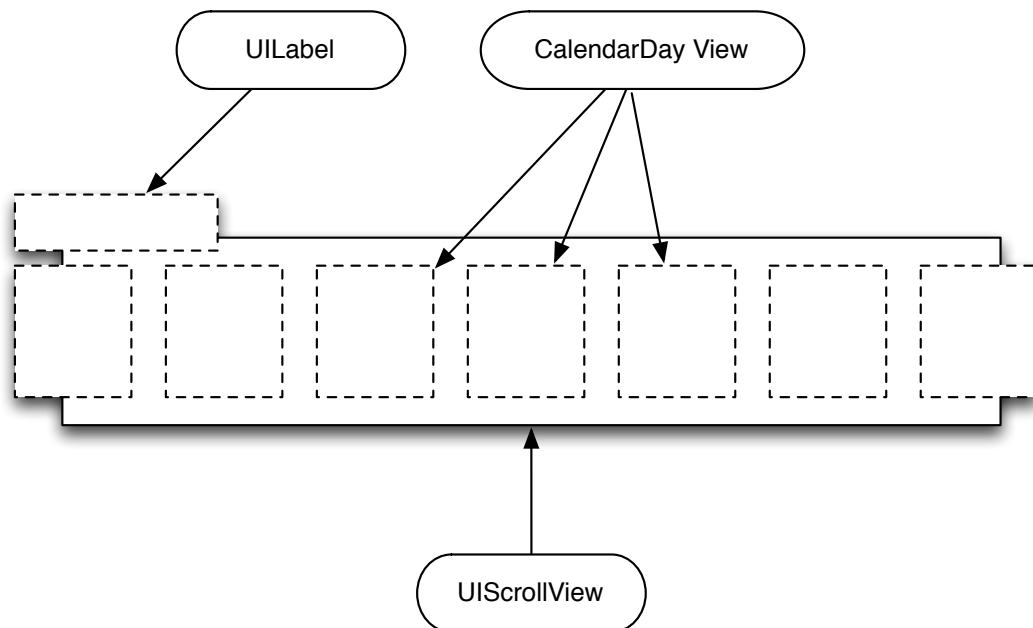


**Figura 4.4.2 Control HorizontalDayPickerView**



L'esquema de creació del control és ell següent:

**Figura 4.4.3 Esquema de creació del copntrol HorizontalDayPickerView**



Un contenidor de la classe *UIScrollView* que conté el nombre de *CalendarDayView* (control específic creat per a *iUOC*) que son visibles. Es controla segons la finestra visible del control quins son els dies que s’han de mostrar i es s’insereixen de forma dinàmica en resposta als esdeveniments de desplaçament del control. A més hi ha un control *UILabel* que es situa sobre el dia 1 de cada mes indicant quin és el més que s’inicia. Aquest control permet les següents controlar les següents accions i propietats:

- **startMonth** - Determina el primer mes que s’ha de mostrar en el control.
- **endMonth** - Determina el darrer mes que s’ha de mostrar en el control.
- **selectedDate** - Retorna la data seleccionada en cada moment.
- **elementWidth** - Determina la amplada que ha de tenir cada element *CalendarDayView*.
- **activities** - Determina la llista d’activitats del mes en curs.

### **UITableView**

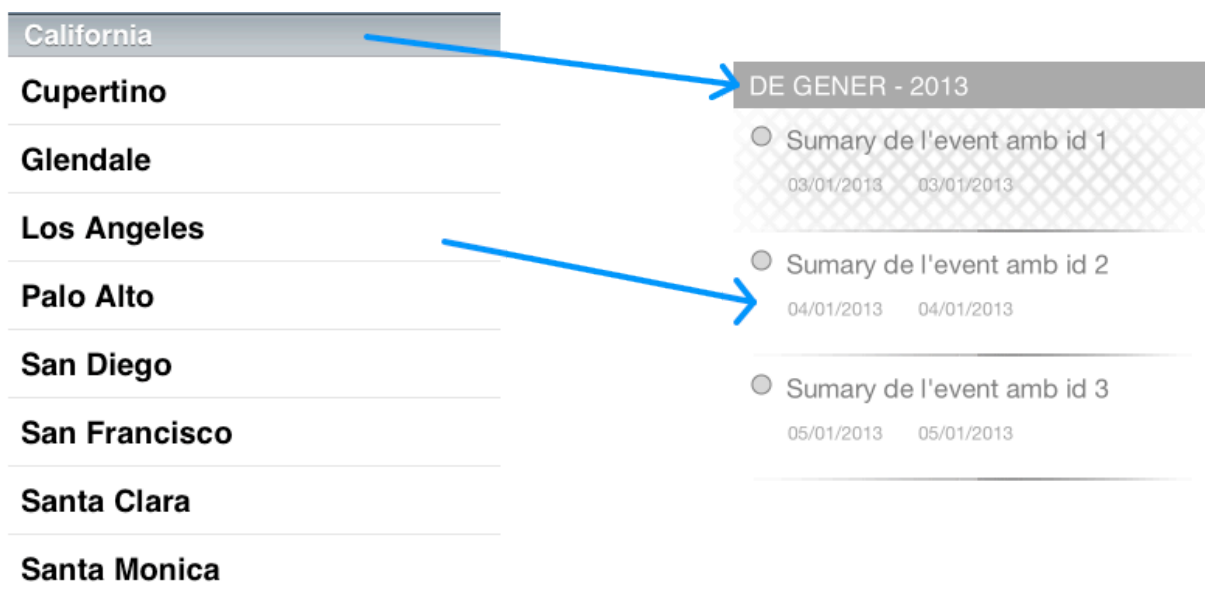
Les adaptacions realitzades a la taula han estat per a adaptar la imatge visual i les dades presentades a les cel·les. Per a això s’ha creat una classe *GlobalDataTableViewCell* que deriva de la classe del framework *UITableViewCell* i que permet modificar les dades presentades així com la presentació de les cel·les seleccionades. L’esquema de la cel·la creat és el següent:

**Figura 4.4.4 Cel·la adaptada per a la presentació d'activitats**



I el resultat de adaptar les taules és el següent:

**Figura 4.4.5 Taula adaptada per a presentar la llista d'activitats**



### 4.4.3. Accés a la OpenAPI de la UOC

El component d'accés a la **OpenAPI** de la *UOC* és el que tècnicament presenta una dificultat més gran. Com s'ha explicat en punts anteriors, el accés a un servei que ofereix els seus recursos mitjançant el protocol OAuth2 no és un desenvolupament trivial i la principal dificultat que presenta és l'autenticació i manteniment de l'accés al sistema.

Tot i que hi ha diferents llibreries desenvolupades a Internet, no ha estat possible utilitzar cap de les que s'han provat com [nxtbgtngn/OAuth2Client](https://github.com/nxtbgtngn/OAuth2Client)<sup>12</sup>, [AFNetworking/AFOAuthClient](https://github.com/AFNetworking/AFOAuthClient)<sup>13</sup> y

<sup>12</sup> <https://github.com/nxtbgtngn/OAuth2Client>

<sup>13</sup> <https://github.com/AFNetworking/AFOAuthClient>

lukeredpath/LROauth2Client<sup>14</sup> degut a la implementació de la **OpenAPI** de la *UOC*. Per aquesta raó s'ha optat per desenvolupar una llibreria pròpia que permetés gestionar les particularitats de la API i obtenir les dades necessàries per al funcionament de l'aplicació.

S'han pres diferents opcions de disseny entre les que es troben el patró a aplicar per a solucionar la casuística de les possible col·lisions en accedir a les dades, com gestionar el retorn de les dades per a permetre refrescar-les a tots els components que en depenguin o la gestió de les connexions que s'ha de realitzar mitjançant l'accés a una plana web.

### Patró de disseny triat

Com s'ha comentat anteriorment s'ha optat per aplicar el patró de disseny Singleton que permet limitar el nombre d'instàncies d'una classes concreta a una única instància. D'aquesta forma es poden canalitzar i controlar totes les peticions de forma que la única instància existent és capaç de gestionar les peticions i servir les dades.

