



# iCompound

*Una aplicació per l'estudi  
de la química sobre iOS*

Josep Hornos Arias  
Informàtica de Gestió ETIG – UOC

Àrea TFC: *Disseny d'aplicacions per a iOS*  
Consultor: *Jorge Márquez Moreno*  
Data de lliurament: *13 de gener del 2012*

*Gràcies a tots aquells que creuen en mi  
i que m'han ajudat a fer més lleu la consecució d'aquesta carrera.  
Els que de veritat han estat, són i seran importants per mi.*

*Ells saben qui són.*

*I també gràcies a Freddie, Rob, Kai, Mario...  
i tants d'altres que han posat banda sonora als meus esforços.*

## Resum

Recordo, quan era petit i llegia relats del mestre Isaac Asimov sobre entitats artificials intel·ligents (basades en cervells positrònics) i d'altres enginys electrònics, tenir la sensació que tota aquella fantasia futurista quedava ben lluny del nostre abast. Quan parlava sobre ordinadors de butxaca, i mirava el meu AMSTRAD CPC 464 amb monitor de fòsfor verd, un somriure d'incredulitat m'envaïa.

Anys després, però, la realitat ha superat a aquella ficció de la meva infantesa. Els grans avenços tecnològics, especialment en l'àmbit de la computació i les ciències de la informació han fet possible tenir tota la potència d'un gran computador en dispositius tant petits com un telèfon mòbil. I sense cap mena de dubte, el màxim promotor i exponent ha estat Apple, gràcies al desenvolupament dels telèfons intel·ligents iPhone.

En l'actualitat, aquests dispositius han evolucionat notablement, oferint a l'usuari dispositius de gran versatilitat, amb innovacions focalitzades en la usabilitat i satisfacció d'usuari (com per exemple, controls 100% tàctils). I tenint en compte la gran acceptació i mercat que tenen, resulta interessant poder desenvolupar aplicacions d'utilitat destinades als seus usuaris.

El treball de final de carrera és la culminació de molts anys d'estudi, i una oportunitat excel·lent de posar en pràctica molts dels coneixements adquirits. A més, poder treballar en l'àrea que ocupa (*disseny d'aplicacions per iOS*), ha estat la millor oportunitat per intentar crear una eina de qualitat per aquelles persones que, essent usuaris de dispositius mòbils d'Apple (iPhone, iPod, iPad), i estant lligats (d'una manera o altre) al món de la química, vulguin tenir a la seva disposició una aplicació que els ajudi a resoldre petits problemes lligats a aquesta disciplina científica.

Aquesta memòria reflexa l'esforç realitzat al respecte.

## Paraules clau

*iPhone, iPod, iPad, Objective-C, iOS, Cocoa-Touch, X-Code, Interface Builder, Model View Controller, OpenGL-ES, Core Graphics, Human Interface Guidelines, User Centered Design*

## Estructura de la memòria

Aquesta memòria s'estructura en diferents capítols, cada un dels quals pretén tractar sobre una de les diferents tasques associades al cicle de vida d'un disseny informàtic.

En el primer capítol, *Pla de treball*, es pretén donar una justificació del perquè s'ha decidit fer el desenvolupament de *iCompound*, tot el procés de generació i desenvolupament de la idea, descripció de l'enfocament utilitzat i planificació de la feina a fer.

El segon capítol, *Anàlisi de requeriments*, aprofundeix en les tasques prèvies a qualsevol disseny, com són l'anàlisi dels potencials usuaris i l'estudi de *benchmarking*, per tal de poder definir les línies mestres a seguir (especificació de requeriments i funcionalitats de l'aplicació).

Al capítol tercer, *Disseny*, es recull tot el procés seguit per posar l'estudi fet al capítol dos sobre unes bases teòriques fermes, que definiran tant l'estructura de l'aplicació (persistència, prototipat, disseny d'interfície) com els comportaments que ha de tenir (navegació, interacció/casos d'ús).

El capítol quart, *Implementació*, descriu com s'ha passat tot el disseny teòric a codi font. Les variacions, decisions preses, així com una descripció detallada dels passos seguit per programar l'aplicació troben en aquest apartat la seva raó de ser.

*Integració i proves* és el capítol cinquè, on es descriu el conjunt de proves duts a terme per avaluar l'aplicació segons els diferents criteris establerts al primer i segon capítol, permetent així aplicar les correccions pertinents en cada cas. A més, pretén deixar constància de les millores i plans de futur respecte l'aplicació desenvolupada.

Finalment, el sisè capítol, *Conclusions*, recull les impressions i fets més destacables succeïts al llarg del procés de realització del TFC.

Com a punt i final, es pot trobar un glossari amb els termes més importants relacionats amb el treball, una bibliografia que recull les fonts emprades per dur a terme les diferents tasques, i uns annexos on es pot consultar informació relacionada que, quedant fora de l'abast d'aquesta memòria, resulta interessant de consultar per entendre certes decisions preses.

## Índex de continguts

1 Pla de treball.....	11
1.1 Descripció del treball.....	11
1.2 Justificació del TFC i context en el que es desenvolupa.....	12
1.3 Objectius generals i específics.....	12
1.4 Descripció de l'aplicació. Justificació del nom proposat.....	13
1.4.1 Elecció del nom.....	13
1.4.2 Descripció.....	13
1.5 Enfocament i mètode utilitzat (DCU, anàlisi, disseny i avaluació iterativa).....	14
1.6 Planificació del projecte.....	14
1.6.1 Definició del projecte i planificació.....	15
1.6.2 Definició i anàlisi de requeriments.....	15
1.6.3 Disseny.....	16
1.6.4 Implementació.....	17
1.6.5 Integració i proves.....	17
1.6.6 Documentació final.....	18
1.6.7 Possibles problemes i desviacions del pla de treball.....	20
2 Anàlisi de requeriments.....	21
2.1 Perfils d'usuari.....	21
2.1.1 Primeres aproximacions.....	21
2.1.2 Indagació i requisits d'usuari.....	22
2.1.3 Determinació dels personatges.....	23
2.1.4 Punts en comú entre els personatges.....	25
2.2 Aplicacions relacionades amb la química (elements i compostos). Benchmarking.....	26
2.2.1 iElements.....	26
2.2.2 Merck PSE HD.....	28
2.2.3 Molecules.....	32
2.2.4 Conclusions de l'estudi.....	33
2.3 Definició de funcionalitats i tasques.....	34
2.3.1 Funcionalitats bàsiques.....	34
2.3.1.1 Relacionades amb els elements i la taula periòdica.....	34
2.3.1.2 Relacionades amb el mòdul de molècules.....	35
2.3.2 Funcionalitats avançades.....	35
2.3.2.1 Relacionades amb els elements i la taula periòdica.....	36

2.3.2.2	Relacionades amb el mòdul de molècules .....	36
2.4	Estudi d'usabilitat.....	36
3	Disseny.....	39
3.1	Esquema de navegació.....	39
3.2	Persistència.....	40
3.3	Interacció: casos d'us.....	42
3.3.1	Elements: taula periòdica i calculadora molar.....	43
3.3.2	Molècules i visualitzacions.....	47
3.4	Prototipat de baix nivell.....	52
3.4.1	Inici de l'aplicació.....	52
3.4.2	Taula Periòdica.....	53
3.4.2.1	Format taula.....	53
3.4.2.2	Format llista.....	53
3.4.2.3	Propietats de l'element seleccionat.....	54
3.4.3	Calculadora molar.....	55
3.4.4	Visor de molècules.....	55
3.4.4.1	Llistat de molècules. ....	55
3.4.4.2	Mode d'edició.....	56
3.4.4.3	Esborrat de molècules.....	56
3.4.4.4	Inserció de molècules.....	57
3.4.4.5	Visor de molècules.....	57
3.5	Disseny de la interfície .....	58
4	Implementació.....	59
4.1	Consideracions prèvies .....	59
4.1.1	Patrons de disseny.....	59
4.1.1.1	Model vista controlador (MVC).....	59
4.1.1.2	Delegat (delegate).....	60
4.1.2	Gestió de la memòria en dispositius mòbils amb iOS.....	60
4.1.3	Persistència.....	61
4.2	Decisions preses en el procés d'implementació.....	62
4.3	Implementació del subsistema Model.....	63
4.3.1	Persistència dissenyada. Modificacions.....	63
4.3.2	Implementació de la persistència.....	64
4.3.2.1	Creació de taules.....	64
4.3.2.2	Creació de triggers i restriccions.....	64
4.3.2.3	Inicialització de dades.....	66
4.3.3	Gestió de la persistència i classes associades.....	68

4.4 Implementació del subsistema View.....	74
4.4.1 Variacions respecte el disseny teòric.....	74
4.4.2 Desenvolupament final .....	76
4.5 Implementació del subsistema Controller.....	83
4.5.1 Mòdul d'elements.....	84
4.5.1.1 ICElementsController.....	84
4.5.1.2 ICElementsOrderCriteria.....	85
4.5.1.3 ICElementsTable.....	86
4.5.1.4 ICElementsBasicController.....	87
4.5.1.5 ICElementsAtomicController.....	88
4.5.1.6 ICAtomViewController.....	88
4.5.1.7 ICElementsAdvancedController.....	90
4.5.2 Mòdul de Calculadora Molar.....	91
4.5.2.1 ICCalculatorController.....	91
4.5.3 Mòdul de Molècules.....	92
4.5.3.1 ICMoleculesController .....	92
4.5.3.2 ICSimpleMoleculeController .....	96
4.5.3.3 simpleMolecule .....	96
4.5.3.4 ICMoleculeInfoController .....	97
4.5.3.5 ICMoleculesSearchController.....	98
5 Integració i proves.....	99
5.1 Verificació de funcionament.....	99
5.1.1 Consideracions prèvies.....	99
5.1.2 Detecció de problemes sobre simulador.....	100
5.1.3 Detecció de problemes sobre iPod Touch.....	100
5.2 Avaluació de l'aplicació.....	101
5.2.1 Deficiències detectades i solucions proposades.....	101
5.2.2 Aspectes destacables.....	103
5.3 Resultat final.....	103
5.4 Possibles millores i futur de l'aplicació.....	103
6 Conclusions.....	105
7 Glossari.....	106
8 Bibliografia.....	108

## Índex de figures

Logotip de portada.....	1
Taula periòdica dels elements.....	11
iPhone + iCompound.....	12
Logo de l'aplicació.....	13
Planificació (diagrama Gantt).....	19
iPhone - iPod Touch (utilització per edats).....	21
Logo de iElements.....	26
Pantalla principal de iElements.....	27
Pantalla de propietats de iMolècules.....	27
Logo de Merck PSE HD.....	28
Taula periòdica de Merck PSE HD.....	29
Selecció d'element Merck PSE HD.....	29
Accés a les propietats dels elements Merck PSE HD.....	30
Calculadora molar Merck PSE HD.....	30
Tipus de taula Merck PSE HD.....	31
Logo Molècules.....	32
Molecule Manager Molècules.....	32
Molecule viewer Molècules.....	32
Esquema de navegació per iCompound.....	39
Persistència per iCompound.....	40
Casos d'us elements per la Taula periòdica i Calculadora Molar.....	43
Casos d'us elements pel Visualitzador de molècules.....	48
Pantalla d'inici de iCompound.....	52
Taula periòdica de iCompound.....	53
Taula en format llista + criteris d'ordenació de iCompound.....	53
Selector de propietats de iCompound.....	54
Vistes de propietats de iCompound.....	54
Calculadora molar de iCompound.....	55
Llistat de molècules de iCompound.....	55
Edició de molècules de iCompound.....	56
Esborrat de molècules de iCompound.....	56
Cerca i inserció de molècules.....	57
Visor de molècules de iCompound.....	57

Model Vista Controlador.....	59
Patró Delegate.....	60
Gestió de la memòria en iOS. Alloc-ratein-release cycle.....	61
Persistència per iCompound (millorada).....	63
Esquema de navegació per iCompound (modificat).....	75
Disseny en Interface Builder de la vista principal.....	76
ICElementsView.....	77
ICElementsBasicView.....	77
ICElementsAtomicView.....	78
ICElementsAdvancedView.....	79
ICOrderCriteriaView.....	79
ICElementsTableView.....	80
ICCalculatorView.....	80
ICMoleculesView.....	81
ICSimpleMoleculeView.....	82
ICMoleculeInfoView.....	82
ICMoleculesSearchView.....	83
Vista inicial del llistat d'elements i vista millorada.....	102
Vista inicial de propietats bàsiques d'element i vista millorada .....	102

# 1 Pla de treball

## 1.1 Descripció del treball

Una de les disciplines més fascinants de ciència és sense cap mena de dubte la química. Però com tots sabem, no és fàcil, ja que s'ha d'adquirir una bona base que permeti solucionar problemes de gran envergadura, i de vegades, hom necessita poder tenir a mà certs coneixements elementals per poder consultar-los en cas de dubte.

El primer que un bon químic ha de conèixer i utilitzar és la Taula Periòdica dels Elements. Aquesta eina totalment imprescindible és un compendi dels elements identificats, agrupats i ordenats per diferents propietats i característiques, on es poden trobar diferents paràmetres (nom químic, pes atòmic, etc). Tota aquesta informació és necessària per poder abordar el següent pas: la formulació de compostos.

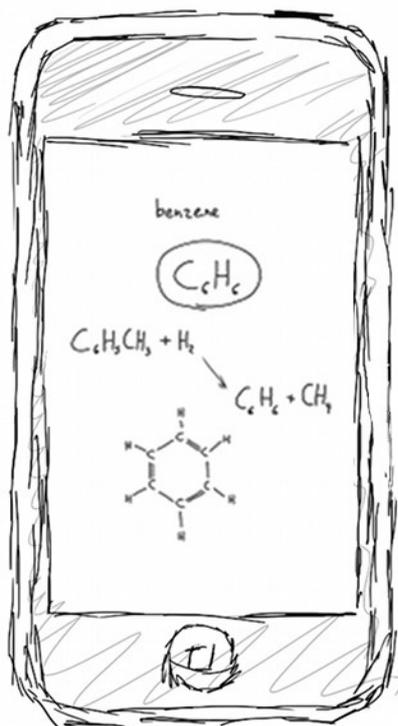
El segon que ha de saber fer un bon químic és formular compostos (orgànics i inorgànics) a partir dels diferents elements. Seguint una sèrie de normes, es poden obtenir molècules (més o menys complexes) que es poden trobar de manera natural a la vida quotidiana: l'aigua (H<sub>2</sub>O), la sal (NaCl), etc. Tot i que molts d'aquests compostos són ben coneguts, el fet d'obtenir-los pot ser una tasca complicada, i encara molt més el fet de demostrar i verificar que les molècules obtingudes responen a algun tipus de compost viable.

Partint d'aquestes dues eines bàsiques, i del fet innegable que la tecnologia ha arribat a les aules, sorgeix la idea de fer una aplicació, en aquest cas per dispositius d'Apple gestionats amb iOS (iPod-iPhone i iPad), la qual pretén ser quelcom més que una eina de consulta. Per una banda, ha de permetre a tot usuari (estudiants, professors, professionals i aficionats a la química) poder fer consultes ràpides en cas de dubte, obtenint informació ampliada sobre els elements (representacions dels àtoms i xifres característiques) i sobre els compostos més comuns existents (amb representacions 2D i 3D de les molècules). D'altra banda, ha de permetre a l'usuari poder verificar formulacions pròpies a partir dels elements, determinar la seva massa molar d'una manera senzilla i intuïtiva.

Imatge 1: Taula periòdica dels elements

## 1.2 Justificació del TFC i context en el que es desenvolupa

Tal i com s'ha explicat en l'apartat anterior, el desenvolupament a realitzar intenta proporcionar una eina d'utilitat basada en dispositius Apple amb iOS al món educatiu, i més concretament, en el de la química.



Imatge 2: iPhone + iCompound

Com ja es sabut, la irrupció d'aquests dispositius a la vida quotidiana és més que una realitat, i per tant, es pretén aprofitar la continua expansió dels mateixos i les possibilitats que ofereix per dissenyar una aplicació tant de consulta com d'estudi i desenvolupament per tot aquell usuari que, d'una manera o altre, hagi d'estudiar o treballar amb elements i compostos químics.

D'aquesta manera, i a través d'una interfície que ha de complir amb les màximes d'usabilitat i utilitat, qualsevol persona interessada en conèixer o consultar la base teòrica d'aquesta disciplina ho podrà fer a través de la taula periòdica dels elements (oferida en diferents formats), obtenint la informació detallada que necessiti. Altrament, serà possible donar un pas endavant, i obtenir igualment resposta a la necessitat de conèixer o verificar tant compostos comuns i ben coneguts, com d'experimentals (a través de les seves representacions gràfiques), amb l'al·licient de poder obtenir paràmetres relatius a la seva massa/pes de manera senzilla i amena.

Així doncs, es presenta un gran repte: crear una aplicació de gran utilitat i versatilitat que, sense cap mena de dubte, pugui fer més senzilles algunes de les tasques tant de professionals com de nous interessats en el món de la química.

## 1.3 Objectius generals i específics

Tenint en compte la naturalesa del projecte, es poden determinar amb claredat els objectius:

- Aprofundir en el procés de desenvolupament de programari, podent treballar en (pràcticament) totes les fases del cicle de vida d'un producte.
- Adquirir coneixements per poder desenvolupar aplicacions per dispositius mòbils d'Apple, a través del llenguatge Objective-C i la utilització tant del *framework Cocoa-Touch* com *Core Graphics* i/o *Open GL ES* (pel modelatge 2D/3D), entre d'altres.
- Elaborar una eina d'utilitat en l'àmbit de l'educació, amb el que això pot comportar (adquisició de nous coneixements relacionats).

- Posar en pràctica i consolidar coneixements adquirits al llarg de la carrera, com poden ser les tècniques de desenvolupament de programari, gestió bases de dades, disseny d'interfícies o modelatge 3D.
- Treballar amb patrons d'arquitectura de programari, i més concretament, amb el *MVC* (*model-view-controller*).
- Conèixer la idiosincràsia del món del desenvolupament d'aplicacions per dispositius mòbils (usabilitat, fiabilitat, disseny de les IPO's, etc).

## 1.4 Descripció de l'aplicació. Justificació del nom proposat

### 1.4.1 Elecció del nom

El nom que es proposa per l'aplicació a dissenyar és *iCompound*. La relació d'idees és immediata: la '*i*' vol ser un indicador del dispositius sobre els quals ha de córrer (Apple *iPod*, *iPhone*, *iPad*), i '*Compound*' és la traducció de compost en anglès. Com es vol treballar amb elements i compostos químics a través d'un conjunt d'eines (que és la novetat respecte a d'altres aplicacions disponibles, la integració), el nom proposat sembla clar i directe.



Imatge 3: Logo de l'aplicació

### 1.4.2 Descripció

Com ja s'ha dit a l'inici, aquesta aplicació pretén ser una eina útil focalitzada en l'àmbit educatiu i divulgatiu, i més concretament, en l'àrea de la química.

Aquest programari es pot descompondre en 3 grans blocs:

- **Taula periòdica dels elements.** L'usuari podrà accedir a la taula periòdica dels elements (actualitzada segons la *IUPAC* el 2011) de dues maneres. La primera, de manera visual, tal i com es coneix (veure *imatge 1* d'aquest mateix document), la qual permetria seleccionar cada element gràcies a les propietats tàctils dels dispositius Apple. La segona, a través d'un llistat dels elements ordenat segons diferents criteris (grup, període o caràcter químic), seleccionables per l'usuari.
- **Base de dades molecular.** L'usuari tindrà a la seva disposició una petita base de dades dels compostos químics més coneguts, focalitzats en la família dels compostos orgànics. Tot ha de ser consultable fent ús d'un llistat organitzat segons els criteris exposats, garantint una navegació senzilla i intuïtiva. Un cop seleccionada la molècula, es tindria accés a una representació en 2D (seguint el format de fórmula desenvolupada) i 3D (model *balls and sticks*), per tal de fer més

entenedora tota la formulació que hi ha al darrera. A més, es donarà la possibilitat a l'usuari de fer una cerca en la bases de dades del NIST per adquirir noves molècules, es quals s'afegiren a les ja existents per defecte.

- **Eina de càlcul de massa molecular.** Aprofitant la possibilitat d'inserir dades gràcies al teclat tàctil (utilitzant Cocoa Touch) es dona la possibilitat a l'usuari d'introduir formulacions pròpies (seguint la nomenclatura IUPAC) i/o conegudes per calcular la seva massa molar total. Amb això és possible tenir un utensili de senzill us i de gran ajuda de cara a realitzar càlculs altrament complexos.

## 1.5 Enfocament i mètode utilitzat (DCU, anàlisi, disseny i avaluació iterativa)

Per poder desenvolupar l'eina descrita, i al ser una peça de programari dintre d'un segment com l'esmentat, no es pot seguir un cicle de vida de desenvolupament clàssic, sinó un en que l'usuari final sigui un valor important. Així doncs, s'opta per emprar el que es coneix com *disseny centrat en l'usuari (DCU)*.

Aquesta filosofia, en la que prima saber posar-se en la pell de l'usuari per garantir temes com la usabilitat, no és pas tant diferent del cicle de vida clàssic. Més aviat és una revisió del mateix en la qual s'introdueix el concepte d'iterativitat en totes les fases, basat en una avaluació continua de cada part del procés. Així doncs, les fases a tenir en compte haurien de ser:

- Anàlisi dels requisits.
- Disseny.
- Implementació (prototipat/final).
- Avaluació.