La implementació a la classe es defineix bàsicament en dos punts, el mètode *sharedInstance* de la classe *OpenAPI* definit com a mètode de classe, el que es definiria com a un mètode estàtic en C++. Per a accedir a la instància de la classe es cridarà a aquest mètode que retornarà una instància, sempre la mateixa, de la classe *OpenAPI*. El segon punt important és a la implementació d'aquest mètode.

```
+(id) sharedInstance
{
    static dispatch_once_t pred;
    static OpenAPI *sharedInstance = nil;
    dispatch_once(&pred, ^{
        sharedInstance = [[OpenAPI alloc] init];
        // Class initialization stuff
    });
    return sharedInstance;
}
```

Els punts a comentar son, la declaració de la variable *sharedInstance* com a static i la crida, mitjançant *dispatch\_once*, mètode de GCD<sup>15</sup> que permet realitzar la execució d'un bloc de codi en un fil separat i executar-ho només un cop, d'aquesta forma en declarar com a static *sharedInstance* no es redefinirà en cap crida i en fer la inicialització mitjançant *dispatch\_once* ens garanteix que només es realitzarà la inicialització una vegada.

---

<sup>14</sup> <https://github.com/lukeredpath/LROAuth2Client>

<sup>15</sup> [http://developer.apple.com/library/ios/#documentation/Performance/Reference/GCD\\_libdispatch\\_Ref/Reference/reference.html](http://developer.apple.com/library/ios/#documentation/Performance/Reference/GCD_libdispatch_Ref/Reference/reference.html)

## Registre d'usuari

Per a realitzar el registre d'usuari el primer que s'ha de tenir en compte és que una de les raons, i una de les avantatges, d'utilitzar OAuth2 és no proporcionar les dades de registre a l'aplicació client del servei. Per a aconseguir-ho OAuth2 especifica que s'ha d'accedir a una plana web d'autoritzacions<sup>16</sup>. Això presenta diferents particularitats doncs no és habitual que s'obtinguin dades a partir d'una plana web ja que per a això es disposa d'objectes específics que implementen connexions a través d'Internet.

Per a salvar aquesta problemàtica, el que s'ha fet és permetre l'accés a través d'un control *UIWebView* (control d'accés i presentació de planes web estàndard de cocoa) al qual la classe *OpenAPI* s'especifica com a delegat, de forma que les diferents notificacions que genera el control siguin capturades per la instància d'*OpenAPI*.

El control es realitza en un punt, on es sol·licita si s'ha de iniciar la càrrega d'una url.

```

- (BOOL)webView:(UIWebView *)webView shouldStartLoadWithRequest:(NSURLRequest *)request
navigationType:(UIWebViewNavigationType)navigationType
{
    NSString *urlRequest = [[request URL] absoluteString];

    if ([urlRequest rangeOfString:kOAuthRedirectURL].location != 0)
        return YES;
    else {
        NSString *tokenRequestCode = [urlRequest substringFromIndex:[urlRequest
rangeOfString:@"?code="].location+6];
        NSLog(@"Requesting code: %@", tokenRequestCode);

        [self accessTokenExchange:tokenRequestCode];

        return NO;
    }
    return NO;
}

```

A aquesta crida s'indica que s'ha sol·licitat, de forma manual o automàtica, la càrrega d'una url al control. Quan la url sol·licitada per a carregar coincideix amb la url de redirecció en cas de una autenticació correcta per part de l'usuari, s'ha de capturar les dades del codi de sol·licitud de token que és proporcionat a través de la url, cridar al mètode que sol·licita obtenir el token amb el codi de sol·licitut i indicar que no s'ha de procedir a la càrrega de la url per a evitar que es presenti un error.

Com es pot observar al codi es comprova si és la url d'autoredirecció i si ho és es captura el codi i es crida a *accessTokenExchange*.

---

<sup>16</sup> Aquest fet no és sempre així i per això d'indica que la implementació d'OAuth2 que fa la *OpenAPI* no és del tot estàndard doncs per a accessos des d'una aplicació mòbil l'estàndard marca que s'ha de poder accedir mitjançant el parell nom d'usuari-password per a obtenir el codi d'autorització.

Si bé és cert que es podria haver optat per un altre model en el qual est realitzes només una encapsulació de la classe `NSURLConnection`, això hagués fet la compartició de resultats mitjançant el sistema de notifikacions hagués estat molt més complicat donat que es podrien realitzar diverses crides simultàniament al mateix servei de dades i que s'atengués a un resultat que no fos el que s'ha crida, malbaratant així les crides. L'altre opció seria fer-ho utilitzant delegats, però novament malbarataríem crides probablement.

## Consultes de dades

Per a la consulta de dades es fa servir una classe anomenada `AsyncURLConnection` que bàsicament és una encapsulació de la classe `NSURLConnection` que ens permet utilitzar blocs per a poder definir la crida d'una forma còmode sense necessitat d'utilitzar delegats si no que podem utilitzar blocs per a donar servei a les notifikacions provinents de `NSURLConnection`.

Totes les consultes de dades acaben cridant al mètode `performGETRequest`. Aquest mètode fa una crida que s'ha adaptat per a tenir el format d'una crida de dades seguint el protocol OAuth2, gestionar el resultat i fer una crida al sistema de notifikacions per a permetre que tots els components que es nodreixin de les dades obtingudes puguin actualitzar-les.

Els paràmetres que rep el mètode son:

- **url** - Adreça d'Internet que ofereix el servei.
- **parameters** - Paràmetres, en cas de que el necessiti, que s'han d'incorporar a la capçalera de la crida. Aquest paràmetre és un `NSDictionary` que bàsicament és una llista de parelles clau-valor.
- **rid** - En cas de que les dades s'hagin sol·licitat per a un objecte amb un id concret, s'haurà de passar aquest id que serà retornat per la crida tal i com es reb. L'objectiu d'aquest paràmetre és identificar al objecte que ha fet la sol·licitud de les dades.
- **notificationKey** - Clau de notifiació que s'ha de cridar quan s'hagin obtingut les dades.

```
- (void)performGETRequest:(NSString *)url withParameters:(NSDictionary *)parameters
forId:(NSString *)rid notifyOnCompletion:(NSString *)notificationKey
{
    NSMutableString *parametersString = [NSMutableString stringWithFormat:@"%s?", url];

    for (NSString* key in parameters)
        [parametersString appendString:[NSString stringWithFormat:@"%s=%s&", key,
[parameters objectForKey:key]]];

    NSMutableURLRequest *openAPIRequest = [[NSMutableURLRequest alloc] initWithURL:
[NSURL URLWithString:parametersString]];

    NSString *requestParams = [NSString stringWithFormat:@"Bearer %@",
self.accessToken];

    [openAPIRequest setValue:requestParams forHTTPHeaderField:@"Authorization"];
```

```

[AsyncURLConnection request:openAPIRequest completeBlock:^(NSData *data, NSString
*rId) {
    dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
        /* process downloaded data in Concurrent Queue */
        NSError *serializationError = nil;
        NSDictionary *responseHandling = [NSJSONSerialization
JSONObjectWithData:data options:NSJSONReadingMutableLeaves error:&serializationError];

        // Error on parsing, something went really bad
        if (responseHandling == nil) {
            dispatch_async(dispatch_get_main_queue(), ^{
                NSError *error = [[NSError alloc] initWithDomain:NSURLErrorDomain
code:NSURLErrorUserCancelledAuthentication userInfo:nil];
                [[NSNotificationCenter defaultCenter]
postNotificationName:kOpenAPINotificationRequestFailed object:error];
            });
            return;
        }

        // Check token error and try to re-request it
        if ([[responseHandling objectForKey:@"error"]
isEqualToString:@"invalid_token"]) {
            dispatch_async(dispatch_get_main_queue(), ^{
                [self refreshTokenAndPerformSelector:@selector(requestEventsList)];
            });
            return;
        }

        dispatch_async(dispatch_get_main_queue(), ^{
            NSMutableDictionary *responseData = [responseHandling mutableCopy];
            if (rId == nil)
                [responseData setObject:@"" forKey:@"requesterid"];
            else
                [responseData setObject:rId forKey:@"requesterid"];

            [[NSNotificationCenter defaultCenter]
postNotificationName:notificationKey object:responseData];
        });
    });
} errorBlock:^(NSError *error) {
    [[NSNotificationCenter defaultCenter]
postNotificationName:kOpenAPINotificationRequestFailed object:error];
} forId:rId];
}

```

El primer que es fa és muntar la url amb la adreça proporcionada i els diferents paràmetres, travessant la llista de paràmetres i muntant-ho un per un en forma de paràmetres GET.