Com s'ha dit, hi ha una avaluació (4 pas) que s'aplica sobre els anteriors, sempre tenint en compte les necessitats, observacions i proves realitzades sobre usuaris, gràcies en gran mesura a la utilització de prototips. No obstant, tot i que s'intentarà fer un desenvolupament el més fidel possible a aquest model, el temps és un factor primordial a l'hora de poder precisar en cada part. Així doncs, la profunditat amb la qual es fa cada fase, especialment pel que es refereix a iterativitat i interactivitat pot no ser tot l'exhaustiva que hauria de ser, fet pel que al final acabaria per apropar-se al model de cicle de vida clàssic.

## 1.6 Planificació del projecte

Partint del descrit al model de disseny a emprar (punt 1.5), la planificació del projecte ha d'incloure alguns punts més que són del tot necessaris per poder arribar des de la idea inicial fins a la presentació de les diferent fites com a resultat final del TFC. Així doncs, les fases a tenir en compte en la planificació són:

- Definició del projecte i planificació.
- Definició i anàlisi de requisits.
- Disseny.
- Implementació.
- Integració i proves.
- Documentació final.

A continuació es detallen les tasques a desenvolupar en cada fase. Els dies posats com a dates d'inici i fi són inclosos a cada un dels temporitzats descrits.

### 1.6.1 Definició del projecte i planificació

En aquesta fase es determina el projecte a desenvolupar, així com les tasques prèvies per començar a treballar, finalitzant amb l'elaboració de la planificació.

Tasca	Descripció	Inici/fi
Seleccionar el projecte	Elecció d'una de les idees de projecte per desenvolupar-la	21-09-2011 22-09-2011
Definir el projecte	Objectius i abast	23-09-2011 24-09-2011
Elaborar la planificació	Establir els temps de cada fase i tasques associades al projecte	25-09-2011 27-09-2011
Elaborar documentació	Redactar i maquetar la documentació que forma el pla de treball (PAC1)	28-09-2011 30-09-2011
Preparar l'entorn de desenvolupament	Instal·lar i verificar, si s'escau, les eines de desenvolupament (X-Code, Interface Builder, iOS simulator, etc)	01-10-2011 02-10-2011
<b>LLIURAR PAC1</b>	<b>ENVIAMENT DEL DOCUMENT FINAL CORRESPONENT AL PLA DE TREBALL</b>	<b>02-10-2011</b> <b>02-10-2011</b>

### 1.6.2 Definició i anàlisi de requeriments

En aquesta fase s'ha de determinar quins són els requisits del sistema i analitzar-los, per així poder identificar i detallar les necessitats. A més, s'han de realitzar tasques de camp, com poden ser la cerca

d'informació sobre els llenguatges a emprar, assimilació de conceptes dels mateixos i familiarització tant amb l'entorn com els llenguatges.

Tasca	Descripció	Inici/fi
Definir l'usuari/s implicats en el projecte	Estudiar a qui va dirigida l'aplicació, per poder donar resposta a les seves necessitats	03-10-2011 04-10-2011
Realitzar estudi de benchmarking	Indagar l'existència d'aplicacions similars al mercat, per determinar en que marcarà la diferència l'eina a desenvolupar	05-10-2011 07-10-2011
Establir les funcionalitats i tasques de l'aplicació	En funció dels perfils d'usuari i de les aplicacions existents, per donar solucions concretes amancances reals	08-10-2011 09-10-2011
Estudi d'usabilitat	Determinar els paràmetres necessaris per garantir l'experiència positiva a l'usuari final	10-10-2011 11-10-2011
Tasques de camp (treball en paral·lel)	Aprentatge i assimilació de conceptes bàsics dels llenguatges de programació a emprar. Familiarització amb l'entorn de treball.	03-10-2011 11-10-2011

### 1.6.3 Disseny

Fase en la que hem de dissenyar tots els elements que han de formar l'aplicació per tal de poder donar resposta a cada una de les funcionalitats i requisits especificats amb anterioritat. A més, s'ha de continuar amb l'estudi de camp iniciat a l'anterior fase, aquest cop aprofundint en els coneixements adquirits.

Tasca	Descripció	Inici/fi
Dissenyar l'arquitectura de la informació	Determinar el arbre de navegació i com queda la informació organitzada (en les classes necessàries)	12-10-2011 14-10-2011
Disseny de la persistència	Establir el model de dades a utilitzar per l'aplicació	15-10-2011 17-10-2011
Definició de la interacció	Establir els diferents casos d'us en funció de les funcionalitat i tasques definides	18-10-2011 20-10-2011
Disseny de prototip a baix nivell	Dissenyar els wireframes de l'aplicació, i les pantalles de navegació (estadi primari d'avaluació)	21-10-2011 25-10-2011
Disseny de la interfície	Prototipat definitiu de la interfície de l'aplicació, inclòs el sistema de navegació (emprar Interface Builder)	26-10-2011 31-10-2011
Elaborar documentació	Redactar i maquetar la documentació que forma el pla	01-10-2011

	de treball (PAC2)	02-11-2011
Tasques de camp (treball en paral·lel)	Aprofundiment en l'estudi dels llenguatges de programació a emprar.	12-10-2011 31-10-2011
<b>LLIURAR PAC2</b>	<b>ENVIAMENT DEL DOCUMENT FINAL CORRESPONENT A LA PAC2 (DEFINICIÓ + DISSENY)</b>	<b>02-11-2011</b> <b>02-11-2011</b>

### 1.6.4 Implementació

Fase del desenvolupament on tots els dissenys teòrics plantejats anteriorment s'han de materialitzar en forma d'aplicació. Gràcies a les tasques de camp es pot fer el desenvolupament a nivell de codi sense problemes. La part de modelatge 2D/3D, tot i poder ser considerada part de control, es destaca com a tasca addicional al tenir una gran envergadura i pes.

Tasca	Descripció	Inici/fi
Implementació de la base de dades (model)	Crear la BBDD, tant a nivell d'estructura com de dades. Preparació de l'aplicació per l'accés	03-11-2011 06-11-2011
Implementació de l'aplicació (control)	Crear la part de l'aplicació encarregada del control global del sistema (core)	07-11-2011 14-11-2011
Implementació de l'aplicació (vista)	Crear la interacció entre l'accés al model de dades i les interfícies creades. Millora de la navegació	15-11-2011 21-11-2011
Implementació del sistema de modelatge	Crear el mòdul de representació 2D/3D amb les llibreries Quartz 2D i/o Open GL ES	22-11-2011 30-11-2011

### 1.6.5 Integració i proves

En aquesta fase es verifica el correcte funcionament de l'aplicació implementada, segons les especificacions marcades en fases anteriors. També inclou la possibilitat de fer correccions/millors en funció dels resultats obtinguts.

Tasca	Descripció	Inici/fi
Integració i proves	Verificar el correcte funcionament i comportament de l'aplicació dissenyada	01-12-2011 05-12-2011
Revisió de l'aplicació (avaluació)	Corregir i/o afegir millores detectades a les proves. Verificació posterior	06-12-2011 09-12-2011
Elaborar documentació	Redactar i maquetar la documentació que forma el pla	10-12-2011

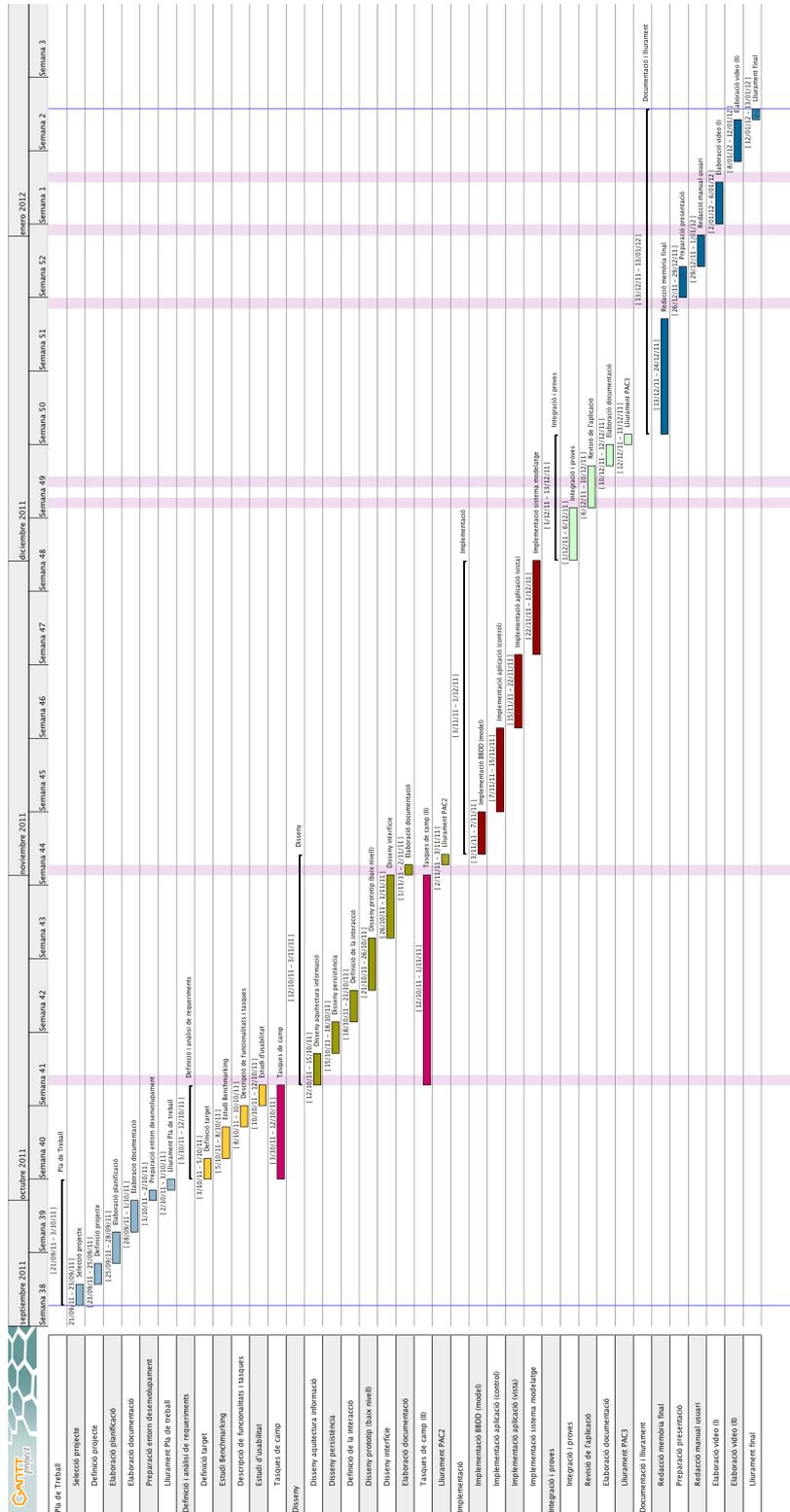
	de treball (PAC2)	12-12-2011
<b>LLIURAR PAC3</b>	<b>ENVIAMENT DEL DOCUMENT FINAL CORRESPONENT A LA PAC3 (IMPLEMENTACIÓ)</b>	<b>12-12-2011</b> <b>12-12-2011</b>

### 1.6.6 Documentació final

Un cop feta tota la feina (disseny, implementació i verificació) s'ha de fer la documentació del projecte. Això inclou un petit manual d'usuari, un informe de desenvolupament del projecte, i la presentació dels resultats obtinguts. A més, s'ha d'incloure la presentació del mateix en format vídeo (gravació, edició, maquetació, etc).

Tasca	Descripció	Inici/fi
Redacció de la memòria final	Recollir, organitzar i presentar tota la informació generada al llarg del projecte, aportant resultats i solucions a possibles millores o deficiències	13-12-2011 23-12-2011
Preparació de la presentació	Maquetació de la presentació, com a passi de diapositives	26-12-2011 28-12-2011
Redacció del manual d'usuari	Elaboració del document d'ús de l'aplicació per l'usuari final	29-12-2011 31-12-2011
Elaboració del vídeo (presentació virtual) I	Gravació del vídeo de presentació del projecte	02-12-2011 05-12-2011
Elaboració del vídeo (presentació virtual) II	Edició i maquetació	08-12-2011 11-12-2011
<b>LLIURAR INFORME FINAL</b>	<b>ENVIAMENT DEL DOCUMENT FINAL (INFORME + PRESENTACIONS + ANNEXOS)</b>	<b>12-01-2012</b> <b>12-01-2012</b>

Per poder veure tota la planificació d'una manera més esquemàtica i visual s'adjunta el corresponent diagrama de Gantt.



Imatge 4: Planificació del projecte

### 1.6.7 Possibles problemes i desviacions del pla de treball

Sota circumstàncies normals, es preveu poder complir els terminis en la pràctica totalitat. Només en casos excepcionals (per motius de feina, indisposició, o per retards inherents al desenvolupament del projecte) és possible haver de modificar algunes de les tasques.

Una de les parts més crítiques pot ser la feina de camp. Tot i que es fa de manera paral·lela a la part de disseny (més teòrica), es preveu que el ritme d'aprenentatge fins arribar al moment d'implementar ha de ser bo i constant, i suficient per assolir els coneixements que es necessitaran. No obstant, qualsevol dels contratemps esmentats podrien afectar a la part d'implementació. De tota manera, es valora que no ha de tenir un impacte excessivament elevat.

Altra part, tal i com s'ha apuntat, és la de la implementació. És una fase en que cometre errors pot derivar en un endarreriment de les tasques posteriors. Per sort, hi ha cert marge de maniobra, ja que tant la part d'integració i proves com la de documentació final es desenvolupen en una moment de l'any en que hi han moltes jornades del calendari laboral festives, i per tant, el temps de dedicació a les mateixes es pot incrementar. A més, en època nadalenca, hi han 5 dies que s'han determinat com lliures, que es poden fer servir en cas de necessitat.

## 2 Anàlisi de requeriments

### 2.1 Perfils d'usuari

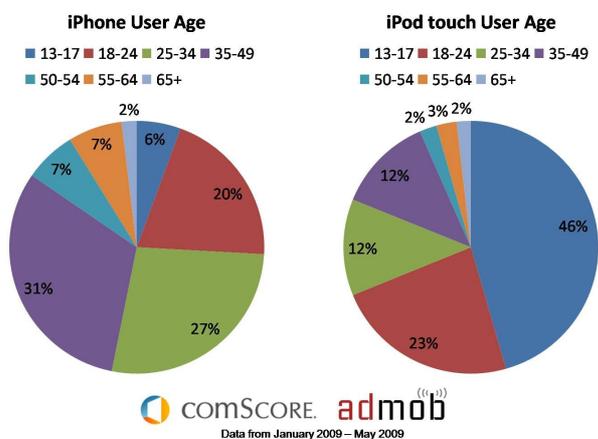
Un cop proposada la idea sobre l'aplicació a dissenyar, s'ha de filtrar el públic objectiu al qual va adreçada. Tot i que hom voldria implementar una eina que fos universal, aquest plantejament és erroni, ja que quant més heterogeni sigui el grup de persones al qual va adreçada, més complicat esdevé poder garantir una experiència satisfactòria per a totes elles. Així doncs, el millor es centrar-se en un grup més petit, més homogeni, al voltant d'unes característiques comuns i ben definides a través del que es coneixen com perfils d'usuari.

#### 2.1.1 Primeres aproximacions

Els usuaris potencials són gent relacionada amb el món de la química. A continuació es fa una classificació per tipus i edat.

1. Estudiants:
  - Estudiants de primària (6 a 12 anys)
  - Estudiants de secundària (12 a 16)
  - Estudiants de cicles formatius (16 cap endavant)
  - Estudiants universitaris (18 cap endavant)
2. Professionals: (18 cap endavant)
3. No professionals, aficionats a la química: (16 cap endavant)

#### iPhone vs. iPod touch: Age Breakdown



Imatge 5: iPhone - iPod Touch (utilització per edats)

Tenint en compte que es vol desenvolupar una aplicació que sigui d'utilitat en l'àmbit educatiu, sembla prou lògic pensar en els usuaris tipus estudiants. Això no exclou les altres dues categories, tal i com es justifica més endavant.

Un aspecte important a l'hora de poder determinar un grup d'usuaris més homogeni és tenir en compte les franges d'edat dels mateixos. Segons diferents estadístiques d'utilització (veure gràfic adjunt, pertanyent a un estudi del 2009 fet per *admob* & *comScore*) es pot veure que pel que fa a l'iPhone el 26% dels usuaris tenen edats compreses entre els 13 i els 25 anys, mentre que per aquesta franja el

percentatge d'usuaris de iPod Touch puja fins als 69%. Així doncs, sembla lògic acotar més la franja d'usuaris en estudiants de secundària, cicles formatius i universitaris. Per raons òbvies (dependència econòmica, nivell educatiu) queden descartats els estudiants de primària.

Dintre d'aquests tres perfils és ben clar que hi haurà una diferència primordial: el nivell de precisió en la informació mostrada en cada un. Així, un usuari de secundària necessita informació més bàsica respecte als elements i la formulació, mentre que estudiants de nivells més avançats han de fer servir propietats molt més específiques per desenvolupar diferents tipus d'estudis associats al seu nivell.

De manera anàloga als estudiants de graus superiors, els professionals poden tenir un perfil similar ja que necessiten més informació i més concreta per poder desenvolupar les seves tasques. En el cas dels aficionats a la química, potser s'adeqüen més al perfil bàsic (quan a quantitat d'informació) associat als estudiants de secundària, ja que potser no tenen la necessitat d'aprofundir tant en el tema, tot i si que els interessaria tenir un coneixement general.

### 2.1.2 Indagació i requisits d'usuari

La primera aproximació realitzada ha permès definir uns grups d'usuaris concret, ajustat a dos grups d'usuaris: els normals i els avançats. Però encara s'ha de fer un estudi més acurat, amb l'objectiu d'obtenir dades tant qualitatives com quantitatives que permetin establir els requisits d'usuari.

L'aplicació que dona sentit a aquest projecte no respon a una idea innovadora o que no existeixi. De fet, tant per iPhone com per iPod i iPad n'hi han aplicacions que realitzen tasques similars a les que s'exposen. El que s'està tractant és desenvolupar un programari que ofereixi millores respecte a les alternatives existents, i per tant, el procés més natural d'indagació ha de basar-se en:

- **Entrevistes.** Sobre mostres petites d'usuaris, i de caràcter obert, són de gran utilitat per obtenir informació referent a preferències, necessitats i experiència amb aplicacions similars.
- **Enquestes.** Desenvolupar un bon qüestionari (clar i concís) i aplicar-lo sobre una mostra dels usuaris potencials pot ajudar a obtenir informació quantitativa sobre el que aquests necessiten o demanden, concretant les dades obtingudes a partir de les entrevistes.
- **Benchmarking.** L'anàlisi comparativa consisteix en analitzar productes similars al que s'està desenvolupant. Amb això es persegueix conèixer les expectatives dels potencials usuaris (ja que poden donar idea de les bondats i defectes de les aplicacions ja existents), a més de poder veure com funciona el mercat relacionat i ajudar a determinar quines són les funcionalitats més comunes i com s'han dut a terme (sempre aprofitant l'experiència i opinió de l'usuari).

De les esmentades, potser la més important i que més informació (sobretot de manera empírica) pot aportar és el *benchmarking*, especialment a l'hora de filtrar quines funcionalitats han de ser primordials pel projecte que ens ocupa en funció de les experiències dels usuaris. Remarcar però que l'anàlisi comparativa no només es fa servir per especificar usuaris, sinó també per poder definir les accions (tasques) que s'esperen de l'aplicació a dissenyar. Més endavant es fa un estudi respecte a les aplicacions més destacades, extraient la

informació necessària.

Un cop conegut el *target* potencial per *iCompound* s'han de definir un (o diversos) personatges o models, i més endavant, situar-los en el context pertinent.

### 2.1.3 Determinació dels personatges.

A continuació es defineixen els personatges que modelen els usuaris dintre de cada perfil/grup.

#### Emma (Estudiant de secundària)

Emma Cruscat és una noia de 14 anys que actualment estudia en el IPS Esteve Terradas i Illa de Cornellà. Dintre de les matèries obligatòries pertanyents al camp de les ciències està la química elemental, gràcies a la qual està aprenent el paper crucial dels elements i les seves interaccions. Tot i que el nivell d'exigència no és especialment elevat, si que ha de conèixer la taula periòdica i com es distribueixen en ella els diferents elements, així com les propietats bàsiques de cada un d'ells (nom, símbol, nombre i pes atòmic, configuració electrònica). A més, ha de començar a adquirir nocions bàsiques de formulació de molècules/compostos.

Com a usuària de dispositius mòbils utilitza el seu iPod Touch generalment per escoltar música i per passar una estona d'oci amb alguns jocs (gratuïts). Al igual que la majoria de gent de la seva edat, l'adaptació a aquests tipus de dispositius ha estat innata, al pertànyer a una generació en que la tecnologia és una cosa totalment quotidiana. Així doncs, no te gaires problemes en la seva utilització ni en entendre de manera intuïtiva el funcionament de noves aplicacions que pot descarregar.

Per l'Emma, aplicacions com l'*iCompound* podrien resultar interessants a l'hora de tenir una petita guia de consulta a l'hora de fer els exercicis del dia a dia, sense dependre de grans quantitats de material en paper.

#### Josep (Estudiant de cicle formatiu)

Josep López estudia a l'Escola del Treball de Barcelona un CFGS en Anàlisi i Control. Als seus 29 anys, i després de ser acomiadat de la seva feina va decidir donar un pas endavant en la seva formació, continuant els estudis de Formació Professional en Química que va acabar temps enrere.