A continuació es crea la petició amb *NSURLRequest* afegint-hi els paràmetres d'OAuth2 a la capçalera i finalment es fa la crida.

En obtenir els resultats de la crida es fa servir *Grand Central Dispatch* per a realitzar el processament de les dades obtingudes en un fil apart del principal i no aturar així la execució del fil principal i no congelar la interfície d'usuari per tant.

En cas d'error en el parsing es realitza una notificació al fil principal d'error per a que sigui gestionat al controlador de la pantalla que estigui visible.

Seguidament es comprova que no s'hagi produït un error degut a que el token és caducat i en cas de que sigui així es fa una crida al fil principal per a que es demani el refresc del token.

Si s'arriba al punt final, entenem que s'han rebut dades correctes i per tant s'afegeix el rld en cas de que hagi estat proporcionat i es fa una crida al sistema de notificacions amb la notificació sol·licitada i les dades com a paràmetre de la notificació.

### **Sistema de notificacions**

Donat que el sistema implementat basa la obtenció de dades a la connexió i obtenció de dades de un servei extern al que es connecta mitjançant Internet, es fa inviable plantejar un sistema síncron que quan l'usuari sol·liciti les dades es connecti, esperi la resposta, obtingui les dades i les processa doncs això bloquejaria constantment la interfície d'usuari fent que resultés una experiència d'ús nefasta.

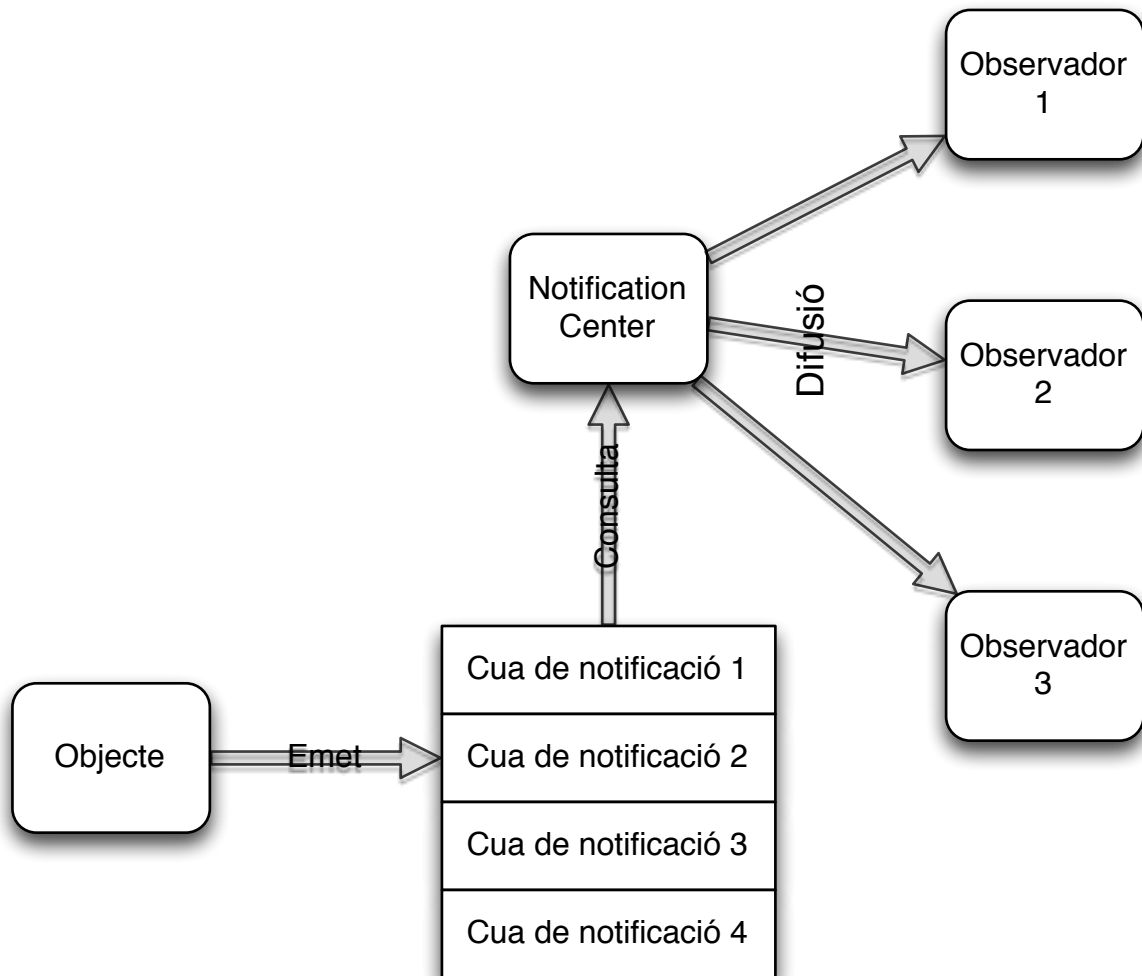
A més de l'anterior situació, també es planteja una segona situació que és el fet que en rebre noves dades son varis el components que necessiten actualitzar les seves dades en base a la nova informació obtinguda, no essent viable fer crides seqüencialment als diferents components que necessiten les dades doncs això ens limitaria alhora de afegir nous components i trencaria la independència de les capes que hem presentat anteriorment al apartat **4.3.4 - Utilització dels patrons de disseny MVC, Singleton i Delegate**.

De les possibles solucions que es plantegen, sempre passant per una execució asíncrona de les crides i processament de dades, s'ha optat per utilitzar el sistema de notificacions que ofereix el framework amb la classe *NSNotificationCenter*. Es tracta d'un sistema de difusió que permet subscriure objectes a determinades notificacions de forma que quan un objecte emet una notificació d'un tipus determinat, tots els objectes que estan observant aquesta notificació reben una crida al mètode que hagin indicat com a resposta davant una notificació del tipus indicat.

L'esquema més entenedor del sistema de notificacions és el mostrat a la figura 4.4.6



Figura 4.4.6 Sistema de notifikacions d'iOS



El principal objecte que emet les notifikacions, com és lògic, en aquest sistema és la classe **OpenAPI** per a accedir a les dades de la **OpenAPI** de la **UOC**. Les notifikacions que emet aquesta classe son:

- **OpenAPILoginCompleted** - Emesa quan s'ha realitzat el registre a la **OpenAPI** amb èxit.
- **OpenAPILoginFailed** - Emesa quan hi ha hagut un error registrant-se a la **OpenAPI**.
- **OpenAPIRequestFailed** - Emesa quan hi ha hagut un error realitzant una connexió.
- **OpenAPIEventListLoaded** - Emesa quan s'ha carregat una nova llista d'activitats.
- **OpenAPIEventListBDLoaded** - Emesa quan s'ha carregat una nova llista d'activitats entre dues dates pre-establertes.
- **OpenAPIEventLoaded** - Emesa quan s'han carregat les dades d'un event.

- **OpenAPISubjectListLoaded** - Emesa quan s'ha carregat una nova llista de assignatures.
- **OpenAPINewResourcesForSubjectLoaded** - Emesa quan s'ha obtingut una nova llista de recursos (documents) per a una assignatura concreta.
- **OpenAPIClassroomsListLoaded** - Emesa quan s'ha obtingut una nova llista d'aules.
- **OpenAPINewMaterialsForClassroomLoaded** - Emesa quan s'ha obtingut una nova llista de recursos (documents) per a un aula concreta.

#### 4.4.4. Incorporació dels dades a la base de dades

Donat que l'únic procés d'emmagatzemament de dades utilitzant *Core Data* es fa per a les descripcions de les activitats, es considera interessant detallar el procés.

El procés d'incorporació de la base de dades es realitza al mètode *saveSummary* de la classe *GlobalDataViewController* que és el controlador de la pantalla principal d'activitats.

En sol·licitar que s'emmagatzemin les dades de l'activitat seleccionada fent tap en la imatge d'un disk de 3.5 ", es realitza la comprovació que no s'està intentant emmagatzemar una nota buida i si no és així es crea un objecte de cerca per a entitats del tipus "Activity". Es cerquen totes les "Activity" amb identificador igual al *id* de l'activitat seleccionada.

En realitzar la consulta, establim el nou valor per a la propietat *summary* i emmagatzemem el resultat.