Gràcies a la seva formació inicial, i especialment a l'experiència laboral adquirida en el camp de la química, les bases de la matèria ja les té assolides. Tot i això, considera que una eina de suport sempre resulta necessària. En els nous estudis ha d'assimilar conceptes més avançats, i per tant, el fet de disposar de manera immediata i eficient a informació extensa resulta del tot interessant.

Com a usuari de noves tecnologies, i persona que vol estar sempre localitzat i connectat amb el món, posseeix un iPhone que no només li permet gaudir de les comunicacions mòbils, sinó que a més utilitza de manera assidua per consultar internet, a més de fer servir petites aplicacions que faciliten les seves tasques diàries (agenda, correu, calculadora, etc). De fet, sempre ha considerat que les seves possibilitats d'aquest dispositiu encara es podrien explotar més.

Cal destacar que el Josep pateix certs problemes de visió (miopia i daltonisme), fet que li dona certs problemes en l'ús del seu dispositiu mòbil depenent de les aplicacions (del seu disseny, més ben dit). Tot i això, la seva destresa i adaptació en l'ús d'aquest dispositiu es bona.

### David (Estudiant universitari)

David Güell, de 21 anys, està actualment estudiant Enginyeria Bioquímica a la UAB. Tot i que la carrera és molt dura ja que se suposen una bona base per poder seguir les matèries, el David no te gaires problemes en poder assimilar els nous conceptes ja que els seus coneixements són prou bons.

El fet d'estar fent el que li agrada fa que el David dediqui molt temps a l'estudi, i sempre ha trobat en falta una eina que sigui un compendi de les petites coses bàsiques (i de vegades, més avançades, com formulació de compostos orgànics) que ja coneix, alguna mena de guia de consulta concisa i de dimensions reduïdes. D'aquesta manera, podria repassar o verificar conceptes que, degut a la gran quantitat d'informació que pot arribar a haver-hi en el món de la química (elements, configuracions, compostos, etc) d'una manera senzilla i eficaç.

També s'ha de destacar la seva devoció per la tecnologia, fet pel qual sempre està al dia de les novetats generades dintre d'aquest mon. Això ha fet que al llarg dels darrers anys hagi adquirit productes com l'iPod (des de la tercera generació fins el Touch) o com el iPhone, als quals els hi treu força rendiment. Els considera eines indispensables en el seu dia a dia, i inclús pensa que es podrien aprofitar més per fer aplicacions en l'àmbit educatiu (la irrupció de les *tablets* al dia a dia, i més concretament, de l'iPad podria ser un gran avanç en l'aula interactiva).

### Pol (Professional)

Pol Requena treballa des de fa més de 5 anys en una empresa lider en el sector de l'anàlisi alimentària, i actualment és responsable de la gestió i control dels processos de producció d'aliments.. Amb 35 anys, va cursar estudis universitaris en Enginyeria Alimentària, a l'Escola Superior d'agricultura de Barcelona (UPC), essent un dels primers de la seva promoció.

Tot i tenir amplis coneixements en química, especialment aplicada al sector alimentari, com a bon professional sempre vol tenir eines que l'ajudin a desenvolupar les seves tasques de manera més eficient i precisa, i per tant, fa ús extensiu de totes les font d'informació que té a l'abast per poder millorar en la seva especialitat.

Pel seu càrrec de responsabilitat a dintre de l'empresa necessita estar disponible i connectat amb el seu equip i col·legues d'altres empreses. Per poder estar al cas de tot el que succeeix en tot moment tant a l'empresa (especialment incidències), estar al cas de les novetats en la seva àrea i/o aprofitar possibles col·laboracions externes en projectes comuns amb d'altres entitats, considera el seu iPhone com eina indispensable. Gràcies a aquest dispositiu pot tant per organitzar la seva feina com mantenir el contacte amb el seu mon. És per això que, amb eines de consulta integrades que l'ajudin en el seu dia a dia, està completament segur que el seu telèfon podria ser una font d'informació prou interessant i útil.

## Conchita (Aficionada a la química)

Des de que es va jubilar (amb 62 anys), la Conchita Carbonell té molt temps lliure per poder dedicar-se a algunes aficions que, per la seva vida dedicada a la seva vocació de joventut (professora de llengua a ensenyament primari) i les circumstàncies, no ha pogut dur a terme.

Entre elles està la química, que sempre l'havia apassionat i a la que ara ha decidit centrar els seus esforços. Degut a la seva edat, pateix certa pèrdua de visió i una artrosi en fase inicial. Això no l'impedeix fer una vida completament normal. És més, com a persona dedicada a l'ensenyament, sempre està oberta a aprendre coses noves, fet que l'ha fet estar molt al dia en el que es refereix a les noves tecnologies.

De fet, les possibilitats que ofereixen certs dispositius com l'iPad o l'iPhone, unit al fet que la pensió que li ha quedat és prou bona, l'han animat a adquirir-los. La seva expectació per poder treballar de forma centralitzada amb uns dispositius de dimensions reduïdes, on pot tenir molta informació i gestionar-la d'una manera senzilla i intuïtiva han pesat sobre les deficiències físiques abans esmentades. Només espera poder trobar aplicacions que l'ajudin a fer realitat el seu somni d'aprendre una disciplina nova, com és la química, encara que sigui dins el paper de simple aficionada.

### 2.1.4 Punts en comú entre els personatges.

Examinant els possibles personatges que poden representar als perfils inicialment establerts, i tenint en compte que no tots els individus (usuaris potencials) han de ser exactament com els especificats (els personatges són models d'estudi, no una realitat infal·lible), es poden extreure clarament un seguit de demandes, expectatives i necessitats comunes:

- Garantir la usabilitat, especialment pel que es refereix tant a visualització com als controls tàctils (senzills i efectius).
- Assegurar un accés a la informació per nivells. Això és, tenir les dades més rellevants i bàsiques accessibles de manera directa, deixant les més específiques en un segon pla.
- Fer que l'experiència de l'usuari a l'hora de fer servir una eina nova i útil sigui satisfactòria, el que implica que el seu ús sigui totalment intuïtiu a partir dels pocs coneixements teòrics associats a l'àrea de l'aplicació.
- Garantir la fidelitat de les dades i la seva precisió.
- Oferir una aplicació atractiva des del punt de vista visual, però que no amagui amb efectes espectaculars la verdadera potència de l'eina i la seva funcionalitat primordial.

Naturalment, això és genèric però no per això menys important. De fet, de poc serveix fer un desenvolupament basat en la potència si després l'usuari no té la sensació que ell controla el que fa, ni tampoc és gens lògic centrar els esforços en fer una eina que destaquï en l'aspecte visual en detriment de la usabilitat o, fins i tot, de la fiabilitat. S'ha de pensar en l'usuari, i una de les premisses bàsiques a seguir és

garantir-li una experiència en la utilització satisfactòria.

Així doncs, per poder captar l'atenció de l'usuari, el millor es seguir una sèrie de pautes (recomanades en les *Human Interface Guidelines*) que resulten totalment òbvies, però no per això s'ha de passar per alt:

- Oferir un ús senzill i obvi. L'usuari no acostuma a fixar-se en aplicacions de difícil gestió.
- Minimitzar la quantitat d'accions per accedir a la informació. A mesura que es complica la cerca del que l'usuari necessita, l'experiència d'usuari es fa més negativa.
- Utilitzar els controls estàndards oferts pel sistema (l'usuari ja n'està familiaritzat) sempre que sigui possible. Aprendre coses noves requereix un esforç, que l'usuari només estarà disposat a fer si és totalment necessari.
- Potenciar les funcions de l'aplicació, oferint un aspecte atractiu. S'ha d'arribar a un equilibri entre la qualitat en la interfície i la potencia de l'aplicació. L'usuari es sent atret per eines visualment ben desenvolupades, però les continua fent servir si les expectatives respecte a la seva utilitat són satisfactòries.

No cal dir que es pot parlar d'altres punts interessants, però els esmentats mostren que les necessitats de cada personatge s'ajusten perfectament als descrits. Així doncs, abans de passar a definir en profunditat les tasques que ha d'implementar el projecte que es tracta, s'ha de fer un estudi de les aplicacions existents i veure si les premisses exposades es compleixen.

## 2.2 Aplicacions relacionades amb la química (elements i compostos). Benchmarking.

Existeixen diverses aplicacions, tant per iPhone-iPod Touch com per iPad que proporcionen als usuaris les funcionalitats (generalment, de manera parcial) que *iCompound* pretén oferir. A continuació es mostra un estudi fet sobre algunes d'elles (de caràcter gratuït), oferint un enfoc tant objectiu (opcions oferides) com subjectiu (basat en l'experiència empírica d'un grup d'usuaris amb coneixements mínims de química) que permetran veure si les especificacions i expectatives es corresponen amb la realitat. Les proves s'han realitzat sobre els dispositius iPod Touch i iPhone, al ser el hardware del qual es disposava.

### 2.2.1 iElements



Desenvolupador: Max Soderstorm (SusaSoftX)

Versió: 1.2

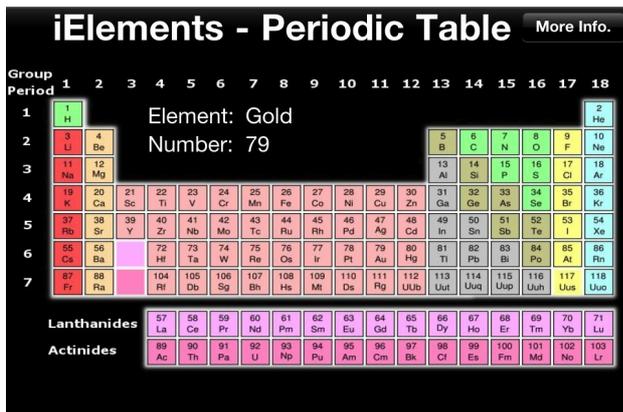
Plataforma: iOS 3.0 i posteriors

Compatibilitat: iPhone, iPod, iPad

<http://itunes.apple.com/es/app/ielements-periodic-table-the/id413632149?mt=8>

iElements és una taula periòdica simple, que ofereix accés a les propietats bàsiques i avançades de cada un dels elements que la formen (informació textual i gràfica, si n'hi ha).

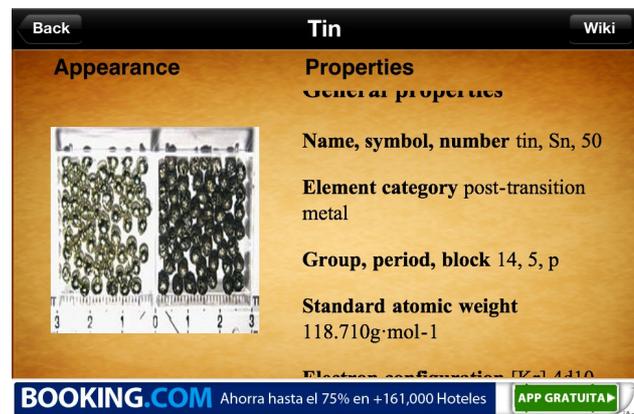
## Funcionament



Imatge 6: Pantalla principal de iElements

Un cop s'accedeix a la pantalla de propietats de l'element, es mostra una imatge (si aquesta existeix) que el mostra en el seu estat natural. Al costat, i a través d'una llista gestionada per un *scroll* vertical, l'usuari pot veure un llistat complet de les propietats de l'element, ordenades de més bàsiques a més complexes. A la barra superior de navegació es troben dues dreceres; la primera, a l'esquerra ('Back'), permet tornar a la primera pantalla per fer una nova selecció. La segona, sota l'epígraf 'Wiki' permet accedir a la informació de l'element emmagatzemada a la *Wikipedia* (pel qual es fa necessària una connexió a Internet).

L'aplicació comença mostrant la taula periòdica de la imatge 6, en la qual es pot seleccionar amb un clic la cel·la corresponent a l'element que es vol inspeccionar. Un cop feta la selecció (la casella queda marcada amb una vora il·luminada), no s'accedeix a la informació fins que no es prem el botó 'More Info', situat a l'extrem superior dret (barra de navegació). La taula, a més, es pot ampliar amb les accions tàctils estàndard emprades per fer zoom i poder fer una selecció més acurada.



Imatge 7: Pantalla de propietats de iElements

## Anàlisi

Es troben certes mancances i aspectes susceptibles de millora:

- **Interfície poc treballada.** L'aspecte en general resulta poc atractiu, a més de la molesta publicitat que apareix en forma de barra inferior.
- **Poc intuïtiu i directe.** Un usuari espera senzillesa i haver de fer poques accions per accedir al que vol. Es possible detectar certes mancances al respecte:

- Quan es selecciona un element de la taula periòdica (pantalla principal) s'espera accedir a la informació del mateix. En canvi, s'ha de prémer un botó per aconseguir-ho, després d'haver fet la selecció.
- En la pantalla de mostra d'informació, el fet de mostrar-la tota fent servir un *scroll* fa tediós l'accés a les dades del final de la llista. A més, aquest no es visible si l'usuari no es posiciona sobre les dades, fet que pot confondre (es pot pensar que tota la informació és la que es veu a primer cop d'ull).
- **Usabilitat i accessibilitat baixa.** La selecció dels elements es fa certament complicada, al ser les caselles que formen els elements de dimensions massa reduïdes. Tot i que es cert que és possible fer un zoom sobre la zona desitjada, millorant així la tasca de fer seleccions, un cop feta obliga a desfer el zoom, o bé moure la pantalla per poder accedir a la informació amb el botó '*More Info*'.

Tot i que en menys proporció, també es troben coses interessants que, potser treballades d'una altra manera poden ser d'utilitat, com pot ser l'accés a recursos externs més rics en informació (millors i més fiable que no pas la *Wikipedia*).

Com a conclusió, es pot dir que iElements no resulta una aplicació especialment recomanable, pel que fa a experiència d'usuari, fet que no queda justificat amb la definició del producte: *una taula simple*. Es pot fer un producte bàsic i senzill, però per focalitzar més en la informació donada cap a un públic específic, i no pas sacrificant aspectes com la usabilitat. Com a mínim, pel que fa a dispositius de pantalla petita (iPhone i iPod Touch), que és sobre els qual s'ha fet el test (de ben segur que l'experiència millora en la versió d'iPad de manera notable).

### 2.2.2 Merck PSE HD



Desenvolupador: Merck KGaA

Versió: 1.6.0

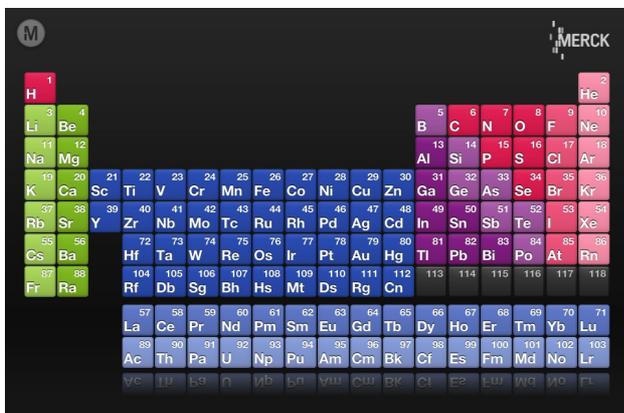
Plataforma: iOS 3.2 i posteriors

Compatibilitat: iPhone, iPod, iPad

<http://itunes.apple.com/mx/app/merck-pse-hd/id375734631?mt=8>

PSE HD és, sense cap mena de dubte, una de les taules periòdiques més completes de les que es troben d'entre les diferents aplicacions per dispositius Apple. A banda d'oferir informació completa i ben estructurada (diferents representacions de la taula, imatges, etc), presenta el producte d'una manera difícilment igualable. Una eina ben dissenyada i pensada per facilitar les consultes de tot aquell usuari familiaritzat (més o menys) amb la química. Tot això, amb una gran empresa al darrera ben coneguda i reconeguda pels professionals d'aquesta àrea científica.

## Funcionament



Imatge 8: Taula periòdica de Merck PSE HD

El punt d'entrada de l'aplicació és la taula periòdica mostrada a la *imatge 8*, corresponent a una classificació segons el caràcter químic (alcalins, alcalinoterris, metal·loides, etc). Prement el botó 'M' (part superior esquerra) es poden realitzar diferents seleccions, com les relacionades amb la classificació segons certs paràmetres de la taula (mostrant diferents colors distintius per cada element/grup), un cercador o fins i tot llistat ordenat d'elements i propietats, entre d'altres opcions.



Imatge 9: Selecció d'element Merck PSE HD

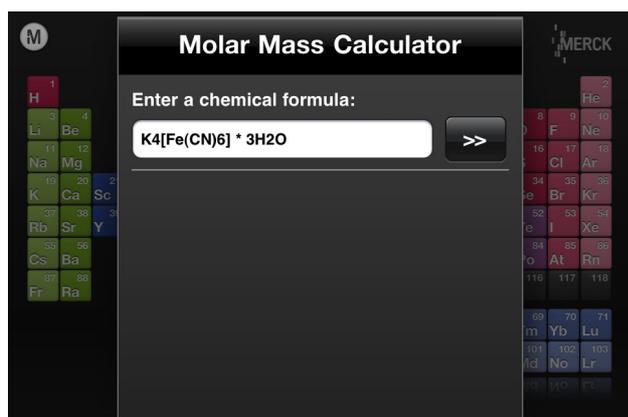
D'altra banda, es possible accedir a les propietats concretes de cada element. Per fer-ho només cal fer el gest de desplaçar sobre la taula, amb el qual es destaca l'element sobre el que es passa. Deixant de tenir contacte amb la pantalla la darrera selecció esdevé l'activa, obrint una nova finestra sobre la principal, en la que es poden veure les dades més elementals de l'element triat (*imatge 9*).

El menú de selecció de propietats és prou complet, permetent a l'usuari accedir a informació sobre l'element de tot tipus (bàsica, avançada, dades del descobriment, imatges, etc) que poden ser de gran utilitat per persones amb diferents graus de

coneixement sobre la matèria (veure *imatge 10*). La navegació es fa sobre la nova finestra de propietats que s'ha obert abans, escollint l'opció desitjada, o tornant al menú de selecció (prement el botó amb el nom de l'element situat a la barra superior de navegació de la sub-finestra). Per tornar a la taula inicial, només s'ha de seleccionar l'àrea de fora de l'espai de dades de l'element.



Imatge 10: Accés a les propietats dels elements Merck PSE HD



Imatge 11: Calculadora molar Merck PSE HD

Cal destacar, d'entre les funcionalitats addicionals que ofereix l'aplicació (accessibles, tal i com s'ha dit, prement el botó 'M', la calculadora de la massa molar.

Aquesta eina resulta de gran utilitat, ja que a partir d'una formulació introduïda per l'usuari s'obté el valor molar del compost donat d'una manera senzilla i clara.



Imatge 12: Tipus de taula Merck PSE HD

## Anàlisi

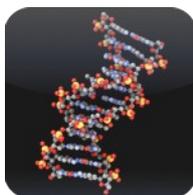
Es pot afirmar que Merck PSE HD és una aplicació molt treballada tant en l'aspecte visual com en fidelitat de les dades i usabilitat. No obstant, hi ha algunes possibles millores:

- **Accés a les dades d'element.** Tot i que el menú tipus llista permet accedir a la informació relacionada amb l'element fent servir a sub-vistes, en algun cas (com en informacions bàsiques) potser seria més còmode per l'usuari no haver de fer servir un *scroll*. Desglossant la informació en diferents nivells (i obviant dades no rellevants), l'experiència resultaria més senzilla i satisfactòria.
- **Accés al menú d'opcions de la taula.** Resulta poc intuïtiva la manera d'accedir a aquest apartat, ja que se suposa que l'usuari ha d'entendre que la 'M' del costat superior esquerra és un botó per anar a un altre lloc, no trobant les alternatives que amaga, al menys, d'inici.

Com a conclusions, i després de l'anàlisi, queda palès que l'aplicació que s'ha vist ofereix un ventall molt ampli d'opcions (la gran majoria útils) per un no menys ampli públic (sempre, del món de la química). No és pas estrany, tenint en compte que Merck és una empresa amb una gran tradició i reconeixement dintre d'aquesta àrea des de fa molts anys.

Tal i com s'ha dit, l'únic però es pot trobar en el fet que l'experiència d'usuari pot no ser tot el satisfactòria que podria, al utilitzar certs efectes i funcionaments no del tot estàndards (segons les *Human Interface Guidelines*), ja que obliguen a aprendre en certa manera nous comportaments, a més d'amagar, sota la seva sorprenent vistositat, opcions molt potents.

### 2.2.3 Molecules



Desenvolupador: Brad Larson (Sunset Lake Software)

Versió: 2.0.2

Plataforma: iOS 4.0 i posteriors

Compatibilitat: iPhone, iPod, iPad

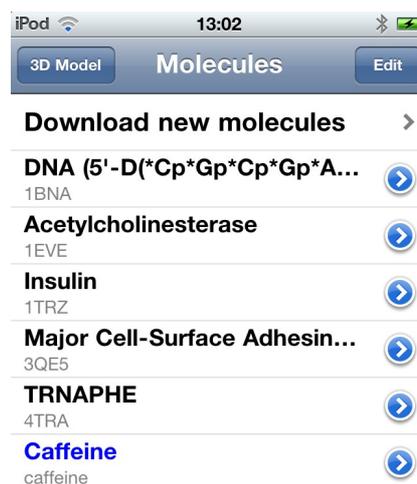
<http://itunes.apple.com/app/molecules/id284943090?mt=8>

Molecules és una aplicació que permet visualitzar models en tres dimensions d'estructures moleculars. Recolzada sobre dues potents bases de dades *on-line* especialitzades, de les quals es poden obtenir aquests compostos, resulta especialment interessant al permetre interaccionar amb els models 3D aprofitant les propietats tàctils dels dispositius Apple.