Es un procés senzill gràcies a l'utilització de Core Data.

```
// Comprovar que no es passa un string buit
if ([selectedActivity isEqualToString:@""]) {
    // Informar de l'error i retornar
    return;
}

// Crear l'objecte de cerca i l'entitat a cercar
NSFetchRequest *fetchRequest = [[NSFetchRequest alloc] init];
NSEntityDescription *entity = [NSEntityDescription entityForName:@"Activity"
                               inManagedObjectContext:self.managedObjectContext];
[fetchRequest setEntity:entity];

// Establir el criteri de cerca
[fetchRequest setPredicate:[NSPredicate predicateWithFormat:@"activity == %@",
                                                             selectedActivity]];

// Executar la cerca
NSError *error;
NSArray *resultList = [self.managedObjectContext executeFetchRequest:fetchRequest
                                                             error:&error];

NSManagedObject *moActivity;
```

```
if ([resultList count]<=0)
{
    // Informar de l'error i retornar
}
else
    moActivity =[resultList objectAtIndex:0];

// Actualitzar els valors
[moActivity setValue:self.activityNote.text forKey:@"summary"];

// Emmagatzemar
if (![self.managedObjectContext save:&error]) {
    // Informar de l'error i retornar
}
```

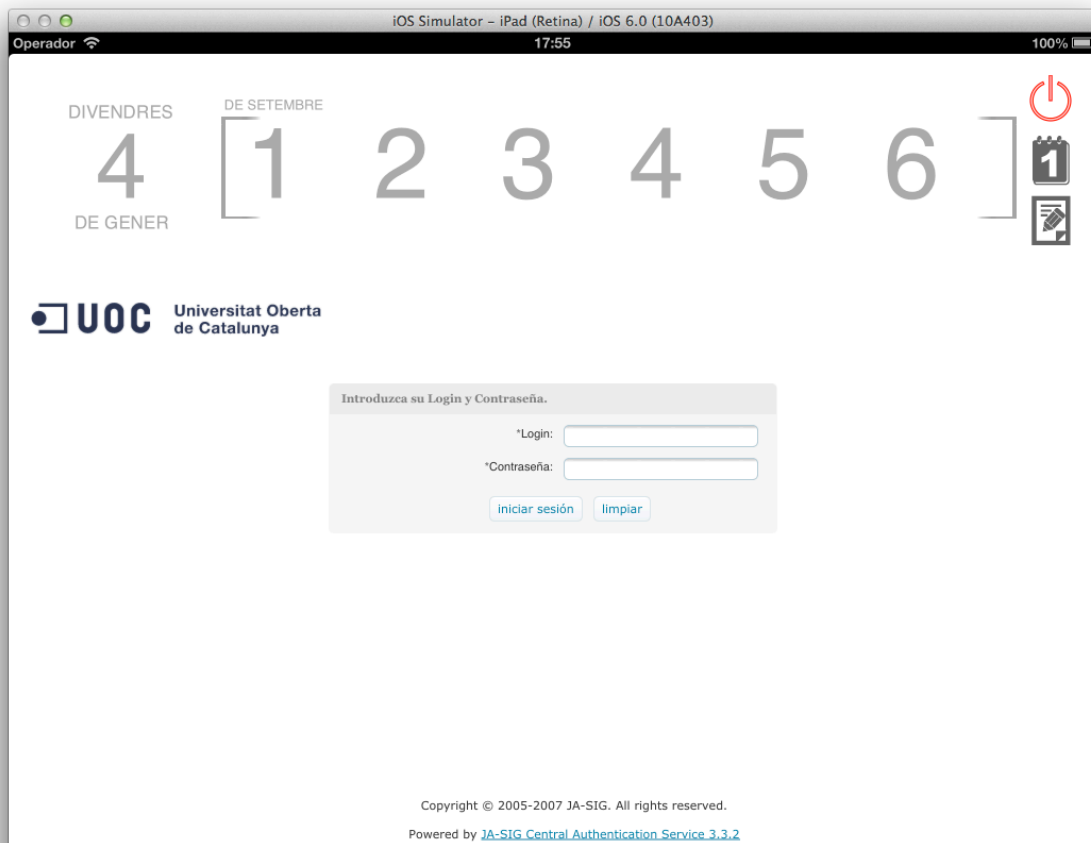
#### 4.4.5. Presentació del funcionament de l'aplicació iUOC

El funcionament de l'aplicació es basa en tres espais específics que són la identificació, les dades de activitats i les dades de documents. Els següents apartats expliquen detalladament cadascun dels apartats, la seva funcionalitat i interaccions amb l'usuari.

## Pantalla inicial

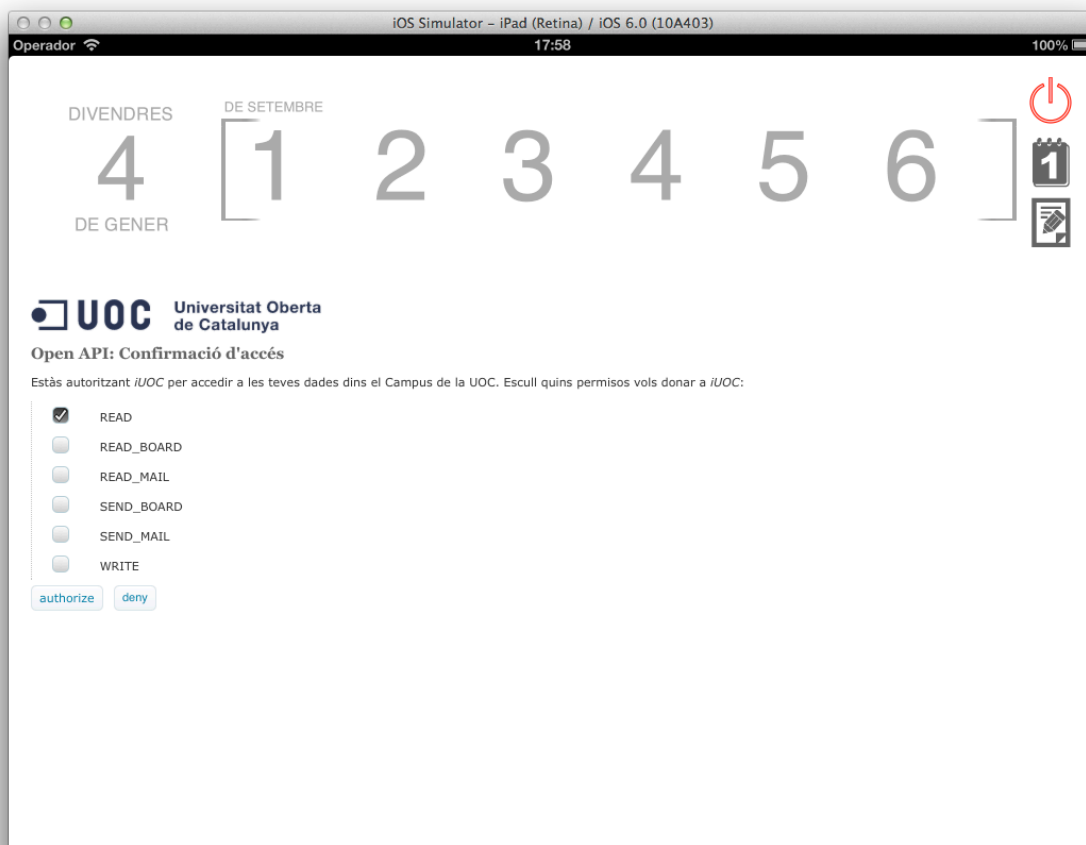
La primera pantalla que es presenta en iniciar l'execució de la aplicació per primer cop, és la pantalla general amb la plana web d'identificació de la **OpenAPI** de la **UOC** a fi de que l'alumne s'identifiqui per primer cop, donat que sense les dades de l'usuari l'aplicació no és capaç de mostrar cap mena de resultat.

**Figura 4.4.7** Pantalla inicial d'iUOC



En identificar-se, la **OpenAPI** de la *UOC* mostrarà una plana a la que hem de triar els permisos que concedim a la aplicació iUOC per a obtenir les dades relatives a l'alumne.