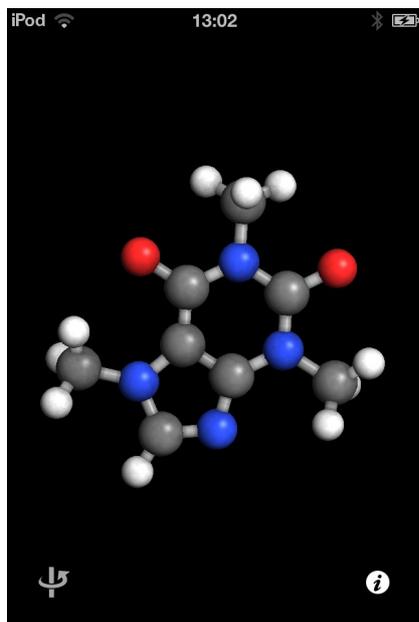
## Funcionament

L'aplicació s'inicia mostrant una llista en format taula que dona l'opció de seleccionar unes molècules predeterminades i realitzar certes operacions sobre cada una de elles.

La primera, prement la icona amb fletxa situada al costat dret de cada entrada, consisteix en veure les característiques de la molècula en qüestió (nom del fitxer que conté les dades, formació, estructura, etc) en una nova vista. La segona és el modelatge en 3D (amb tècniques de renderitzat) de la molècula seleccionada. Aquesta és, sense cap mena de dubte, la part central de l'aplicació.



Imatge 13: Molecule Manager



Imatge 14: Molecule viewer

La visualització tridimensional està pensada per facilitar a l'usuari la comprensió sobre com està estructurada i distribuïda la molècula, a partir de les interconnexions dels diferents àtoms elementals que la conformen. Per poder tenir una experiència interactiva, i poder escollir diferents punts de vista i angles, es tenen dos opcions. La primera és un sistema de rotació sobre el seu propi eix (concretament, el vertical) automàtic que es pot habilitar / inhabilitar prement la icona situada en la cantonada inferior esquerra. La segona es basa més en la interacció, i per tant l'usuari pot moure la molècula en el sentit que desitgi fent servir les propietats tàctils estàndards del dispositiu, a més d'oferir la possibilitat de fer un zoom en el moment que es vulgui, per veure més detalls de la representació.

Gràcies al botó situat a la cantonada inferior dreta de la pantalla pot tornar a la pantalla principal.

Finalment, i tornant a la pantalla inicial, existeix una tercera opció que consisteix en connectar-se a dos de les més grans i fiables bases de dades moleculars existents ([Pub Chem](#) i [Protein Data Bank](#)) per obtenir-ne de noves, prèvia cerca, i així poder examinar el seu model tridimensional. Naturalment, aquesta opció requereix una connexió activa a Internet. Això és una de les característiques de l'eina examinada: les

representacions s'obtenen gràcies a uns fitxers en un tipus de format estàndard que permet el modelatge, seguint una sèrie de premisses. Com a valor afegit, des del menú principal també és possible editar cada una de les entrades, per eliminar-les si s'escau.

## Anàlisi

L'eina de la que s'està parlant compleix la seva tasca sense cap mena de dificultat i amb màxima eficiència: modelar en tres dimensions molècules a partir de fitxers formatats per aquesta tasca, els quals s'obtenen des de bases de dades contrastades. Tot i això, i havent de reconèixer que les representacions tridimensionals estan ben aconseguides, hi han alguns aspectes que poden afectar la usabilitat:

- **Problemes amb el zoom en el mode representació.** Aquesta opció pot ser de gran utilitat per veure detalls d'altre manera difícils d'observar. El problema ve quan hom vol fer un zoom-out (és a dir, allunyar la vista), ja que no hi ha límit i la molècula arriba a desaparèixer de la pantalla, essent molt complicat tornar-la a recuperar.
- **Dificultat en la cerca de noves molècules.** La cerca de noves molècules pot resultar complicada al exigir ple coneixement del nom del que es vol obtenir. Per un usuari novel, potser l'experiència, en aquest sentit, pot resultar frustrant al no tenir prou coneixements (o tenir aquesta sensació), tot i que no pas per usuaris amb coneixements sobre compostos (pels quals pot resultar clar i directe)

Per finalitzar, dir que no es pot negar l'exactitud i precisió en l'execució de la tasca primordial de l'eina, a més de la senzillesa en la utilització. Potser es pot posar algun però en temes d'usabilitat, però són coses fàcilment corregibles en properes versions. Destacar també que, a pesar que les representacions són força aclaridores, no estaria de més tenir una llegenda dels colors emprats per cada àtom, per facilitar a l'usuari la seva comprensió (especialment sinó és un expert en la matèria).

### 2.2.4 Conclusions de l'estudi.

Després d'analitzar dues aplicacions relacionades amb la taula periòdica i una amb modelatge en tres dimensions de molècules, cal destacar certes mancances de cada una d'elles, però sobretot els aspectes positius que poden ajudar sense cap mena de dubte a saber donar a l'usuari el que vol. Així doncs, per poder desenvolupar amb èxit *iCompound*, i garantir una total satisfacció quant a usabilitat i utilitat, s'ha de tenir en compte:

- Donar la informació ben estructurada i organitzada segons la lògica intrínseca a la disciplina amb la que s'està tractant.
- Intentar respectar al màxim les recomanacions donades a les *Human Interface Guidelines*. Això és, entre d'altres, facilitar al màxim l'experiència d'usuari no obligant-lo a aprendre comportaments d'aplicació nous i massa complicats, aprofitar al màxim els recursos estàndards oferts pel sistema (a excepció que no fer-lo estigui justificat per millorar) i/o no ocultar sota espectaculars efectes la finalitat real de l'eina desenvolupada.

- Intentar centrar els esforços en les funcionalitats reals que l'usuari (els personatges creats) necessita, per tal de donar un resposta òptima al que espera trobar-se.

A partir de l'esmentat, dels anàlisis dels personatges / aplicacions escollides (gràcies a les seves experiències amb elles), s'està en disposició de començar a definir de forma detallada cada una de les funcionalitats que han de conformar *iCompound*, que permetin satisfer els dos grans grups d'usuaris als que va destinat: els de perfil bàsic i els de perfil avançat.

## 2.3 Definició de funcionalitats i tasques

A partir de les necessitats de cada perfil, i tenint en compte l'estudi de benchmarking realitzat es poden definir en la pràctica totalitat les funcions i tasques que ha de oferir *iCompound*. Tenint en compte el grau de profunditat requerit per cada tipus d'usuari, es procedeix a classificar-les sota dos grups: bàsiques i avançades.

### 2.3.1 Funcionalitats bàsiques

Són totes aquelles que qualsevol usuari ha de tenir a la seva disposició, independentment del seu grau de coneixement respecte a la química. Haurien de ser:

#### 2.3.1.1 Relacionades amb els elements i la taula periòdica

1. Seleccionar format de Taula Periòdica (clàssic)
2. Visualitzar elements disposats sobre la taula
3. Seleccionar tipus de visualització de la Taula segons:
  - caràcter químic (per defecte)
  - electronegativitat
  - radi atòmic
  - estat d'oxidació
4. Seleccionar format distribuït com a llista ordenada
5. Visualitzar llista amb elements
6. Ordenar la llista (en grups si s'escau) segons:
  - Alfabèticament
  - Número atòmic
  - Grup
  - Període
  - Caràcter químic (per defecte)
  - Electronegativitat
  - Pes atòmic

(Tasques comunes a ambdues visualitzacions)

7. Seleccionar element i accedir a les propietats
8. Accedir a les propietats bàsiques de l'element seleccionat
  - Nom
  - Nombre atòmic
  - Classificació (caràcter químic)
  - Grup
  - Període
  - Electronegativitat
  - Nivells d'oxidació
9. Accedir a les propietats relacionades amb l'àtom
  - Gràfic de l'àtom
  - Estructura electrònica
  - Pes atòmic
  - Radi atòmic

### **2.3.1.2 Relacionades amb el mòdul de molècules**

1. Visualitzar llistat de molècules comuns (predefinides)
2. Seleccionar una molècula
3. Seleccionar model (commutar entre 2D i 3D)
4. Veure model 2D (fórmula desenvolupada)
5. Veure el model 3D (*balls and sticks*)
6. Interaccionar amb el model 3D (controls tàctils)
  - Desplaçar
  - Rotar
  - Funcions de Zoom

### **2.3.2 Funcionalitats avançades**

Són totes aquelles que, com a norma general, van adreçades a usuaris amb un grau de coneixement respecte a la química més elevat. Tot i això, **no son funcionalitats exclusives**, i per tant, poden resultar d'interès per d'altres perfils menys instruïts (per exemple, per poder aprofundir en la matèria).

#### **2.3.2.1 Relacionades amb els elements i la taula periòdica**

10. Accedir a les propietats específiques de l'element seleccionat
  - Densitat
  - Volum atòmic
  - Temperatura de fusió

- Temperatura d'ebullició
  - Radi covalent
  - Potencial 1er d'ionització
11. Utilitzar calculadora molar
    - Inserir dades
    - Obtenir resultats

### 2.3.2.2 *Relacionades amb el mòdul de molècules*

7. Afegir molècula al llistat
8. Editar llistat de molècules \*
9. Esborrar molècula del llistat \*
10. Cercar i descarregar noves molècules de la base de dades del NIST

\*Disponible només per molècules inserides de manera manual.

## 2.4 Estudi d'usabilitat

Determinades les funcionalitats, tasques i requeriments de l'aplicació a desenvolupar, queda fixar quins paràmetres s'han de fer servir per tal de garantir màxima usabilitat i fer que l'experiència d'usuari sigui altament satisfactòria.

Una garantia d'èxit en aquest sentit és tenir en compte certs criteris (ja esmentats) establerts a les per Apple (*User Experience Guidelines*), i aplicar-los per poder oferir un alt grau d'usabilitat. De gran ajuda resulta en aquest punt les observacions fetes a l'estudi de benchmarking, ja que donen idea de què espera l'usuari de l'eina que ha de fer servir.

Així doncs, i seguint aquestes recomanacions, es tindrà en compte:

1. **Centrar esforços en la tasca principal.** *iCompound* és una aplicació en la qual l'usuari pot interactuar amb els elements disposats en dos formats (taula i llista), organitzables segons diferents paràmetres, per tal d'obtenir informació rellevant. A més, permet visualitzar compostos químics en 3D (predefinits) i en format de fórmula desenvolupada, podent calcular propietats molars dels compostos desitjats. Aquestes són les funcions principals, i per tant, no s'ha de fer res que no quedi marcat per elles.
2. **Destacar el que l'usuari pot necessitar.** En una eina com aquesta, el principal actiu és la informació referent tant a elements com a compostos. Per tant, s'ha de deixar a l'usuari ben clar que això és l'important de l'aplicació, i no pas els efectes o coses supèrflues; s'ha de fer que l'accés a la informació sigui el més important.

3. ***Distribuir la informació segons la seva rellevància, garantint un accés lògic a les dades.*** Segons els perfils definits, que es podrien reagrupar en dos elementals (usuari normal, usuari avançat), el més lògic és que la informació destinada al primer col·lectiu sigui accessible a primer cop d'ull i sense haver de fer cap esforç. Això és, posar-la primera en ordre de visualització, deixant la destinada al segon col·lectiu en un segon lloc, al ser més específica (però sense complicar l'accés).
4. ***Assegurar un ús senzill i obvi, fent ús dels controls estàndard sempre que sigui possible.*** L'usuari està acostumat a realitzar una sèrie d'accions i uns formats (de llistes, de barres, etc) prou estandarditzats. Així, no cal reinventar res que ja existeixi, si amb el que ofereix el sistema (i una mica de personalització) és possible fer que l'usuari pugui navegar per l'aplicació de manera senzilla. A més, una bona disposició de la informació (com es deia a l'apartat anterior) farà l'ús més intuïtiu i obvi.
5. ***Utilitzar terminologies que l'usuari pugui comprendre, amb informació clara i precisa.*** No cal fer servir llenguatge complicat en els missatges, títols o menús, ja que això resulta farragós per l'usuari. S'ha d'informar, però no omplir amb retòrica innecessària la pantalla. D'altra banda, utilitzar el llenguatge tècnic químic en el cas estudiat pot ser de gran ajuda, ja que promou que la persona es senti en el seu medi al interactuar amb l'aplicació.
6. ***Tenir cura de la visualització (colors, efectes espectaculars, dimensions, etc).*** L'aspecte de l'aplicació és important. Que tot estigui ben distribuït en la finestra, que les combinacions de colors no siguin desagradables a la vista de l'usuari, i fer servir efectes per donar sensació d'interactivitat són coses que milloren l'experiència de l'usuari. En el cas que ens ocupa, fer transicions entre pantalles d'informació o avisar del progrés de renderització d'una molècula (o de descàrrega) pot ajudar a tenir sensació que s'estan fent coses. Això si, no cal fer efectes massa espectaculars per no perdre de vista que la informació és el tema primordial a destacar.
7. ***Aprofitar els controls tàctils i de orientació.*** La possibilitat de poder fer accions amb un toc de dit, arrossegant-lo, etc dona sensació a l'usuari d'interactivitat i sobretot de control. Així doncs, s'ha de garantir que la interacció és basi en aquestes accions (ja assimilades, d'altra banda) per poder fer servir l'aplicació. Especialment crític pot resultar quan s'accedeix a la Taula periòdica (molts elements, de dimensions reduïdes), i per tant s'ha d'assegurar que les funcions tàctils siguin eficients i no duguin a confusions o funcionaments que puguin generar certa sensació de mal funcionament. També s'han de garantir, pel que fa a la visualització de molècules en 3D, que mai es perdi de vista el model mostrat.  
En el cas de l'eina *iCompound*, els controls d'orientació no són primordials ja que el posicionament de les vistes es fixa per defecte, per tal que l'accessibilitat i usabilitat quedin totalment garantides. Així doncs, en la mesura del possible, aquestes funcionalitats no s'han de fer servir al no ser indispensables.

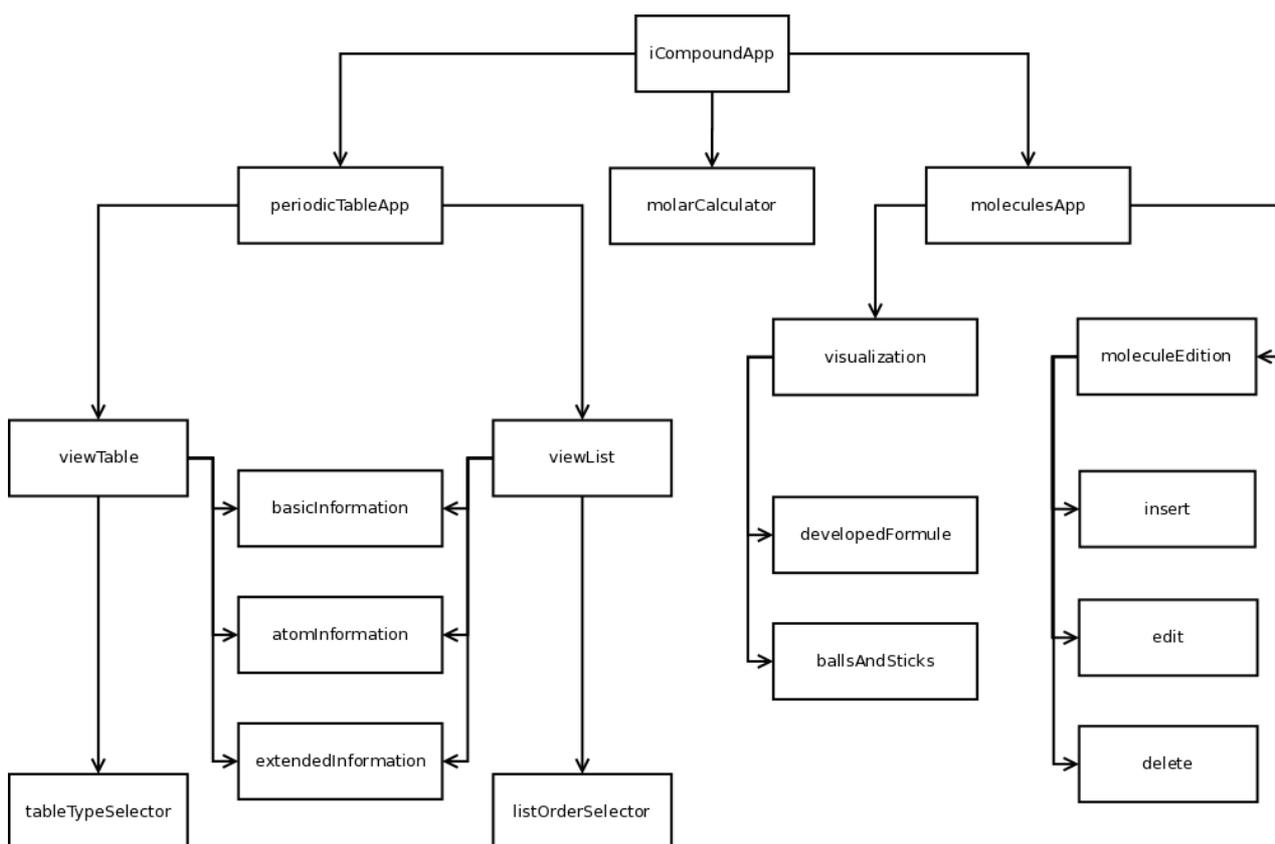
8. **Garantir rapidesa de resposta.** Un punt crític. Tot i que hi han operacions que poden reduir temporalment la velocitat de resposta (com pot ser el renderitzat d'una molècula complexa, la descàrrega de noves molècules, etc), l'usuari no ha de tenir la sensació que això passa (com s'ha dit, es pot, per exemple, mostrar una barra de progrés). En casos com el que es tracta, s'ha de prestar especial atenció sobretot en la part de modelació d'objectes en tres dimensions, i intentar tenir cura (analitzant el codi font, revisant processos i funcions, etc) per fer que l'aplicació estigui el més optimitzada possible.

### 3 Disseny

Un cop determinats aspectes previs com són conèixer els perfils d'usuari associats a l'aplicació i les funcionalitats / tasques que ha de oferir, el següent pas és definir l'estructura de funcionament, l'estructura de dades que ha de donar suport a la informació que gestiona, i finalment, fer el prototipat de la interfície d'usuari.

#### 3.1 Esquema de navegació

L'usuari ha de realitzar un seguit d'accions per poder anar d'una finestra a altre, les quals s'han de definir de manera que aquesta navegació sigui lògica i senzilla. Tenint en compte les funcionalitats i definicions fetes en l'apartat d'anàlisi, es pot definir un arbre de navegació òptim tal i com el que es mostra en la següent il·lustració.



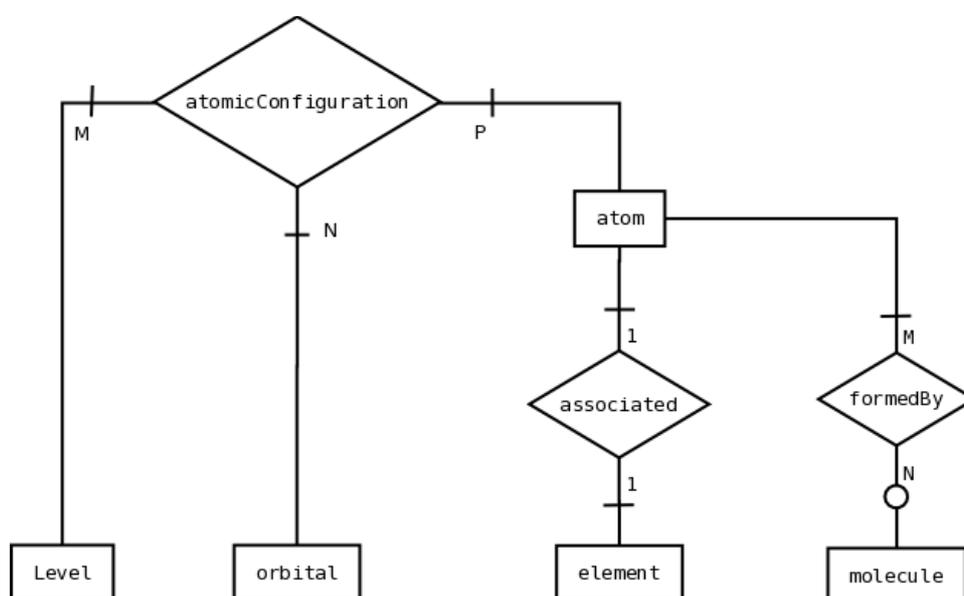
Imatge 15: Esquema de navegació per iCompound

Més endavant es concretaran certs aspectes que tenen a veure amb el model utilitzat a l'hora d'implementar el disseny (concretament el conegut com *Model View Controller, MVC*). De moment però, l'esquema de

navegació mostrat permet fer-se una idea de com ha de ser la interacció de l'usuari amb l'aplicació per tal d'obtenir els resultats que espera de manera satisfactòria.

### 3.2 Persistència

El primer que s'ha de fer per poder definir les classes involucrades en l'aplicació és establir el model de negoci. En el cas que ens ocupa, aquest resulta certament simple al haver-hi entitats completament definides i acotades. Això queda ben representat al diagrama E-R que es mostra a continuació.



Imatge 16: Persistència per iCompound

Sobre les entitats és precís fer alguns apunts per tal de no incórrer en equívocs que podrien donar com a resultat la no optimització de l'estructura de dades:

- La taula periòdica està formada per elements, concretament (en l'actualitat) per 118. Tot i que pot ser que en un futur s'afegeixi un de nou (o més), l'estructura de la mateixa és prou estable com per posar un límit concret. És una de les parts principals de l'aplicació, però la seva organització es basa en atributs recollits en les entitats representades, i per tant, no té sentit crear una entitat per ella (només hi ha una per tots els elements).
- Un àtom té una i només una configuració electrònica (*electronicConfiguration*) associada, i si no existeix àtom la seva existència deixa de tenir sentit. Es presenta com entitat associativa, ja que relaciona els nivells d'energia (*level*) i els seus corresponents orbitals (*orbital*) amb l'àtom.
- Un àtom té d'un a set nivells d'energia, i cada un d'ells pot contenir d'un a quatre orbitals. Cal dir

que dintre d'un nivell no es pot repetir un tipus d'orbital, i que no qualsevol tipus d'orbital pot existir dintre de cada nivell. A continuació es mostra una taula que resumeix, de manera simplificada, com han d'organitzar-se.

Nivell d'energia (n)	Orbitals
1	s
2	s,p
3	s,p,d
4	s,p,d,f
5	s,p,d,f
6	s,p,d
7	s,p

- Un element està format per un o més àtoms d'aquest element. Així, la dependència de l'un està lligada a l'altre de manera inequívoca. Tot i així, de cara a la informació de la taula, el que interessa és conèixer l'estructura de l'àtom associat a l'element i, per això, l'associació és 1:1.
- Una molècula (*molecule*) es forma, com a mínim, amb dos àtoms que poden correspondre a elements diferents (H<sub>2</sub>O, NaCl, etc) o iguals (O<sub>2</sub>, O<sub>3</sub>, H<sub>2</sub>, etc). Del segon cas, alguns són coneguts com molècules diatòmiques, i són a l'hora elements (al trobar-se així en la natura) A més, que no existeixi una molècula no implica que no existeixin àtoms.
- Un compost (*compound*) és un tipus de molècula que està formada per dos o més elements (o el que és el mateix, per àtoms diferents).
- Per cada molècula, l'usuari pot obtenir la seva representació gràfica, sempre i quan aquesta existeixi. A més, si existeix, es pot tenir més d'una diferent a l'hora (per poder observar diferents característiques), al ser complementàries. Com per fer la representació gràfica s'utilitzen paràmetres recollits en les entitats mostrades (més concretament, a la molècula) no cal crear una entitat pròpia.

Un cop fetes les precisions pertinents, s'ha de concretar les entitats amb els atributs corresponents.

#### ATOM

atomic-name, atomic-weight, atomic-radius, atomic-volume

#### LEVEL

level-number

#### ORBITAL

orbital-name, number-of-electrons

#### ELEMENT

element-name, chemical-character, group, period, aggregation-state,

oxidation-level, boiling-point, melting-point, electronegativity,  
covalent-radius, first-ionization-potencial

MOLECULE

inchiKey, moleculeName, formula, fileId

A partir del model de E-R que s'ha descrit es pot definir el model relacional de les dades (a partir de les corresponents transformacions) com:

ATOM (atomic-name, atomic-weight, atomic-radius, atomic-volume)

LEVEL (level-number)

ORBITAL (orbital-name, number-of-electrons)

ATOMICCONFIGURATION(atomic-name, level-number, orbital-name, formula)

on {atomic-name} referencia ATOM,  
{level-number} referencia LEVEL  
i {orbital-name} referencia ORBITAL

ELEMENT (element-name, chemical-character, group, period, aggregation-state,  
oxidation-level, boiling-point, melting-point, electronegativity,  
covalent-radius, first-ionization-potencial, atomic-name)  
on {atomic-name} referencia ATOM

MOLECULE (inchiKey, moleculeName, formula, fileId)

FORMEDBY(molecule-name, atomic-name)

on {atomic-name} referencia ATOM  
i {inchiKey} referencia MOLECULE

Amb això, les classes que s'han d'implementar queden fixades.

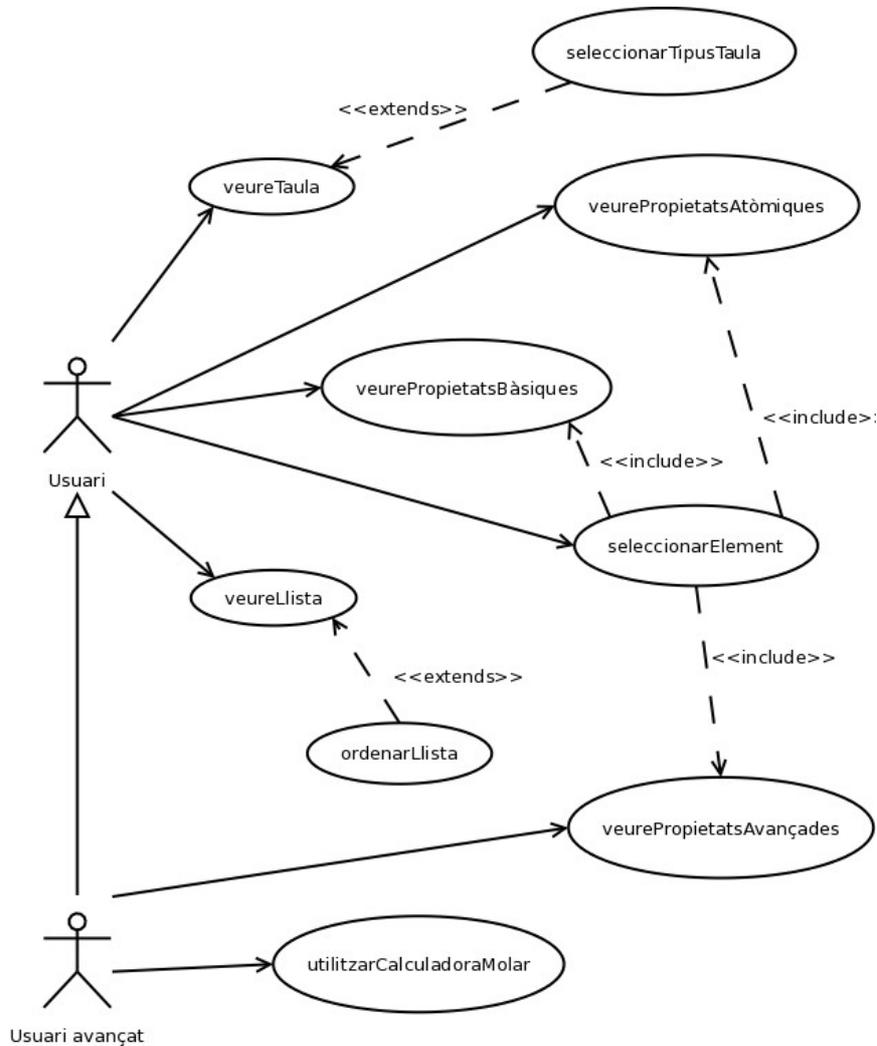
### 3.3 Interacció: casos d'us

A partir de les funcionalitats definides en anteriors apartats, i tenint en compte els perfils d'usuari definits s'ha d'analitzar la interacció entre usuari i aplicació.

Com a punt de partida el que es fa es separar en dos grans grups les tasques a realitzar: les relacionades amb els elements (la taula periòdica i calculadora molar) i els relacionats amb les molècules i la seva visualització. En conseqüència, i coneixent la resta de paràmetres es poden definir i posteriorment analitzar els casos d'us corresponents.

### 3.3.1 Elements: taula periòdica i calculadora molar

El diagrama de casos d'us d'aquest bloc es mostra a continuació.



Imatge 17: Casos d'us elements per Taula periòdica i Calculadora Molar

S'ha de fer un petit aclariment respecte els *includes* sobre *seleccionarElement*. La idea és que quan l'usuari selecciona l'element pugui visualitzar una de les propietats indicades (per defecte, les bàsiques). No es pot dir que totes en siguin d'obligatòries, però sí que ha de succeir com a mínim una d'elles, i per això, no poden ser opcionals en sentit estricte.

Definit l'esquema de casos d'us s'han d'especificar cada un d'ells de manera textual.

ETP1	veureTaula
<i>Resum</i>	Permet veure els elements disposats en format de taula clàssic
<i>Actors</i>	Usuari,Usuari avançat
<i>Casos d'us relacionats</i>	seleccionarTipusTaula
<i>Precondició</i>	S'ha accedit al mòdul d'elements
<i>Postcondició</i>	Es visualitzen les dades per pantalla
<i>Descripció</i>	1.- L'usuari selecciona veure les dades (elements) disposats sobre una taula, tal i com es coneix formalment 2.- Com a resultat, obté la visualització demanada
<i>Excepcions</i>	Cap
<i>Observacions</i>	Cap

ETP2	seleccionarTipusTaula
<i>Resum</i>	Permet seleccionar el tipus de visualització de la taula
<i>Actors</i>	Usuari,Usuari avançat
<i>Casos d'us relacionats</i>	veureTaula
<i>Precondició</i>	L'usuari ha seleccionat el format de taula clàssic
<i>Postcondició</i>	Es visualitza la taula segons el tipus seleccionat
<i>Descripció</i>	1.- L'usuari tria l'opció de canviar el tipus de taula 2.- El sistema obre una finestra nova amb les diferents opcions 3.- L'usuari selecciona una de les opcions 4.- El sistema mostra el nou tipus de taula per pantalla
<i>Excepcions</i>	Cap
<i>Observacions</i>	El format taula permet seleccionar 4 tipus de visualització en funció de certes propietats: caràcter químic (per defecte), electronegativitat, radi atòmic, estat d'oxidació

ETP3	veureLlista
<i>Resum</i>	Permet veure els elements disposats en format llista
<i>Actors</i>	Usuari,Usuari avançat
<i>Casos d'us relacionats</i>	ordenarLlista
<i>Precondició</i>	S'ha accedit al mòdul d'elements
<i>Postcondició</i>	Es visualitzen les dades per pantalla
<i>Descripció</i>	1.- L'usuari selecciona veure les dades (elements) disposats sobre una llista ordenada 2.- El sistema ofereix per pantalla la visualització demanada
<i>Excepcions</i>	Cap
<i>Observacions</i>	Cap

ETP4	ordenarLlista
<i>Resum</i>	Permet ordenar les dades de la llista
<i>Actors</i>	Usuari,Usuari avançat
<i>Casos d'us relacionats</i>	veureLlista
<i>Precondició</i>	L'usuari ha seleccionat el format de taula com a llista
<i>Postcondició</i>	Es visualitza la llista, ordenada, en funció de la selecció feta
<i>Descripció</i>	1.- L'usuari tria l'opció d'ordenar la llista 2.- El sistema obre una finestra nova amb les diferents opcions 3.- L'usuari selecciona una de les opcions 4.- El sistema mostra per pantalla la llista ordenada segons el paràmetre triat
<i>Excepcions</i>	Cap
<i>Observacions</i>	El format llista permet seleccionar 7 tipus d'ordenació: per número atòmic (per defecte), alfabèticament, per grup, per període, per caràcter químic, per electronegativitat, i pes atòmic

ETP5	seleccionarElement
<i>Resum</i>	Permet accedir a les característiques d'un element
<i>Actors</i>	Usuari,Usuari avançat
<i>Casos d'us relacionats</i>	veurePropietatsBàsiques, veurePropietatsAvançades, veurePropietatsAtòmiques,
<i>Precondició</i>	S'ha accedit a la taula/llista d'elements

<b>Postcondició</b>	Es mostren les dades de la selecció
<b>Descripció</b>	1.- L'usuari selecciona un dels elements 2.- El sistema obre una finestra que permet fer la selecció del tipus de dades a visualitzar
<b>Excepcions</b>	Cap
<b>Observacions</b>	Cap

<b>ETP6</b>	<b>veurePropietatsBàsiques</b>
<b>Resum</b>	Permet veure la informació bàsica d'un element
<b>Actors</b>	Usuari,Usuari avançat
<b>Casos d'us relacionats</b>	veurePropietatsAtòmiques, veurePropietatsAvançades, seleccionarElement
<b>Precondició</b>	S'ha accedit a la selecció de propietats de l'element
<b>Postcondició</b>	Es visualitzen les dades bàsiques per pantalla
<b>Descripció</b>	1.- L'usuari selecciona de la llista l'opció ' <i>basic properties</i> ' 2.- Com a resultat, obté les dades demanades
<b>Excepcions</b>	Cap
<b>Observacions</b>	Si l'usuari vol sortir cap a la llista / taula ha de prémer el botó de la barra (superior) de navegació ' <i>periodic table</i> '

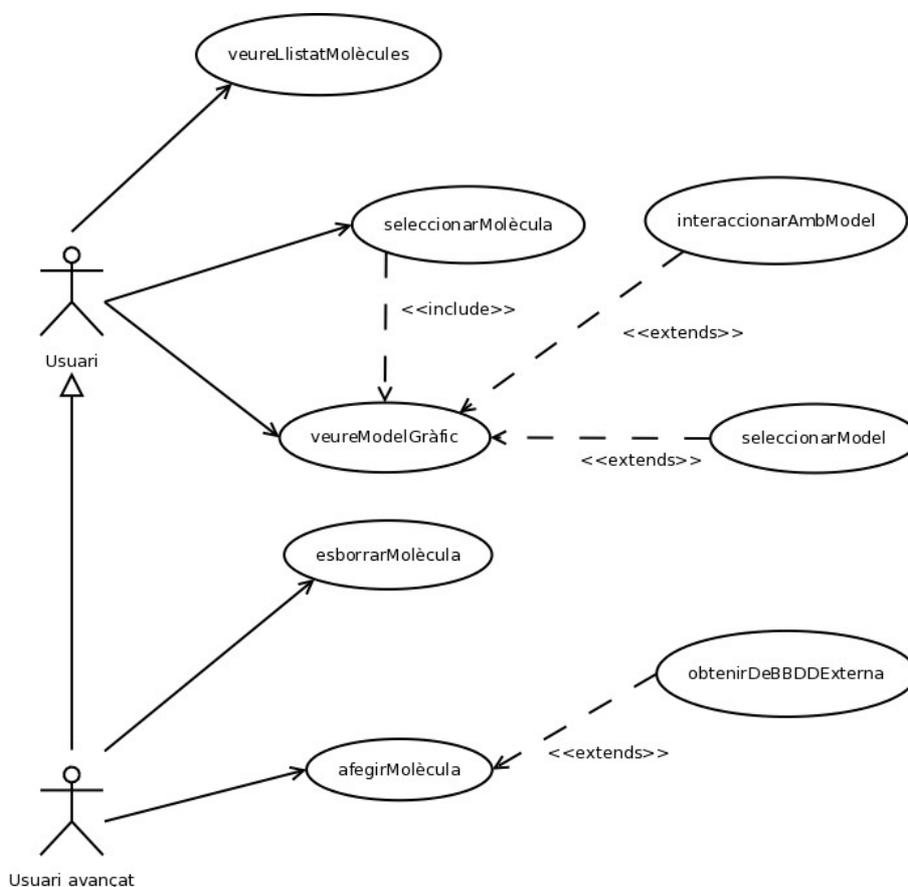
<b>ETP7</b>	<b>veurePropietatsAtòmiques</b>
<b>Resum</b>	Permet veure la informació d'un element referent al seu àtom
<b>Actors</b>	Usuari,Usuari avançat
<b>Casos d'us relacionats</b>	veurePropietatsBàsiques, veurePropietatsAvançades, seleccionarElement
<b>Precondició</b>	S'ha accedit a la selecció de propietats de l'element
<b>Postcondició</b>	Es visualitzen les dades atòmiques per pantalla
<b>Descripció</b>	1.- L'usuari selecciona de la llista l'opció ' <i>atomic properties</i> ' 2.- Com a resultat, obté les dades demanades
<b>Excepcions</b>	Cap
<b>Observacions</b>	Si l'usuari vol sortir cap a la llista / taula ha de prémer el botó de la barra (superior) de navegació ' <i>periodic table</i> '

ETP8	veurePropietatsAvançades
<i>Resum</i>	Permet veure la informació avançada d'un element
<i>Actors</i>	Usuari avançat
<i>Casos d'us relacionats</i>	veurePropietatsBàsiques, veurePropietatsAtòmiques, seleccionarElement
<i>Precondició</i>	S'ha accedit a la selecció de propietats de l'element
<i>Postcondició</i>	Es visualitzen les dades avançades per pantalla
<i>Descripció</i>	1.- L'usuari selecciona de la llista l'opció ' <i>advanced properties</i> ' 2.- Com a resultat, obté les dades demanades
<i>Excepcions</i>	Cap
<i>Observacions</i>	Si l'usuari vol sortir cap a la llista / taula ha de prémer el botó de la barra (superior) de navegació ' <i>periodic table</i> '

ETP9	utilitzarCalculadoraMolar
<i>Resum</i>	Permet fer servir la calculadora molar
<i>Actors</i>	Usuari avançat
<i>Casos d'us relacionats</i>	Cap
<i>Precondició</i>	S'ha accedit al mòdul de la calculadora molar
<i>Postcondició</i>	S'obté el resultat, en grams per mol, i d'altres informacions associades a la fórmula inserida
<i>Descripció</i>	1.- L'usuari selecciona accedir a la calculadora 2.- L'usuari insereix la fórmula amb la que realitzar el càlcul 3.- El sistema torna el valor en mols de l'operació, a més d'estadístiques per element
<i>Excepcions</i>	Si la fórmula introduïda conté errors, el sistema avisa, permetent corregir l'error o fer una nova operació
<i>Observacions</i>	Cap

### 3.3.2 Molècules i visualitzacions

El diagrama de casos d'us d'aquest bloc es mostra a continuació.



Imatge 18: Casos d'us elements pel Visualitzador de molècules

Un cop definit l'esquema de casos d'us s'han d'especificar cada un d'ells de manera textual.

MV1	veureLlistatMolècules
<b>Resum</b>	Permet veure el llistat de molècules disponibles
<b>Actors</b>	Usuari, Usuari avançat
<b>Casos d'us relacionats</b>	Cap
<b>Precondició</b>	S'ha accedit al mòdul de molècules
<b>Postcondició</b>	Es mostra la llista de molècules
<b>Descripció</b>	1.- L'usuari selecciona veure el llistat de molècules disponibles 2.- El sistema mostra el llistat demanat
<b>Excepcions</b>	Cap
<b>Observacions</b>	L'ordre de visualització del llistat és alfabètic descendent

MV2	seleccionarMolècula
<i>Resum</i>	Permet accedir a la visualització d'una molècula
<i>Actors</i>	Usuari, Usuari avançat
<i>Casos d'us relacionats</i>	veureModelGràfic
<i>Precondició</i>	S'ha accedit al llistat de molècules
<i>Postcondició</i>	Es mostra la visualització de la molècula seleccionada
<i>Descripció</i>	1.- L'usuari selecciona una molècula de les disponibles a la llista 2.- El sistema accedeix a la representació gràfica
<i>Excepcions</i>	Cap
<i>Observacions</i>	Cap

MV3	veureModelGràfic
<i>Resum</i>	Permet veure el model gràfic d'una molècula
<i>Actors</i>	Usuari, Usuari avançat
<i>Casos d'us relacionats</i>	seleccionarMolècula, interaccionarAmbModel, seleccionarModel
<i>Precondició</i>	S'ha seleccionat una molècula
<i>Postcondició</i>	Es mostra el model gràfic en pantalla
<i>Descripció</i>	El sistema mostra la visualització del model per pantalla
<i>Excepcions</i>	Cap
<i>Observacions</i>	La visualització per defecte és 'ballsAndSticks' (3D). La barra superior de navegació mostra dos botons: 'back', per tornar a la llista, i un altre per seleccionar el model de visualització (2D o 3D, en funció de l'actual) Si no existeix representació de la molècula pel model establert, el sistema alerta a l'usuari i es torna al llistat

MV4	seleccionarModel
<i>Resum</i>	Permet seleccionar el model gràfic de visualització
<i>Actors</i>	Usuari, Usuari avançat
<i>Casos d'us relacionats</i>	veureModelGràfic

<b>Precondició</b>	S'ha accedit al model gràfic
<b>Postcondició</b>	Es mostra la visualització en el model escollit
<b>Descripció</b>	1.- L'usuari selecciona el model 2.- El sistema mostra la molècula segons el model triat
<b>Excepcions</b>	Si la representació gràfica del model no existeix, s'alerta a l'usuari i es torna a la visualització anterior
<b>Observacions</b>	Els models són: <i>ballsAndSticks</i> (3D) i <i>developedFormula</i> (2D)

MV5	interaccionarAmbModel
<b>Resum</b>	Permet interaccionar amb el model gràfic
<b>Actors</b>	Usuari, Usuari avançat
<b>Casos d'us relacionats</b>	Cap
<b>Precondició</b>	S'ha accedit al model gràfic
<b>Postcondició</b>	Es mostra el resultat de la interacció
<b>Descripció</b>	1.- L'usuari interactua amb el model gràfic via control tàctil 2.- El sistema mostra la representació d'aplicar les transformacions pertinents
<b>Excepcions</b>	Si el model no permet algun tipus de transformació l'usuari ho detecta amb la no resposta al moviment fet
<b>Observacions</b>	La interacció pot consistir en desplaçar, rotar o ampliar

MV6	esborrarMolècula
<b>Resum</b>	Permet eliminar una molècula de la llista
<b>Actors</b>	Usuari avançat
<b>Casos d'us relacionats</b>	Cap
<b>Precondició</b>	S'ha accedit al mòdul de molècules, la molècula existeix i ha estat inserida manualment
<b>Postcondició</b>	Es mostra la llista de molècules sense l'entrada esborrada.
<b>Descripció</b>	1.- L'usuari selecciona el mode edició 2.- L'usuari selecciona la molècula a esborrar del llistat 3.- El sistema requereix confirmació de l'operació 4.- Amb la validació de l'usuari s'esborra l'entrada i es mostra la llista

	resultant 5.- L'usuari surt del mode edició
<b>Excepcions</b>	Si la molècula pertany al grup de predefinides, l'operació no es pot dur a terme
<b>Observacions</b>	L'opció d'edició (botó 'edit') es troba a la barra superior de navegació. En mode edició, pren el nom 'done', i permet tancar aquest mode Per esborrar un element de la llista, s'ha de prémer el botó situat a l'esquerra de l'entrada

MV7	afegirMolècula
<b>Resum</b>	Permet inserir una molècula al llistat
<b>Actors</b>	Usuari avançat
<b>Casos d'us relacionats</b>	obtenirDeBBDDExterna
<b>Precondició</b>	S'ha accedit al mòdul de molècules
<b>Postcondició</b>	La nova molècula forma part del llistat
<b>Descripció</b>	1.- L'usuari selecciona l'opció d'edició 2.- El sistema mostra les opcions d'inserció 3.- L'usuari selecciona l'opció d'inserir nova molècula del llistat 4.- El sistema redirigeix cap a la finestra d'inserció
<b>Excepcions</b>	Cap
<b>Observacions</b>	Per afegir una entrada, es selecciona la darrera entrada del llistat (etiquetada com 'Add new molecule').