**Figura 4.4.8 Pantalla d'autoritzacions d'iUOC**



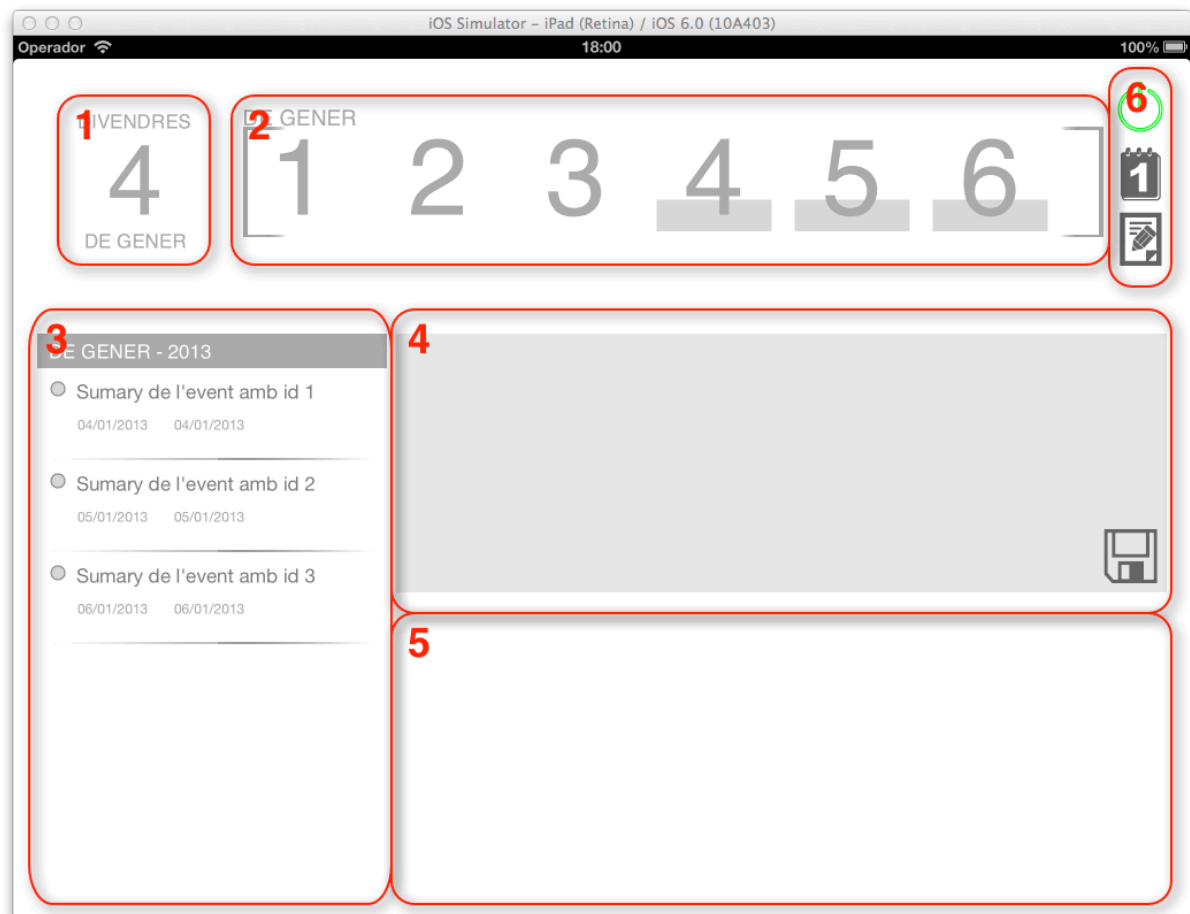
Com es pot comprovar a la imatge, l'únic permís requerit, que és el de lectura, ja és seleccionat. Si be es poden donar la resta de permisos, l'aplicació no el requereix en aquest moment.

En autoritzar-ho es passarà a la pantalla d'activitats.

### **Pantalla d'Activitats**

La pantalla d'activitats és on es mostren les dades relatives a les activitats de les assignatures de les que l'alumne és matriculat. Aquestes activitats es mostraran agrupades per mesos i amb el títol de l'activitat, la data d'inici de l'activitat i la data de finalització de l'activitat. També permetrà accedir als detalls de l'activitat per a la seva visualització i modificació en el cas del sumari de l'activitat.

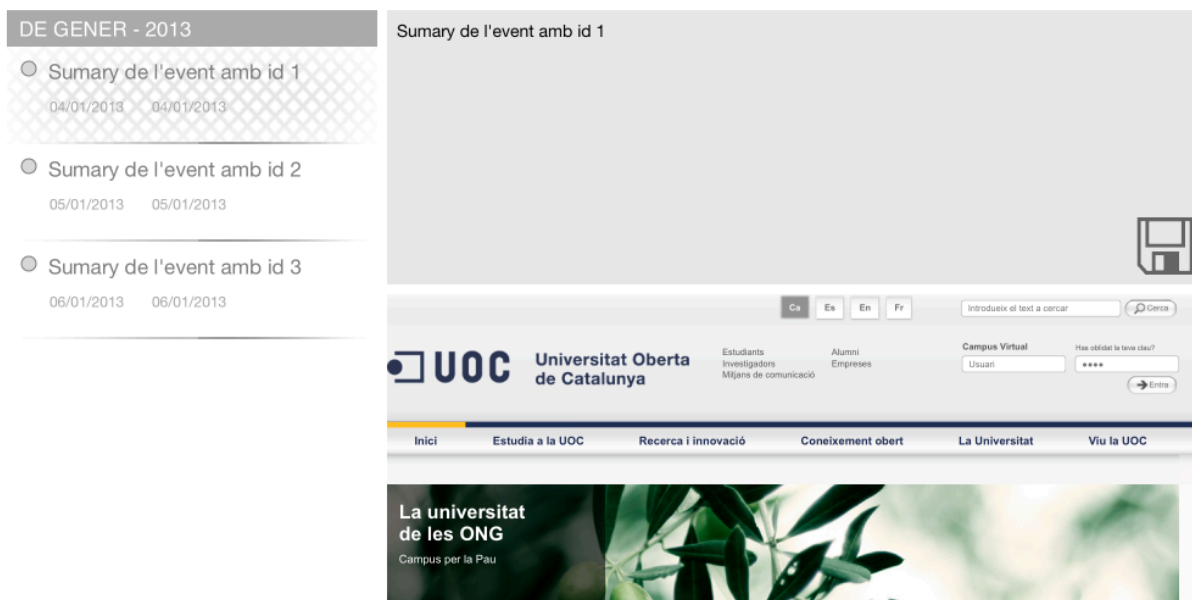
Figura 4.4.9 Pantalla d'activitats d'iUOC



1. Calendari que mostra la data seleccionada, inicialment mostra la data del dia en curs.
2. Control per a la selecció del dia del qual es volen veure les activitats
3. Llista d'activitats agrupades per mesos. En fer tap a una de les activitats ens mostrarà els detalls a les zones 4 i 5.
4. Sumari de l'activitat seleccionada. En cas de fer cap modificació al sumari, podem emmagatzemar-ho al dispositiu fent tap a la icona del disquet de tres polzades i mitja.
5. Control web que mostra el contingut de la plana web a la que apunta la url de l'activitat seleccionada.
6. Botons de control de connexió i desconnexió del servei, refresc i visualització de les activitats i refresc i visualització dels documents de les assignatures.

En seleccionar una activitat, fet tap en una de les files de la taula de la zona 3, aquesta es marcarà com a seleccionada i ens mostrarà a les zones 4 i 5 la informació relativa a la activitat. Donat que el la versió de test de la **OpenAPI** no es proporcionen url adequades, s'ha optat per mostrar sempre la plana inicial de la *UOC*.

**Figura 4.4.10 Pantalla de detall d'una activitat d'iUOC**



Els botons de la zona 5 permetran a l'usuari realitzar les següents accions:



El botó de connexió i desconnexió, que inicialment és de color vermell, indicarà si l'aplicació es troba o no connectada a la **OpenAPI**. En cas de mostrar-se de color verd, vol dir que estem validats i connectats. En cas de mostrar-se de color vermell indicarà que no hi ha connexió i que per tant les dades no es poden mostrar ni actualitzar. En cas de prémer el botó quan s'està connectat es desconnectaria del servei i en cas de prémer quan s'està desconnectat es presentaria la pantalla inicial amb la plana d'autorització de la **OpenAPI**.



El botó d'activitats permet refrescar les activitats connectant-se a la **OpenAPI** i obtenint les darreres dades d'activitats, o, en cas de que la pantalla que es mostra no sigui la d'activitats, es mostri la pantalla d'activitats amb les activitats actualitzades.