MV8	obtenirDeBBDDExterna
<b>Resum</b>	Permet fer la cerca de la molècula en una BBDD externa, descarregar-la i afegir-la a l'aplicació
<b>Actors</b>	Usuari avançat
<b>Casos d'us relacionats</b>	afegirMolècula
<b>Precondició</b>	S'ha accedit al mode d'inserció
<b>Postcondició</b>	Es torna a la llista de molècules
<b>Descripció</b>	1.- L'usuari insereix el nom de la molècula a cercar 2.- L'usuari indica l'inici de la cerca 3.- El sistema verifica l'existència de la molècula a la BBDD

	<p>1a.- si existeix, mostra la informació associada</p> <p>1a1.- L'usuari accepta la descàrrega</p> <p>1a1a.- El sistema descarrega la molècula i l'afegeix a <i>iCompound</i></p> <p>1a2.- L'usuari cancel·la el procés</p> <p>1b.- si no existeix, el sistema avisa de la incidència</p> <p>4a.- L'usuari torna al llistat</p> <p>4b.- L'usuari fa una nova cerca</p>
<b>Excepcions</b>	Si la molècula no existeix, o és incorrecta, s'adverteix a l'usuari i es dona l'opció de realitzar noves cerques. S'actua igualment un cop descarregada la nova molècula (fer nova cerca).
<b>Observacions</b>	Cap

### 3.4 Prototipat de baix nivell

Un cop coneguts i definits els casos d'us implicats, s'ha de fer el disseny a baix nivell de les interfícies, o *wireframes*. A continuació es mostren els diferents prototips corresponents a cada situació definida.

#### 3.4.1 Inici de l'aplicació.



Imatge 19: Pantalla d'inici de *iCompound*

Tot just després d'una pantalla d'inici, l'usuari accedeix al mòdul d'elements (concretament al llistat), on troba per primera vegada l'arrel de l'esquema de navegació.

Per mitjà al *Tab Bar* inferior (present en tot moment) és possible utilitzar les diferents eines que ofereix *iCompound*:

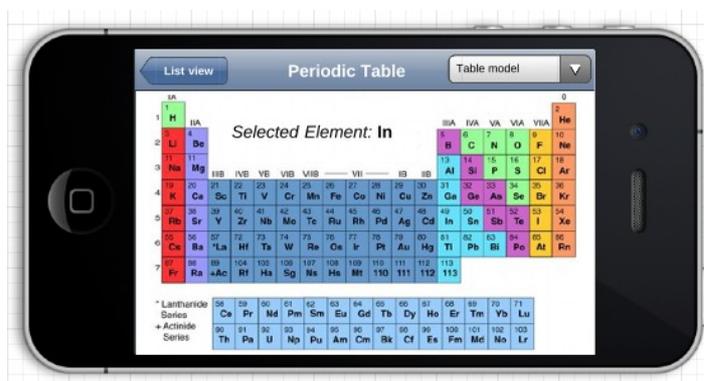
- Taula periòdica, a la qual es pot accedir prement la primera pestanya, etiquetada com '*Periodic Table*'.
- Calculadora molar, accessible prement la segona pestanya (etiquetada com '*Molar Calculator*').
- Visualitzador de molècules. S'accedeix prement la tercera pestanya ('*Molecule Viewer*').

### 3.4.2 Taula Periòdica

#### 3.4.2.1 Format taula

Els elements queden disposats en una taula (visualització clàssica), la qual queda ordenada per grups (columnes), període (files) i el nombre atòmic (de 1 a 188).

L'usuari selecciona l'element posant-se sobre la casella en qüestió. La barra superior permet diverses opcions:



Imatge 20: Taula periòdica de iCompound

- Tornar al llistat d'elements (botó 'List view').
- Seleccionar un model de taula en funció d'unes propietats prefixades (selector desplegable 'Table model').

#### 3.4.2.2 Format llista



Imatge 21: Taula en format llista + criteris d'ordenació

És el format que l'usuari troba d'inici en accedir a l'eina de la taula periòdica. Els elements es visualitzen en una llista ordenada (per defecte, segons caràcter químic) i són seleccionables en prémer la cel·la desitjada.

La llista es ordenable segons diferents característiques prement el botó 'Order Criteria' de la barra de navegació inferior. La nova vista permet seleccionar un criteri d'ordenació clicant sobre el valor escollit, o bé tornar al llistat (amb el botó de la barra superior 'Periodic Table'). En ambdós casos, el retorn a la llista és automàtic, i aplica la selecció feta amb efecte immediat..

Tanmateix, és possible escollir la visualització de taula mitjançant el botó 'Table view'.

### 3.4.2.3 Propietats de l'element seleccionat



Seleccionar un element, tant en el mode llista com en el mode taula, obre una nova vista en la qual és possible seleccionar les diferents propietats de la selecció feta.

La navegació és simple: fent clic sobre la cel·la del grup de propietats a examinar, s'accedeix a la vista que conté la informació relacionada. Dintre de cada una d'elles, és possible tornar a la selecció de propietats fent servir el botó 'Properties' de la barra superior de navegació.

Si l'usuari vol torna a la llista (o taula), ho pot fer prement el botó 'Periodic Table' de la barra superior, en la vista de selecció de propietats.

Imatge 22: Selector de propietats



Imatge 23: Vistes de propietats

### 3.4.3 Calculadora molar

La calculadora molar permet obtenir la molaritat (pes) d'una molècula/compost químic en g/mol, a més d'informació relativa a la composició.

L'usuari introdueix, via teclat alfanumèric, la fórmula corresponent a la caixa de text '*Formula*', i obté el resultat en l'espai entre el teclat i la caixa prement el botó vermell etiquetat amb un signe d'igualtat. Destacar que el teclat només s'activa en editar la caixa de text, i un cop feta l'operació (o en sortir del camp '*Formula*', sense fer-la) desapareix fins que es produeixi una de nova.

Recordar que es pot passar d'aquesta eina a les altres amb la *Tab Bar* inferior, només oculta quan el teclat està present.



Imatge 24: Calculadora molar

### 3.4.4 Visor de molècules.

#### 3.4.4.1 Llistat de molècules.



Imatge 25: Llistat de molècules

Al accedir a l'eina de visualització de molècules, el primer que es veu és una llista ordenada alfabèticament amb les molècules disponibles dintre de l'aplicació. L'usuari pot accedir a diferents tasques des d'aquesta pantalla:

- Veure la representació 2D/3D. S'ha de prémer el botó blau a la dreta del nom.
- Editar el llistat. Prement el botó *Edit* de la barra superior es pot editar el llistat (afegir noves entrades o modificar/esborrar entrades existents).
- Canviar a alta eina, fent servir la *Tab Bar* inferior.

### 3.4.4.2 Mode d'edició.

En el mode d'edició es pot modificar el llistat de molècules disponibles. Diferents controls permeten efectuar les operacions:

- Inserir una entrada. Prement la cel·la on es troba el botó amb signe '+' sobre fons verd (entrada anomenada 'Add new molecule') es pot accedir a la vista d'inserció de noves molècules.
- Esborrar una entrada. S'ha de prémer el símbol de prohibició vermell situat a l'esquerra de la cel·la seleccionada. Llavors, apareix un botó vermell a la dreta de l'entrada que permet realitzar l'operació.
- Sortir del mode edició. S'ha de prémer el botó vermell de la barra superior ('Done') per tornar a la llista inicial un cop fets els canvis.



Imatge 26: Edició de molècules

### 3.4.4.3 Esborrat de molècules.

Cal destacar que tant les accions d'esborrat com d'edició donen a l'usuari l'opció de no realitzar l'operació, i la confirmació en cas que decideixi seguir endavant.

És important recordar que l'usuari només pot esborrar entrades creades per ell mateix, mai de les predefinides (ja instal·lades amb l'aplicació).

Es mostren les finestres corresponents a l'esborrat; les de modificació són idèntiques, a excepció que el missatge que es mostra a l'usuari és diferent.



Imatge 27: Esborrat de molècules

#### 3.4.4.4 Inserció de molècules

L'usuari pot inserir noves molècules a la llista de les ja existents (predefinides) fent servir una vista d'inserció, la qual conté un cercador que permet a l'usuari introduir el nom de la molècula a cercar. Un cop l'ha trobat, tindrà l'opció de descarregar-la fent servir el botó '*Download*' situat a una barra inferior.

Cal puntualitzar l'operació, ja que el seu comportament és important de cara a la visualització de les molècules. Una molècula predefinida adjunta un model en 3D en un format específic que permet la renderització. A més, és possible fer la seva representació com a fórmula desenvolupada (2D), la qual també té el seu propi model en un format diferent al anteriorment citat.

Remarcar que l'operació només es durà a terme si la molècula a descarregar no és ja part de la base d'*iCompound* (de manera predefinida o per adquisició prèvia).



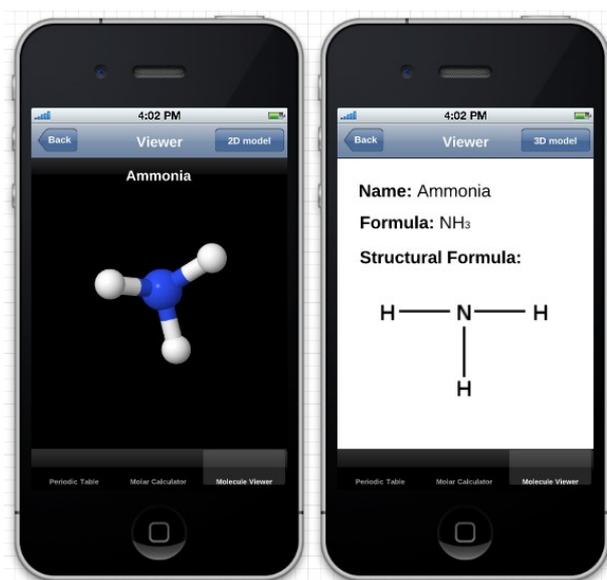
Imatge 28: Cerca i inserció de molècules

#### 3.4.4.5 Visor de molècules.

Quan l'usuari selecciona veure una molècula, per defecte (i sempre que existeixi) es mostra el model en 3D, conegut com '*Balls and Sticks*'. Si no existeix, es mostra el bidimensional ('*Developed Formula*').

L'usuari pot commutar entre les visualitzacions fent servir el botó situat a la dreta de la barra superior de navegació. A més, en el cas del mode tridimensional el pot moure (rotar, desplaçar i fer zoom) posant-se sobre la pantalla i fent servir els controls tàctils propis del dispositiu.

Tanmateix, per tornar al llista de molècules, s'ha de prémer el botó '*Back*', situat a la barra superior de navegació.



Imatge 29: Visor de molècules

### 3.5 Disseny de la interfície

El disseny de la interfície es realitza amb l'eina *Interface Builder* en la seva pràctica totalitat, tot i que algunes de les vistes es generen directament amb codi C-Objective, i han de respondre a l'aspecte visual definit amb anterioritat.

Encara que el disseny de la interfície pugui quedar inclosa en una vessant més teòrica, es considera que aquesta és una part indivisible a la pròpia implementació, al començar ja a establir certes rutines de comportament associades a codi font. És per això, que el disseny de la interfície s'inclou en la part d'implementació.

## 4 Implementació

Aquesta és la fase on es procedeix a generar tot el codi de programació en Objective-C, emprant els frameworks necessaris (com *Cocoa-Touch*, *OpenGL ES*, *Core Graphics*, etc) per poder crear l'aplicació. Tant mateix, es genera tota la part de persistència associada per tal d'oferir les funcionalitats establertes en la fase de disseny.

Abans d'aprofundir en aquest procés cal aclarir certs conceptes sobre els quals s'ha de fonamentar la fase d'implementació.

### 4.1 Consideracions prèvies

#### 4.1.1 Patrons de disseny

Els patrons de disseny són solucions pràctiques d'alta qualitat que permeten als programadors aplicar una sèrie de bones pràctiques, amb la finalitat d'aconseguir complir amb les màximes de reusabilitat, acoblament, encapsulació, etc, tot generant un codi font elegant i ben estructurat.

##### 4.1.1.1 Model vista controlador (MVC)

Un dels més antics i eficients és el conegut com a Model Vista Controlador (MVC). És un patró de disseny optimitzat per garantir la cooperació entre grans quantitats d'objectes, establint un alt grau d'organització a través de tres subsistemes:

- **Model:** Subsistema format per objectes que garanteixen la interacció amb els sistemes d'emmagatzemament de dades que ha de fer servir l'aplicació.
- **Vista:** Subsistema format per objectes que permeten presentar la informació referent a les dades obtingudes amb el subsistema Model, a més de ser l'encarregat de garantir la interacció entre l'usuari i la informació.
- **Controlador:** Subsistema encarregat de garantir el desacoblament entre les dades i les interfícies (és a dir, entre el Model i les Vistes). Interactua entre els altres dos subsistemes, processant la informació per que sigui visualitzable i/o emmagatzemable, segons el flux de treball entre ambdós.

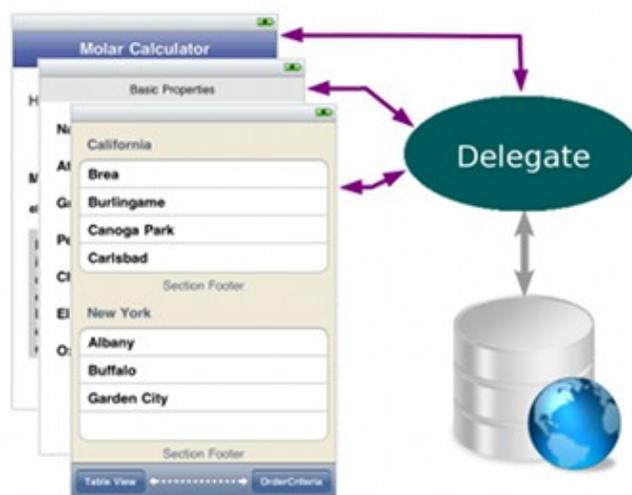


Imatge 30: Model Vista Controlador

Aquest model és, principalment, sobre el que s'organitza Cocoa i d'altres frameworks de gran utilitat en el desenvolupament d'aplicacions amb iOS. Així doncs, i tenint en compte els avantatges que aporta aquest patró, es decideix que MVC sigui l'emprat a l'hora d'implementar *iCompound*.

#### 4.1.1.2 Delegat (*delegate*)

És un dels patrons de programació utilitzats per millorar el desacoblament el subsistema Model i els altres. De gran senzillesa i versatilitat, te com a finalitat fer d'intermediari entre un o diversos objectes amb els quals intercanvia missatges per dur a terme certes tasques o informar de certes situacions. Tot això es sustenta sobre un protocol bàsic que s'estableix d'inici i que han de conèixer tant el delegat com l'objecte que delega.



Imatge 31: Patró Delegate

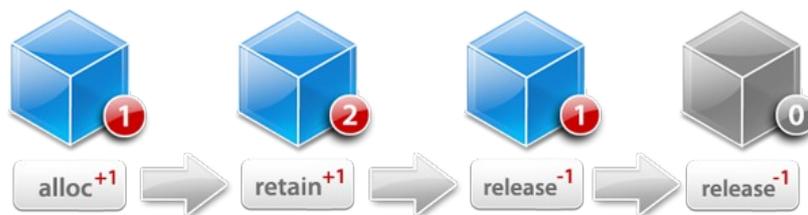
Per la seva versatilitat i senzillesa ofereix un servei d'intermediari molt útil, ja que reacciona davant les peticions dels objectes al qual serveix, i a l'hora pot determinar el seu funcionament. Soluciona problemes de coordinació entre objectes de manera transparent, fet que fa que sigui altament reutilitzable, minimitzant l'acoblament i simplificant el desenvolupament.

### 4.1.2 Gestió de la memòria en dispositius mòbils amb iOS

A diferència d'altres llenguatges (com *Java*, base d'*Android*) Objective-C no fa ús per defecte del que es coneix com a *garbage collector* per alliberar recursos de memòria de manera eficient i totalment transparent pel programador. La gestió de memòria en aquest llenguatge requereix que el s'alliberin de manera explícita els recursos creats quan ja no tenen més utilitat. Una gestió poc eficient pot arribar a produir problemes de memòria (*memory leaks*) que poden derivar en un consum de recursos inassequible pels dispositius (d'altra banda, limitats en aquest aspecte tecnològicament).

Tot i que és possible crear objectes que siguin alliberats de manera automàtica pel sistema (gràcies a la *Autorelease Pool* oferida pel framework Cocoa), no és recomanable ja que pel seu funcionament (que ajorna l'alliberament d'objectes) pot alentir de manera notable l'aplicació.

Així doncs, és responsabilitat del programador fer un alliberament de memòria destruint els objectes creats o retinguts seguint el que es coneix com *alloc-retain-release cycle*.



Imatge 32: Gestió de la memòria en iOS. Alloc-retain-release cycle.

Els detalls d'aquest procés s'allunyen del propòsit d'aquest informe. Per trobar més informació al respecte es poden consultar algunes de les referències contingudes a la bibliografia.

### 4.1.3 Persistència

Tal i com s'ha detallat en la fase de disseny, és imprescindible tenir certa quantitat de dades emmagatzemades i que aquestes siguin totalment accessibles.

Existeixen diversos mecanismes que ofereixen persistència de dades per programari desenvolupat per dispositius iOS (totes fan ús del seu sistema d'arxius):

- **Llistes de propietats.** Son fitxer editables de manera manual, que permeten emmagatzemar dades de manera ordenada. Indicades quan el volum de dades és petit i limitada en ús a certs objectes, és d'utilitat per exemple per guardar informació sobre configuració / preferències d'usuari.
- **Arxius d'objectes.** Basat en la *serialització*, permet emmagatzemar i recuperar objectes en fitxers de text. Quan el volum de dades (objectes i/o instàncies dels mateixos) creix la seva gestió és més complicada i ineficient.
- **Core Data.** És l'eina que Apple recomana per tasques de persistència. Aquesta API permet realitzar, de manera senzilla i eficient, totes les tasques relacionades amb la interacció amb una estructura de base de dades. Cal remarcar que no és, en cap cas, una base de dades relacional, tot i que pot interaccionar amb elles.

- **SQLite3.** És la bases de dades encastada en iOS. Permet fer les operacions més comuns a qualsevol SGBD de manera eficient, podent tractar amb grans quantitats de dades. Tot això, fent servir el llenguatge estàndard SQL per les consultes.

De les quatre solucions, les més adients per l'aplicació que es vol desenvolupar són Core Data i SQLite3, ja que el volum de dades, tot i estar acotat i tenir una variabilitat (quantitat de registres) poc elevada, és prou gran.

Tot i ser Core Data la solució recomanada per Apple a l'hora d'implementar la persistència en dispositius amb iOS, es considera que pel cas de l'aplicació *iCompound* és més adient utilitzar SQLite3, principalment per la senzillesa de l'estructura de dades a crear i per la potència que aquest SGBD ofereix (remarcant que fins l'aparició de *Core Data* era l'estàndard recomanat per Apple). A més, les operacions a realitzar no han de suposar un elevat cost pel que es refereix a recursos.

Cal afegir que, el cas de les molècules, quan l'usuari decideix afegir una de nova a la base de l'aplicació, es descarregarà un fitxer de text del tipus *molFile* (extensió *.mol*). Aquest fitxer conté les dades necessàries per a poder realitzar la representació gràfica, i quedarà emmagatzemat a la carpeta *Documents* de l'usuari. Tot i que la resta de la informació referent al compost adquirit s'inserirà en la base de dades, resulta més eficient i senzill tenir un repositori estàndard de dades gràfiques pel seu posterior tractament sobre fitxers de text. Als annexos es pot trobar una petita referència sobre el format triat per aquesta tasca.