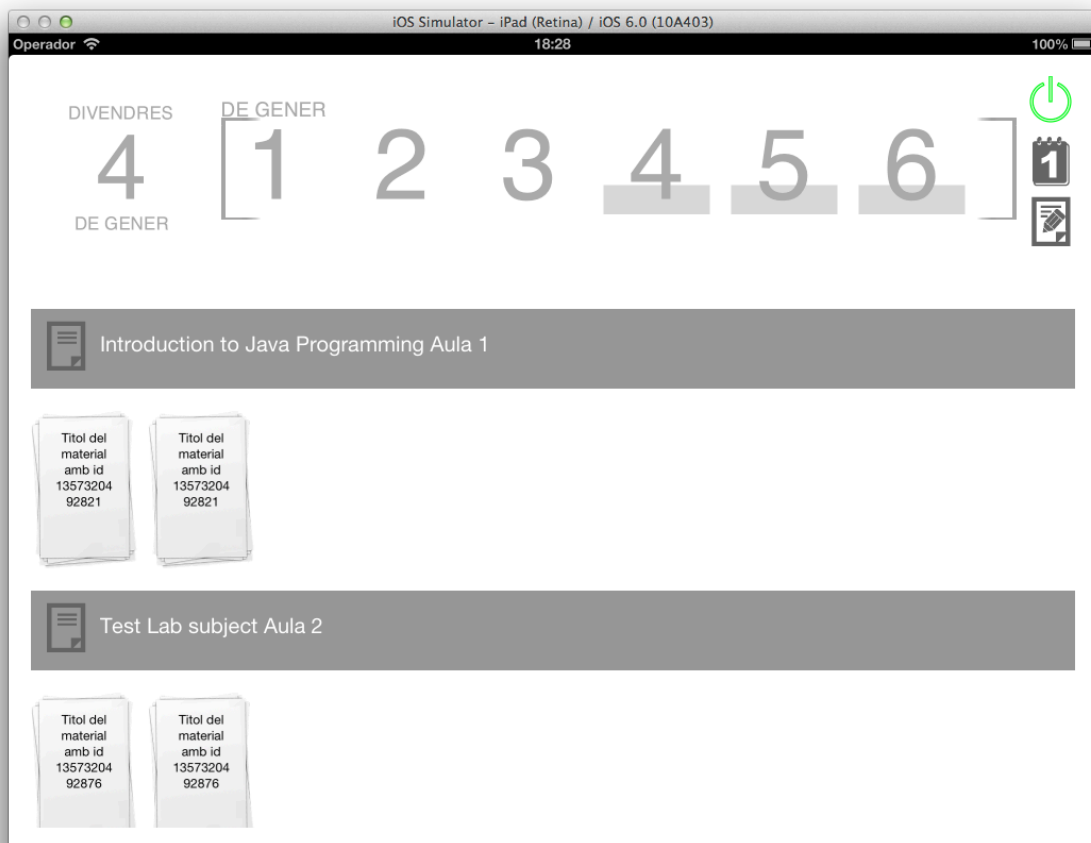


El botó de documents permet refrescar els documents de les assignatures a les que l'alumne és matriculat connectant-se a la **OpenAPI** i obtenint les darreres dades, o, en cas de que la pantalla que es mostra no sigui la de documents de les assignatures, es mostri la pantalla de documents de les assignatures amb les dades actualitzades.

## Pantalla de Documents

La pantalla de documents presenta els documents de les aules a les que es troba inscrit l'alumne dins les assignatures de les que es troba matriculat en un semestre donat. Els documents es presenten en forma d'icones contenint el títol del document i agrupades per aules.

**Figura 4.4.11 Pantalla de documents d'iUOC**



En fer tap a un dels documents, es presentarà un diàleg modal contenint el contingut del document per a la seva lectura.



**Figura 4.4.12 Pantalla de lectura de document d'iUOC**



Un cop finalitzada la lectura, es pot fer tap sobre l'icona de les aspes de la cantonada superior dreta del document i així es tancarà el diàleg de lectura i es retornarà a la pantalla de documents.

La aplicació està dissenyada fent servir la tecnologia d'*autolayouts*, per la qual cosa les pantalles anteriorment descrites es reorganitzen de forma controlada en tenir el dispositiu en posició vertical en lloc de en posició apaïxada.