## 4.2 Decisions preses en el procés d'implementació

Per poder dur a terme el procés d'implementació i oferir un producte 100% funcional ha estat necessari introduir variacions en l'aplicació. Tot i que la idea inicial i el disseny fet és el que ha de tenir en el l'aplicació en el seu format final (al menys, quant a funcionalitats), el temps de que es disposa per dur a terme el projecte és limitat.

Aquestes modificacions són especialment sensibles en el mòdul de molècules. Tenint en compte l'alta corba d'aprenentatge de Open GL ES, es decideix que dels dos formats de visualització sigui el de tres dimensions el sacrificat. No obstant, i per compensar, o més bé, per que es pugui tenir una idea de com hauria de quedar, es millora la representació 2D (fórmula desenvolupada) afegint un cert aspecte tridimensional (això sí, no interactiu), aprofitant les característiques gràfiques que ofereix el framework Core Graphics. D'aquesta manera, s'aconsegueix oferir una bona aproximació visual de les molècules.

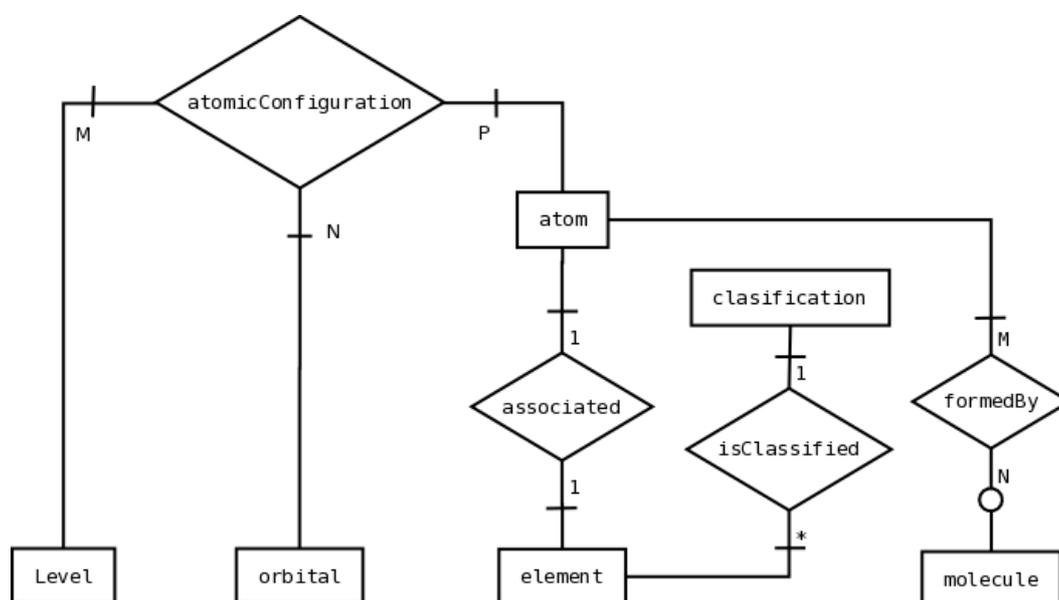
Tanmateix, es poden veure certes variacions addicionals fetes per introduir millores respecte al disseny teòric, derivades de l'intent de fer un procés amb certa iterativitat, sempre destinat a la cerca d'un increment de la usabilitat i satisfacció d'usuari.

### 4.3 Implementació del subsistema Model

El Model és el subsistema encarregat d'interaccionar amb la persistència (que tal com s'ha dit utilitzarà el SGBD SQLite3).

#### 4.3.1 Persistència dissenyada. Modificacions.

Tot i que en la fase de disseny es va definir el model de dades a utilitzar, en la implementació final aquest queda lleugerament modificat, arribant al model que es mostra a continuació:



Imatge 33: Persistència per iCompound (millorada)

Les modificacions també afecten al model relacional de dades:

ELEMENT (element-name, chemical-character, group, period, aggregation-state, oxidation-level, boiling-point, melting-point, electronegativity, covalent-radius, first-ionization-potencial, atomic-name, **classification**)  
 on {atomic-name} referencia ATOM  
 on {classification} referencia CLASSIFICATION

CLASSIFICATION (idClass, name)

### 4.3.2 Implementació de la persistència

El model relacional s'implementa sobre SQLite3, i el procés s'ha de dividir en 2 fases. La primera, la creació de les taules i restriccions/disparadors, i la segona, la introducció de les dades associades al projecte. A continuació es mostra el codi SQL necessari per dur a terme aquestes tasques:

#### 4.3.2.1 Creació de taules

```
CREATE TABLE orbital(orbitalName CHAR not null PRIMARY KEY, maxElectrons INTEGER not null)
CREATE TABLE level(levelId INTEGER not null PRIMARY KEY, maxElectrons INTEGER not null)
CREATE TABLE classification(idClass INTEGER NOT NULL PRIMARY KEY, name TEXT NOT NULL)
CREATE TABLE atom(atomicName TEXT not null PRIMARY KEY, atomicWeight FLOAT not null,
    atomicRadius FLOAT not null, atomicVolume FLOAT not null)
CREATE TABLE atomicConfiguration(atomicName TEXT not null, level INTEGER not null,
    orbital CHAR not null, electrons INTEGER not null DEFAULT 0,
    FOREIGN KEY(atomicName) REFERENCES atom(atomicName),
    FOREIGN KEY(level) REFERENCES level(levelId),
    FOREIGN KEY(orbital) REFERENCES orbital(orbitalName))
CREATE TABLE element(atomicNumber INTEGER NOT NULL PRIMARY KEY, atomicName TEXT NOT NULL,
    elementName TEXT NOT NULL, period INTEGER NOT NULL,
    egroup TEXT NOT NULL, oxidationState TEXT NOT NULL,
    boilingPoint REAL NOT NULL, meltingPoint REAL NOT NULL,
    electronegativity REAL NOT NULL, covalentRadius REAL NOT NULL,
    firstIonizationPotential REAL NOT NULL, classification INTEGER NOT NULL,
    density float not null default 0.0, FOREIGN KEY(atomicName)
    REFERENCES atom(atomicName), FOREIGN KEY(classification)
    REFERENCES classification(idClass))
CREATE TABLE molecule(inchi TEXT not null PRIMARY KEY, moleculeName TEXT not null,
    formule TEXT not null, fileId TEXT not null);
```

#### 4.3.2.2 Creació de triggers i restriccions

```
CREATE TRIGGER fki_classification
    BEFORE INSERT ON element
    FOR EACH ROW BEGIN
    SELECT CASE
    WHEN ((NEW.classification IS NOT NULL) AND ((SELECT idClass FROM classification
        where idClass = NEW.classification) IS NULL))
    THEN RAISE(ABORT, 'INSERT on table "element" violates level foreign key constraint
        "fk_classification"')
    END;
END
```

```
CREATE TRIGGER fki_levels
  BEFORE INSERT ON atomicConfiguration
  FOR EACH ROW BEGIN
  SELECT CASE
    WHEN ((NEW.level IS NOT NULL) AND ((SELECT levelId FROM level where levelId = NEW.level)
      IS NULL))
      THEN RAISE(ABORT, 'INSERT on table "atomicConfiguration" violates level foreign
        key constraint "fk_levelId"')
  END;
END

CREATE TRIGGER fki_orbitals
  BEFORE INSERT ON atomicConfiguration
  FOR EACH ROW BEGIN
  SELECT CASE
    WHEN ((NEW.orbital IS NOT NULL) AND ((SELECT orbitalName FROM orbital where
      orbitalname = NEW.orbital) IS NULL))
      THEN RAISE(ABORT, 'INSERT on table "atomicConfiguration" violates level foreign
        key constraint "fk_orbitalName"')
  END;
END

CREATE TRIGGER fki_electronsPerOrbital
  BEFORE INSERT ON atomicConfiguration
  FOR EACH ROW BEGIN
  SELECT CASE
    WHEN ((NEW.electrons IS NOT NULL) AND ((SELECT maxElectrons FROM orbital where
      orbitalName = NEW.orbital) < NEW.electrons))
      THEN RAISE(ABORT, 'INSERT on table "atomicConfiguration" violates level foreign
        key constraint "fk_electronsPerOrbital"')
  END;
END

CREATE TRIGGER fki_atom
  BEFORE INSERT ON atomicConfiguration
  FOR EACH ROW BEGIN
  SELECT CASE
    WHEN ((NEW.atomicName IS NOT NULL) AND ((SELECT atomicName FROM atom where
      atomicName = NEW.atomicName) IS NULL))
      THEN RAISE(ABORT, 'INSERT on table "atomicConfiguration" violates level foreign
        key constraint "fk_atomicName"')
  END;
END
```

```
CREATE TRIGGER fki_atom_element
  BEFORE INSERT ON element
  FOR EACH ROW BEGIN
  SELECT CASE
  WHEN ((NEW.atomicName IS NOT NULL) AND ((SELECT atomicName FROM atom where
    atomicName = NEW.atomicName) IS NULL))
    THEN RAISE(ABORT, 'INSERT on table "element" violates level foreign key constraint
      "fk_atomicName_element"')
  END;
END
```

### 4.3.2.3 Inicialització de dades

A continuació es mostra un petit extracte de les insercions realitzades. Per més detall, consultar l'apartat d'annexos.

```
INSERT INTO "level" VALUES(1,2), (2,8), (3,18), (4,32), (5,32), (6,18), (7,8);

INSERT INTO "orbital" VALUES('s',2), ('p',6), ('d',10), ('f',14);

INSERT INTO "classification" VALUES(0,'Metalloids'), (1,'Alkali metals'),
                                     (2,'Alkali earth metals'), (3,'Transition metals'),
                                     (4,'Other metals'), (5,'Non-metals'), (6,'Halogens'),
                                     (7,'Noble gases'), (8,'Rare Earth Metals L'),
                                     (9,'Rare Earth Metals A');

INSERT INTO "atom" VALUES('H',1.00797,0.53,14.2);
INSERT INTO "atom" VALUES('Li',6.939,1.55,12.97);
INSERT INTO "atom" VALUES('Be',9.0122,1.12,4.87);
INSERT INTO "atom" VALUES('He',4.0026,'-',31.77);
INSERT INTO "atom" VALUES('B',10.811,0.98,4.68);
INSERT INTO "atom" VALUES('C',12.0111,0.91,5.31);
...
INSERT INTO "atom" VALUES('Hs',277,'-', '-');
INSERT INTO "atom" VALUES('Mt',268.139,'-', '-');
INSERT INTO "atom" VALUES('Ds',281,'-', '-');
INSERT INTO "atom" VALUES('Rg',280,'-', '-');
INSERT INTO "atom" VALUES('Cn',285,'-', '-');
```

```

INSERT INTO "atomicConfiguration" VALUES('C',1,'s',2);
INSERT INTO "atomicConfiguration" VALUES('C',2,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Fe',1,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Fe',2,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Fe',2,'p',6);
INSERT INTO "atomicConfiguration" VALUES('Fe',3,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Fe',3,'p',6);
INSERT INTO "atomicConfiguration" VALUES('Fe',4,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Fe',3,'d',6);
...
INSERT INTO "atomicConfiguration" VALUES('Cn',1,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Cn',2,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Cn',2,'p',6);
INSERT INTO "atomicConfiguration" VALUES('Cn',3,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Cn',3,'p',6);
INSERT INTO "atomicConfiguration" VALUES('Cn',3,'d',10);
INSERT INTO "atomicConfiguration" VALUES('Cn',4,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Cn',4,'p',6);
INSERT INTO "atomicConfiguration" VALUES('Cn',4,'d',10);
INSERT INTO "atomicConfiguration" VALUES('Cn',4,'f',14);
INSERT INTO "atomicConfiguration" VALUES('Cn',5,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Cn',5,'p',6);
INSERT INTO "atomicConfiguration" VALUES('Cn',5,'f',14);
INSERT INTO "atomicConfiguration" VALUES('Cn',5,'d',10);
INSERT INTO "atomicConfiguration" VALUES('Cn',6,'s',2);
INSERT INTO "atomicConfiguration" VALUES('Cn',6,'p',6);
INSERT INTO "atomicConfiguration" VALUES('Cn',6,'d',10);
INSERT INTO "atomicConfiguration" VALUES('Cn',7,'s',2);

INSERT INTO "element" VALUES(1,'H','Hydrogen',1,'1A','1,-1',-257.7,-259.2,2.1,0.32,13.6,5,
0.0709);
INSERT INTO "element" VALUES(2,'He','Helium',1,'9','-','-268.9,-269.7','-','-24.58,7,0.126);
INSERT INTO "element" VALUES(3,'Li','Lithium',2,'1A','1,1,1330,180.5,1,1.23,5.39,1,0.535);
...
INSERT INTO "element" VALUES(110,'Ds','Darmstadtium',7,'8','-','-','-','-','-','-3,-');
INSERT INTO "element" VALUES(111,'Rg','Roentgenium',7,'1B','-','-','-','-','-','-3,-');
INSERT INTO "element" VALUES(112,'Cn','Copernicium',7,'2B','-','-','-','-','-','-3,-');

INSERT INTO molecule VALUES ('XLYOFNQVPJNP-UHFFFAOYSA-N','Water','H2O','7732-18-5-2d');
...
INSERT INTO molecule VALUES ('GZCGUPFRVQAUEE-SLPGGIOYSA-N','Glucose','C6H12O6','50-99-7-2d');

```

### 4.3.3 Gestió de la persistència i classes associades

Un cop establert el model relacional i les dades que ha de contenir, s'ha de garantir l'accés a les mateixes mitjançant el subsistema Model. iOS permet accedir gràcies a la llibreria *sqlite3*, ja integrada en la SDK.

El primer pas és definir una sèrie de classes que permetin realitzar la gestió de les dades d'una manera lògica. Aquestes es corresponen a objectes representatius del sistema que es vol implementar, i permeten emmagatzemar temporalment les dades extretes per tal poder tractar-les. Aquestes classes associades són:

- **AtomicConfiguration.** Classe que permet emmagatzemar informació sobre la configuració atòmica de l'element al qual està lligat, paràmetre que es presenta com una cadena de caràcters (*formula*), relacionada amb els nivells que conté (*levels*).

```
// AtomicConfiguration.h
// ICompound
#import <CoreData/CoreData.h>

@interface AtomicConfiguration : NSObject {
    NSString *formula;
    NSMutableArray *levels;
}
@property (nonatomic, retain) NSString *formula;
@property (nonatomic, retain) NSMutableArray *levels;
@end
```

- **Atom.** Classe que permet emmagatzemar en un objecte les principals dades relacionades amb l'àtom associat a un element. Aquestes dades poden ser numèriques (com el radi, pes o volum), mentre que el nom atòmic ha de ser una cadena de caràcters. A més, permet instanciar un objecte de la classe *atomicConfiguration*, al ésser la configuració atòmica un paràmetre associat a l'àtom (únic).

```
// Atom.h
// ICompound
#import <CoreData/CoreData.h>

@class AtomicConfiguration;

@interface Atom : NSObject {
    NSNumber *atomicRadius;
    NSNumber *atomicWeight;
    NSNumber *atomicVolume;
}
```

```

    NSString *atomicName;
    AtomicConfiguration *atomicConfiguration;
}
@property (nonatomic, retain) NSNumber *atomicRadius;
@property (nonatomic, retain) NSNumber *atomicWeight;
@property (nonatomic, retain) NSNumber *atomicVolume;
@property (nonatomic, retain) NSString *atomicName;
@property (nonatomic, retain) AtomicConfiguration *atomicConfiguration;
@end

```

- **Element.** Classe per crear objectes on s'emmagatzema tota la informació referent a un element. Aquestes dades inclouen instància tant a la classe *Atom* com a *atomicConfiguration*, ja que cada element està indivisiblement lligat a un àtom i la seva configuració.

```

// Element.h
// ICompound
#import <Foundation/Foundation.h>

@class Atom;
@class AtomicConfiguration;

@interface Element : NSObject{
    NSNumber *atomicNumber;
    NSString *atomicName;
    NSString *elementName;
    NSNumber *period;
    NSString *egroup;
    NSString *oxidationState;
    NSNumber *meltingPoint;
    NSNumber *boilingPoint;
    NSNumber *electronegativity;
    NSNumber *covalentRadius;
    NSNumber *firstIonizationPotential;
    NSString *classification;
    NSNumber *density;
    Atom *eAtom;
    AtomicConfiguration *atomicConfiguration;
    NSString *orderCriteria;
}
@property (nonatomic, retain) NSNumber *atomicNumber;
@property (nonatomic, retain) NSString *atomicName;
@property (nonatomic, retain) NSString *elementName;
@property (nonatomic, retain) NSNumber *period;

```

```

@property (nonatomic, retain) NSString *egroup;
@property (nonatomic, retain) NSString *oxidationState;
@property (nonatomic, retain) NSNumber *meltingPoint;
@property (nonatomic, retain) NSNumber *boilingPoint;
@property (nonatomic, retain) NSNumber *electronegativity;
@property (nonatomic, retain) NSNumber *covalentRadius;
@property (nonatomic, retain) NSNumber *firstIonizationPotential;
@property (nonatomic, retain) NSString *classification;
@property (nonatomic, retain) NSNumber *density;
@property (nonatomic, retain) Atom *eAtom;
@property (nonatomic, retain) AtomicConfiguration *atomicConfiguration;
@property (nonatomic, retain) NSString *orderCriteria;
@end

```

- **Molecule.** Classe per crear objectes on s'emmagatzema tota la informació referent a una molècula. Aquestes dades bàsiques permeten tenir informació útil, especialment en processos de cerca.

```

// Molecule.h
// ICompound
#import <Foundation/Foundation.h>

@interface Molecule : NSObject {
    NSString *inchi;
    NSString *moleculeName;
    NSString *formule;
    NSString *fileId;
}
@property (nonatomic, retain) NSString *inchi;
@property (nonatomic, retain) NSString *moleculeName;
@property (nonatomic, retain) NSString *formule;
@property (nonatomic, retain) NSString *fileId;
@end

```

Un cop definides les classes relacionades, és el moment d'accedir a la informació emmagatzemada a la base de dades, fent ús de les funcions que ofereix la llibreria sqlite3.

Hi han diversos mètodes encarregats de les transaccions de dades:

- *(NSString \*)copyDatabaseToDocuments*

Permet accedir a la base de dades i fer una còpia exacta (si pot) al directori de documents d'aplicació. D'aquesta manera l'aplicació pot escriure i llegir dades sobre una còpia de la base, i no directament sobre l'original, situat al directori *resources*, aïllant les dades introduïdes respecte a

actualitzacions de l'aplicació. Això és important, sobretot de cara al mòdul de molècules, en el qual l'usuari pot introduir noves dades.

```
- (NSString *)copyDatabaseToDocuments {
    NSFileManager *fileManager = [NSFileManager defaultManager];
    NSArray *paths =
        NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);
    NSString *documentsPath = [paths objectAtIndex:0];
    NSString *filePath = [documentsPath stringByAppendingPathComponent:@"icompound.sqlite"];
    if ( ![fileManager fileExistsAtPath:filePath] ) {
        NSString *bundlePath = [[[NSBundle mainBundle] resourcePath]
            stringByAppendingPathComponent:@"icompound.sqlite"];
        [fileManager copyItemAtPath:bundlePath toPath:filePath error:nil];
    }
    return filePath;
}
```

La funció descrita retorna el directori on queda allotjada la base de dades.

- ***(void)readElementsFromDatabase:(NSString \*)databasePath:(NSString \*)query***

Permet, coneguda la ubicació dels fitxers de persistència (*databasePath*) i la consulta a fer (*query*) connectar amb la bases de dades i bolcar la informació en els objectes que correspon en cada cas. El procediment utilitzat per fer aquestes tasques és elemental:

1. Establir connexió amb base de dades: *sqlite3\_open([databasePath UTF8String], &database)*
2. Preparar la consulta: *const char \*sqlStatement = [query UTF8String]*
3. Preparar el contenidor de resultat: *sqlite3\_stmt \*compiledStatement*
4. Realitzar consulta: *sqlite3\_prepare\_v2(database, sqlStatement, -1, &compiledStatement, NULL)*
5. Recórrer la llista de resultats: *sqlite3\_step(compiledStatement)*
6. Per cada resultat, extreure les dades en el format corresponent:
  - *sqlite3\_column\_int(compiledStatement, 0)*
  - *sqlite3\_column\_text(compiledStatement, 1)*
7. Bolcar les dades a l'objecte corresponent
8. Alliberar contenidor de resultats: *sqlite3\_finalize(compiledStatement)*
9. Tancar connexió: *sqlite3\_close(database)*

Al final de tot el procés s'obtenen els objectes necessaris per tal que l'aplicació pugui fer les tasques requerides per l'usuari.

```

-(void) readElementsFromDatabase:(NSString *)databasePath:(NSString *)query {
    sqlite3 *database;
    NSDictionary *properties;
    NSString *orderedField = nil;

    if(sqlite3_open([databasePath UTF8String], &database) == SQLITE_OK) {
        const char *sqlStatement = [query UTF8String];
        sqlite3_stmt *compiledStatement;
        if(sqlite3_prepare_v2(database, sqlStatement, -1, &compiledStatement, NULL)
            == SQLITE_OK) {
            while(sqlite3_step(compiledStatement) == SQLITE_ROW) {
                NSNumber *atomicNumber = [NSNumber numberWithInt:
                    sqlite3_column_int(compiledStatement, 0)];
                NSString *atomicName = [NSString stringWithUTF8String:(char *)
                    sqlite3_column_text(compiledStatement, 1)];
                ...
                NSNumber *atomicRadius = [NSNumber numberWithFloat:
                    sqlite3_column_double(compiledStatement, 8)];
                ...
                Element *newElement = [[Element alloc] init];
                newElement.atomicNumber = atomicNumber;
                newElement.atomicName = atomicName;
                ...
                [self.elements addObject:newElement];
                [newElement release];
            }
        }
        else {
            NSLog(@"Failed to open database cat %@ with error %s",
                databasePath, sqlite3_errmsg(database));
        }
        sqlite3_finalize(compiledStatement);
        [sectionKeys addObject:orderedField];
        [self createOrderedDictionaryOfElements];
    }
    sqlite3_close (database);
}

```

En relació a la gestió del model de dades de les molècules, existeixen tres mètodes que intervenen en aquest procés:

- *(void) readMoleculesFromDatabase:(NSString \*)databasePath:(NSString \*)query*

Mètode anàlog a readElementsFromDatabase, amb la diferència que el que es recupera són molècules.

```
...
    NSString *inchi = [NSString stringWithUTF8String:(char *)
        sqlite3_column_text(compiledStatement, 0)];
    NSString *moleculeName = [NSString stringWithUTF8String:(char *)
        sqlite3_column_text(compiledStatement, 1)];
    NSString *formule = [NSString stringWithUTF8String:(char *)
        sqlite3_column_text(compiledStatement, 2)];
    NSString *molFile = [NSString stringWithUTF8String:(char *)
        sqlite3_column_text(compiledStatement, 3)];

    Molecule *molecule = [[Molecule alloc] init];
    molecule.inchi = inchi;
    molecule.moleculeName = moleculeName;
    molecule.formule = formule;
    molecule.fileId = molFile;
    [molecules addObject:molecule];
    [molecule release];
...