Figura 4.4.13 Pantalla d'activitats d'iUOC en posició vertical

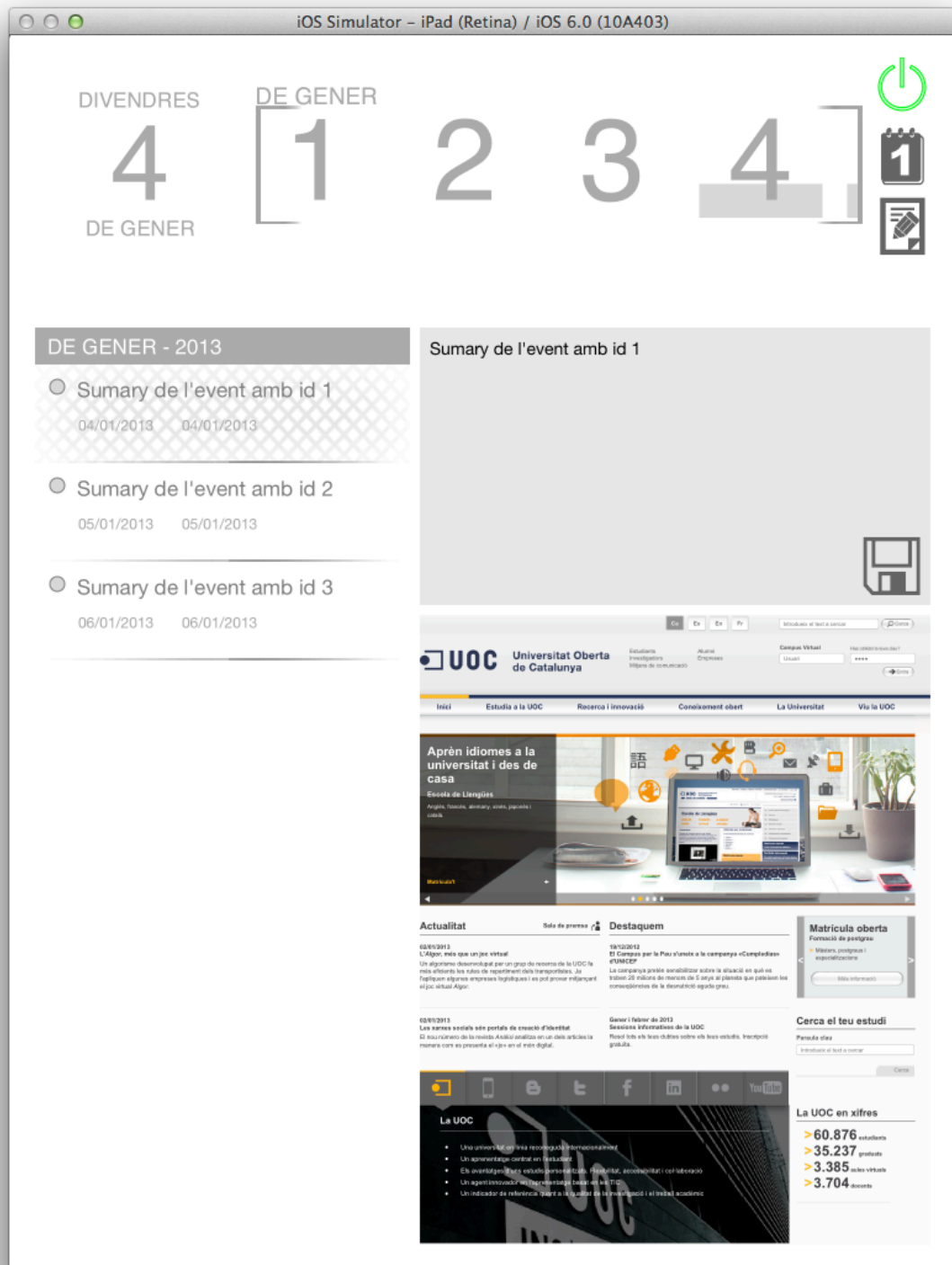


Figura 4.4.14 Pantalla de documents d'iUOC en posició vertical

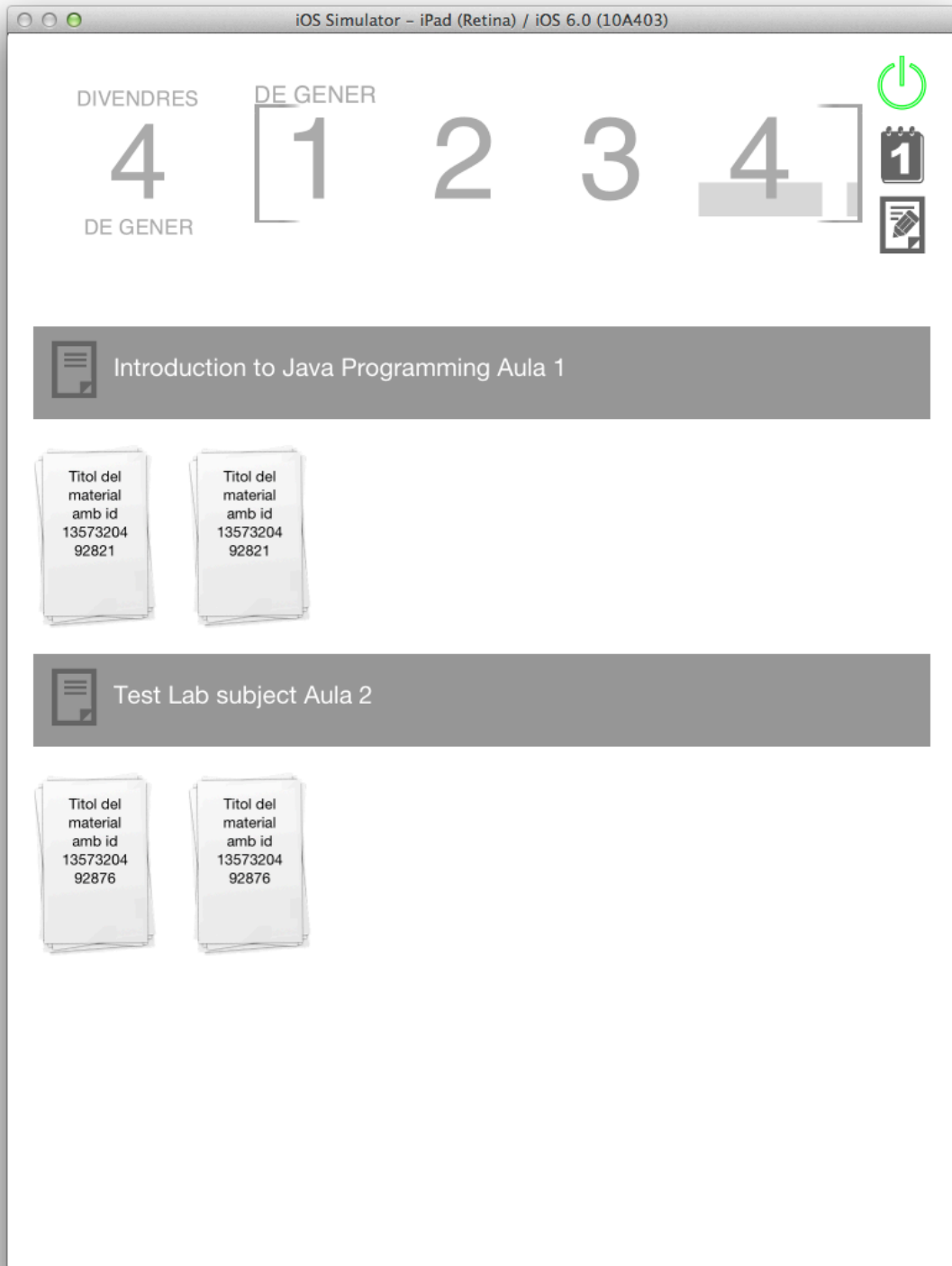
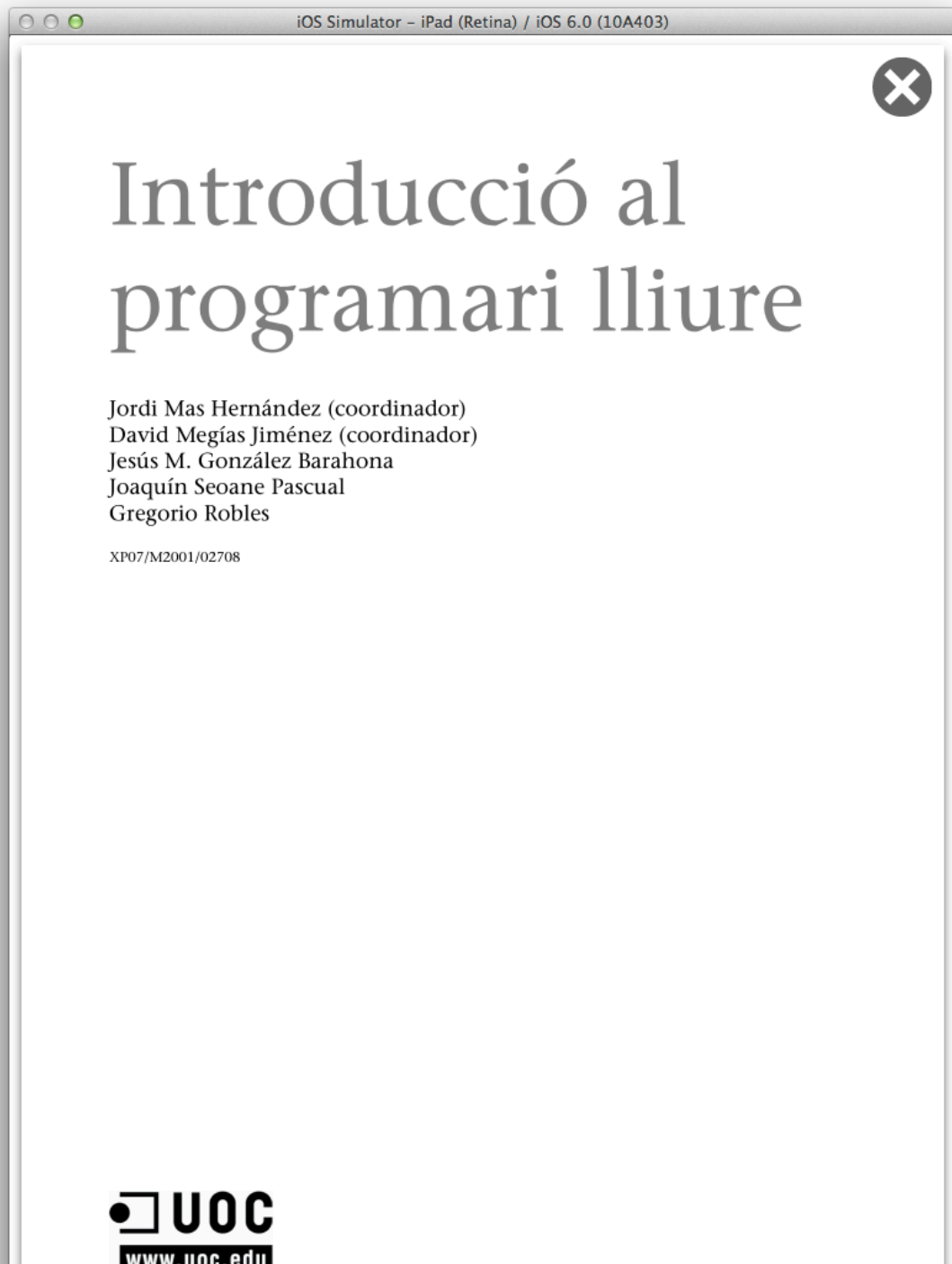


Figura 4.4.15 Pantalla de lectura de document d'iUOC en posició vertical



#### 4.4.6. Futures funcionalitats i millores per a iUOC

Per a millorar l'aplicació iUOC inicialment es presenten tres línies principals. En primer lloc, una millora evident és la estandardització de la llibreria per a accedir a la OpenAPI de la UOC, doncs aquesta primera versió està força cohesionada amb l'aplicació i no compleix amb els requeriments d'una API pública que pugui ser extreta de l'aplicació. Un cop estandarditzada com a API resultarà molt més senzill realitzar qualsevol tipus de afegit en els components de l'aplicació.

La segona línia de millora és el permetre accedir a les modificacions de les activitats al la **OpenAPI** de la UOC, no només fer anotacions en local si no enviar aquestes anotacions als serveis de la UOC i poder així recuperar des de qualsevol localització els canvis.

La darrera línia consisteix en la implementació real del suport offline, és a dir, l'emmagatzemament de totes les dades que s'obtenen de la OpenAPI, de forma que quan no es disposa de connexió es pugui seguir utilitzant l'aplicació per a gestionar les dades i informació de la UOC.

Hi ha evidentment moltes altres millores, però que ja passarien per un redisseny de l'aplicació donat que impliquen afegir noves pantalles com ara el suport per al correu, els foros, etc ...

## 5. Conclusions

Les principals conclusions que es poden extraure d'aquest projecte, segons la experiència obtinguda, són que una planificació massa ambiciosa a l'inici és un error que es pot cometre molt fàcilment i que porta a anar retallant funcionalitats del projecte fins a tenir una definició d'aplicació realitzable en el temps del que es disposa. Iniciem el projecte volent donar tot el que tenim dins i fent una aplicació com la que en agradaria trobar en el mercat i la realitat, com sempre, ens ensenya que els productes que tenim al mercat són el resultat de moltes iteracions a les quals es va millorant poc a poc el producte, que per un altre banda mai serà tant perfecte com ens agradaria, però que en algun moment tenim que posar en circulació.

Un altre conclusió que puc extraure és que la planificació del projecte a l'estil tradicional quan es treballa sol no és el mètode més efectiu, doncs no tenim recolzaments en altres companys i els buffers que podem inserir no són gairebé mai realistes, ben al contrari és molt més efectiu un enfoc de tipus àgil en que ens anem plantejant tasques individuals i podem anar adaptant el resultat final a l'esforç que hem pogut realitzar.

La darrera conclusió que extrec és que hi ha un món apassionant davant de les eines per a estudiar i que pot sortir un bon producte de la idea plantejada i que per tant serà un producte a evolucionar fins a tenir, com ja he comentat, el producte que m'hagués agradat tenir per a estudiar a la UOC.

## 6. Glossari

**Activitats** - Conjunt de les diferents activitats programades per a les assignatures de les que un estudiant pot estar matriculat a la UOC. Aquestes comprenent tant les PAC com les pràctiques o els inicis de mòdul per exemple.

**App Store** - Botiga d'aplicacions mitjançant la qual es poden distribuir les aplicacions desenvolupades per a dispositius Apple amb el sistema operatiu iOS tals com iPhone, iPad o iPod.

**Autenticació** - Procés mitjançant el qual un usuari s'identifica en un sistema. Hi ha molts diversos tipus d'autenticació, fins i tot automàtica si s'han emmagatzemat les dades necessàries per a realitzar-la.

**Autolayouts** - Tecnologia creada per Apple que permet inserir les subvistes dins una vista de forma que s'estableixen relacions de posició entre elles i respecte a la vista superior. Això permet que en canviar la morfologia de la pantalla les vistes es reordenin de forma automàtica sense perdre la presentació i organització desitjades.

**Cocoa** - Framework de desenvolupament creat per a permetre el desenvolupament d'aplicacions per al sistema operatiu OS X, en la seva versió touch està creat per a desenvolupar aplicacions per a dispositius iOS.

**Core Data** - Tecnologia creada per Apple per a facilitar l'emmagatzemament de dades en diferents sistemes i encapsular el desenvolupament de manera que s'aïlli la forma de persistència del model de dades triat.

**DCU** - veure Disseny Centrat Usuari.

**Disseny Centrat en l'Usuari** - Metodologia de disseny que es focalitza en comprendre les necessitats limitacions, comportament i característiques de l'usuari, involucrant en molts casos a usuaris potencials o reals en el proces.

**Editable** - Possibilitat d'editar quelcom.

**Emmagatzemable** - Possibilitat d'emmagatzemar quelcom.

**Framework** - Conjunt de classes, funcions o mètodes creats i agrupat per a facilitar la feina en determinats aspectes del desenvolupament. És un terme adoptat de l'anglès i equival a entorn de treball o, també, marc de treball.

**Grant** - Garantir, permetre. Consisteix al fet de garantir l'accés a un servei o recurs.

**Hardcodejat** - Inclòs de forma textual al codi, i que per tant no és possible canviar-ho sense compilar de nou l'aplicació.

**Interfície** - medi amb què l'usuari pot comunicar-se amb una màquina, un equip o una computadora, i comprèn tots els punts de contacte entre l'usuari i l'equip.

**iOS** - Sistema operatiu dels dispositius d'Apple iPad, iPhone i iPod Touch.

**iPad** - Tauleta electrònica de la marca Apple que corre el sistema operatiu iOS.

**Markdown** - és un llenguatge de marques lleuger, originalment creat per John Gruber i Aaron Swartz que permet "escriure utilitzant un format de text planer fàcil d'escriure i de llegir i després convertir-ho en un XHTML o HTML.

**Managed Object** - Entitat d'una base de dades gestionable des de Core Data.

**MultiMarkdown** - Ampliació de Markdown.

**OAuth2** - És un protocol obert, proposat per Blaine Cook y Chris Messina, que permet la autorització segura d'una API d'un mode estàndard i simple per a aplicacions d'escriptori, mòbils i web de forma que no s'exposin les credencials de l'usuari.

**OpenAPI** - Servei ofert per la UOC per a accedir a la informació dels alumnes.

**Parser** - Desenvolupament que permet recórrer un format de dades per a adaptar-ho a un altre format. En el present projecte es recorre un format JSON per a adaptar-ho a un format NSDictionary.

**Publicabilitat** - Capacitat de ser publicat.

**SQLite** - És una base de dades relacional continguda en una llibreria escrita en C. A diferència d'altres bases de dades relacionals, SQLite no és un sistema de gestió de base de dades que funcioni amb un paradigma client-servidor. SQLite no és un procés autònom, sinó que s'integra dins d'altres programes.

**Store** - Medi d'emmagatzematge de dades.

**Testejable** - Que permet que es passin una serie de tests que permetin comprovar la seva validesa, integritat o solidesa.



**TDD** - Veure Test Driven Development.

**Test Driven Development** - És una pràctica de programació que involucra altres dues pràctiques: Escriure les proves primer (Test First Development) y refactorització (Refactoring) després.

**Threads** - És la unitat més petita de processament que pot ser programada pels sistemes operatius i que permet a un procés executar diferents tasques al mateix temps.

**Token** - És un codi numèric generat de forma pseudoaleatòria i que es fa servir per a comprovacions d'identitat posteriorment.

**Usabilitat** - Que és susceptible de ser utilitzat d'una forma senzilla o intuïtiva.

**User defaults** - Paràmetres de configuració d'usuari utilitzats per a la execució d'una aplicació seguint les preferències de l'usuari.

**XCode** - IDE de desenvolupament utilitzat per a la programació d'aplicacions per a dispositius amb el sistema operatiu iOS o OSX.

## 7. Bibliografia

**UOC** (2012, setembre). Redacció de textos científicotècnics. Documentació de l'assignatura TFC.

**UOC** (2012, setembre). Disseny centrat en l'usuari. Documentació de l'assignatura TFC.

**Javier Velasco M.** (2005). Qué es el Diseño de Interacción. Mantruc.com publicaciones.

**UOC** (2012, setembre). Documentació OpenAPI. <http://blogs1.uoc.es/developer>.

**Apple** (2012, setembre). Documentació del Framework Cocoa. <http://developer.apple.com/library/ios/navigation/>

**Theresa Neil** (2012). Mobile Design Pattern Gallery. O'Reilly

**Ryan Boyd** (2012). Getting Started with OAuth 2.0. O'Reilly

**Theresa Neil** (2012). Mobile Design Pattern Gallery. O'Reilly

**Fletcher T. Penney** (2010). MultiMarkdown User's Guide 2.0.b6. <http://fletcherpenney.net/>

**Joe Conway and Aaron Hillegass** (2010). iPhone Programming: The Big Nerd Ranch Guide. Pearson Technology Group

# 8. Annexos

## **ANNEX I**

Formulari presentat per a obtenir les dades d'usuari

## **ANNEX II**

Resultats de l'enquesta d'usuari

## **ANNEX II**

Llicència d'ús de WCAAlertView

Copyright (c) 2012, Michał Zaborowski

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the Software.