```

- *(void) writeMoleculeToDatabase:(Molecule \*)molecule*

Mètode encarregat de inserir una nova molècula a la base de dades. El procediment emprat per aquesta tasca és idèntic a l'utilitzat per llegir dades, canviant el format de la consulta i preparant prèviament les dades en objectes de tipus text per tal de poder dur a terme l'operació amb èxit. Un extracte d'aquesta funció és mostra a continuació.

```
...
NSString *query = @"INSERT INTO molecule (inchi, moleculeName, formule, fileId)
    VALUES (?, ?, ?, ?)";
...

if(sqlite3_prepare_v2(database, sqlStatement, -1, &compiledStatement,
    NULL) == SQLITE_OK) {
    sqlite3_bind_text(compiledStatement, 1, [molecule.inchi UTF8String],
        -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(compiledStatement, 2, [molecule.moleculeName UTF8String],
        -1, SQLITE_TRANSIENT);
    sqlite3_bind_text(compiledStatement, 3, [molecule.formule UTF8String],
        -1, SQLITE_TRANSIENT);
}

```

```

        sqlite3_bind_text(compiledStatement, 4, [molecule.fileId UTF8String],
                        -1, SQLITE_TRANSIENT);
    }
    if(sqlite3_step(compiledStatement) == SQLITE_DONE) {
        sqlite3_finalize(compiledStatement);
        [self.molecules addObject:molecule];
    }
    ...

```

- *(void) dropMoleculeFromDatabase:(Molecule \*)molecule*

De manera anàloga a la inserció de molècules, el procés a seguir és tal i com s'ha mostrat en aquest cas. Només canvia el tipus de consulta a executar.

```

...
NSString *query = @"DELETE FROM molecule WHERE inchi=?";
...

```

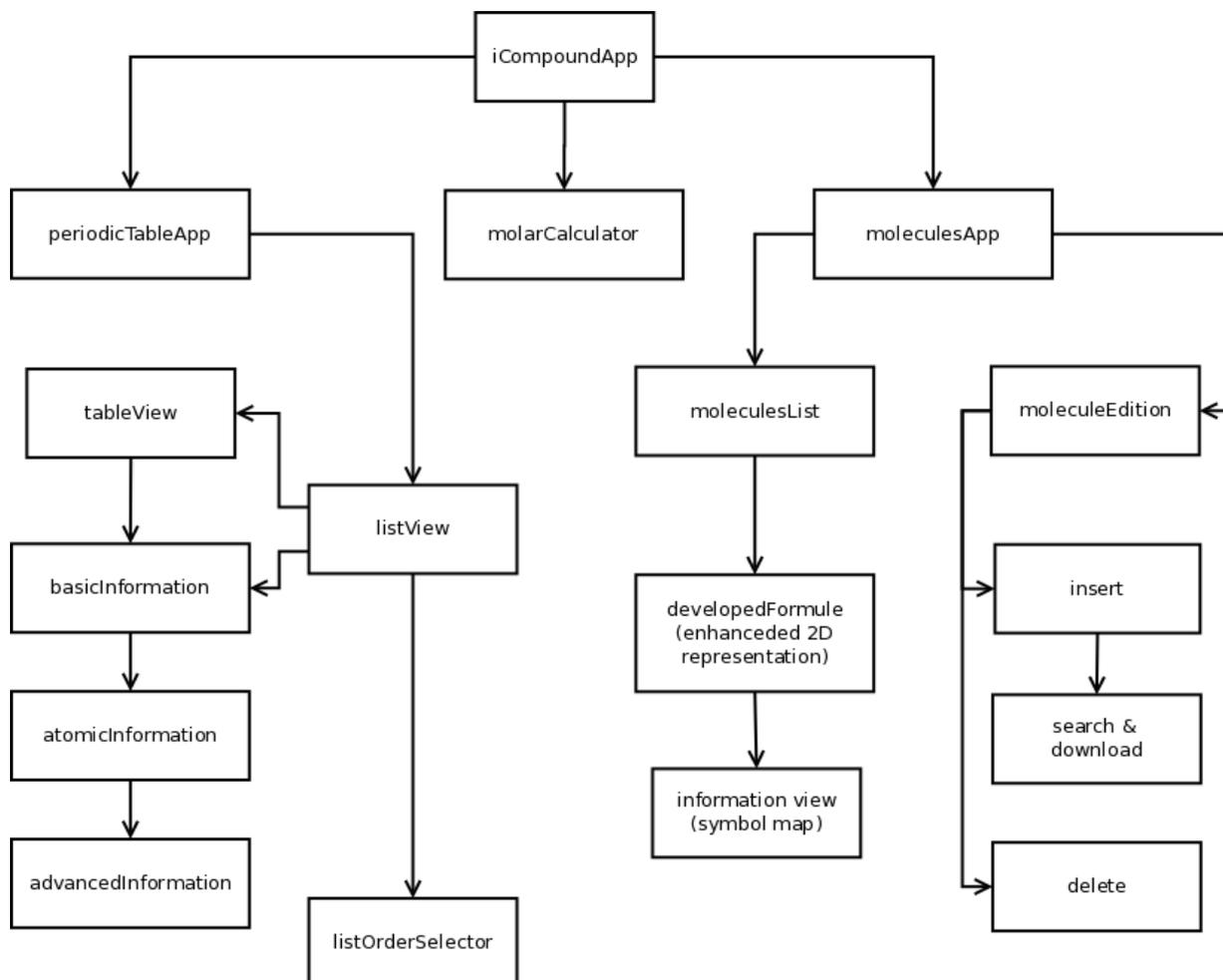
Tot el mostrat fins ara (en versió reduïda), explica com el subsistema Model pot fer la seva tasca principal, que no és més que garantir l'accés a les dades i oferir-les al subsistema de Control. Cal dir que aquestes funcionalitats queden recollides (principalment) en una classe delegada anomenada *iCompoundDelegate*. D'aquesta manera, i d'un mode estandarditzat, es pot fer l'aproximació al patró MVC de la manera recomanada i utilitzada per Apple en els frameworks d'iOS.

## 4.4 Implementació del subsistema View

Aquest és l'encarregat de mostrar tota la informació requerida per l'usuari i respondre a les decisions que prengui en cada moment. En el cas de *iCompound*, l'aplicació ha de tenir una interfície basada en diferents vistes accessibles de manera senzilla, desglossada bàsicament en tres grups: taula / llista d'elements, calculadora molar i visualitzador de molècules.

### 4.4.1 Variacions respecte el disseny teòric

Cal destacar que, tot i que s'ha intentat seguir el disseny teòric plantejat en anteriors apartats, en algun cas s'ha optat per una implementació lleugerament diferent. Les modificacions aplicades donen com a resultat un esquema de navegació lleugerament modificat respecte el plantejament inicial:



Imatge 34: Esquema de navegació per iCompound (modificat)

Les variacions són:

- La navegació entre les diferents propietats de l'element es fa a través de la barra superior. Resulta més intuïtiu i senzill que no haver de tornar cada vegada a una vista anterior per seleccionar el tipus de propietat desitjat. A més, es destaca la importància de cada tipus al tenir diferents profunditats d'accés.
- L'accés a la vista de taula periòdica es fa des de la vista del tipus llistat, que és la mostrada per defecte. Així doncs, les seleccions d'elements, del criteri d'ordre i d'accés al format taula queden integrats a la vista de llistat.
- La vista de taula (clàssica) no permet la possibilitat de tenir diferents formats. Es simplifica així el ventall de possibilitats i opcions, fet que podria despistar a l'usuari del que realment està fent. A més, simplifica també el desenvolupament (tot i que no es descarta incloure aquesta possibilitat en la versió final).

- Queda descartada la representació 3D en detriment d'una representació de dos dimensions basada en un format de fórmula desenvolupada millorada, tal i com ja s'ha explicat anteriorment.

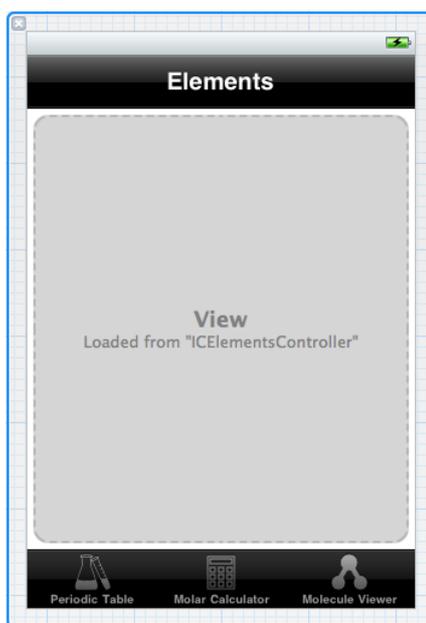
Aquest canvis simplifiquen la implementació, a més de millorar aspectes relacionats amb la usabilitat.

#### 4.4.2 Desenvolupament final

El desenvolupament de cada una de les vistes principals s'ha fet emprant *Interface Builder*, mentre que d'altres més secundàries s'han implementat via codi *Objective-C*. El resultat del desenvolupament permet obtenir les següents vistes, organitzades dins la carpeta *Resources* segons funcionalitats:

##### 1. *General*

- **MainWindow**. Vista principal, la qual conté la barra inferior amb pestanyes que garanteixen accés a les vistes associades a cada un dels mòduls de *iCompound*.



Imatge 35: Disseny en Interface Builder de la vista principal

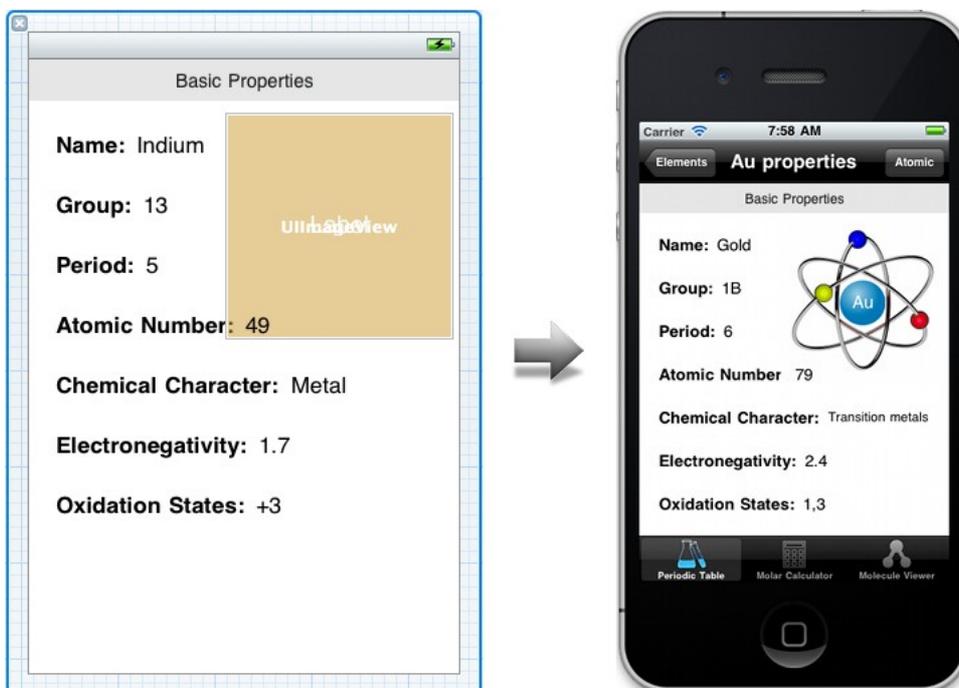
##### 2. *Taula periòdica dels Elements*

- **ICElementsView**. Vista contenidora d'una llista (taula) amb tots els elements, agrupats segons diferents criteris. Tant la taula com la navegació entre aquesta vista i les característiques d'element (accessibles des de la taula) es gestionen des de la barra de navegació, al igual que selecció d'ordre (botó '*Order Criteria*') o la vista de format taula periòdica (botó '*TableView*').



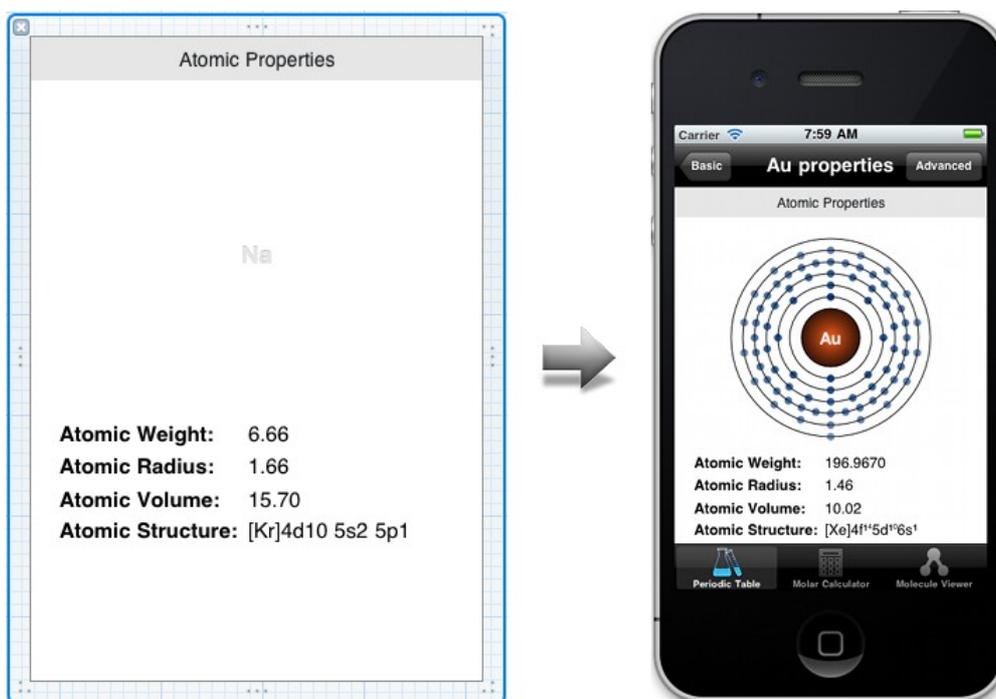
Imatge 36: ICElementsView. Esquerra, en Interface Builder, dreta en Simulador iOS.

- ICElementsBasicView. Vista que mostra les propietats més bàsiques de cada element.



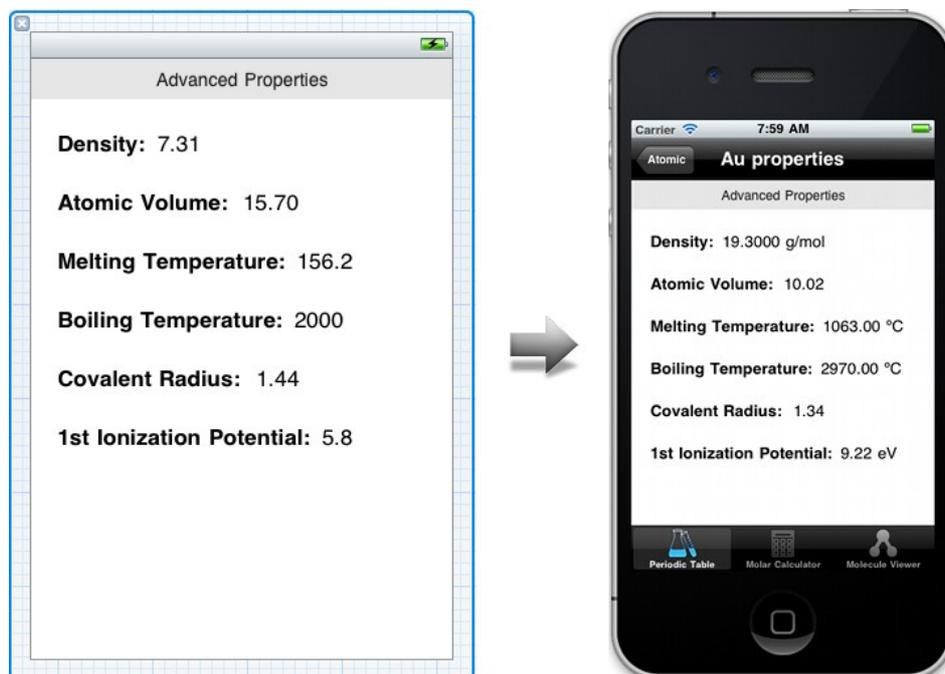
Imatge 37: ICElementsBasicView. Esquerra, en I.B. , dreta en Simulador iOS.

- **ICElementsAtomicView.** Vista que mostra la configuració atòmica de l'element en format gràfic, a més de certes propietats lligades a l'àtom.



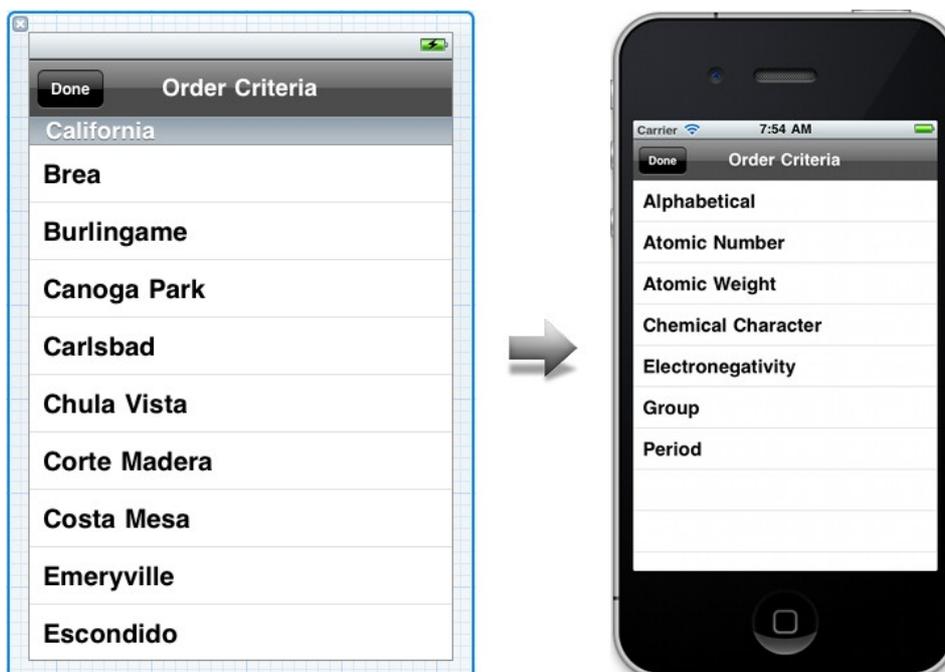
Imatge 38: ICElementsAtomicView. Esquerra, en I.B. , dreta en Simulador iOS.

- **AtomView.** Vista generada amb codi C-Objective, encarregada de mostrar la imatge de l'àtom. Aquesta es genera en temps real amb les dades de cada element, i per tant, alleuja l'aplicació d'inserir imatges estàtiques per cada cas. Destacar que aquesta es genera en temps real: per cada element seleccionat s'invoca en seleccionar les seves propietats atòmiques.
- **ICElementsAdvancedView.** Vista que mostra dades addicionals de l'element, propietats que no són tan rellevants però de gran utilitat per usuaris avançats.



Imatge 39: ICElementsAdvancedView. Esquerra, en I.B. , dreta en Simulador iOS.

- **ICElementsOrderCriteriaView.** Vista modal que mostra en una taula d'elements seleccionables les opcions d'ordenació a aplicar a ICElementsView. La gestió de l'ordenació és possible gràcies al protocol implementat al controlador d'aquesta vista.



Imatge 40: IOrderCriteriaView. Esquerra, en I.B. , dreta en simulador iOS.

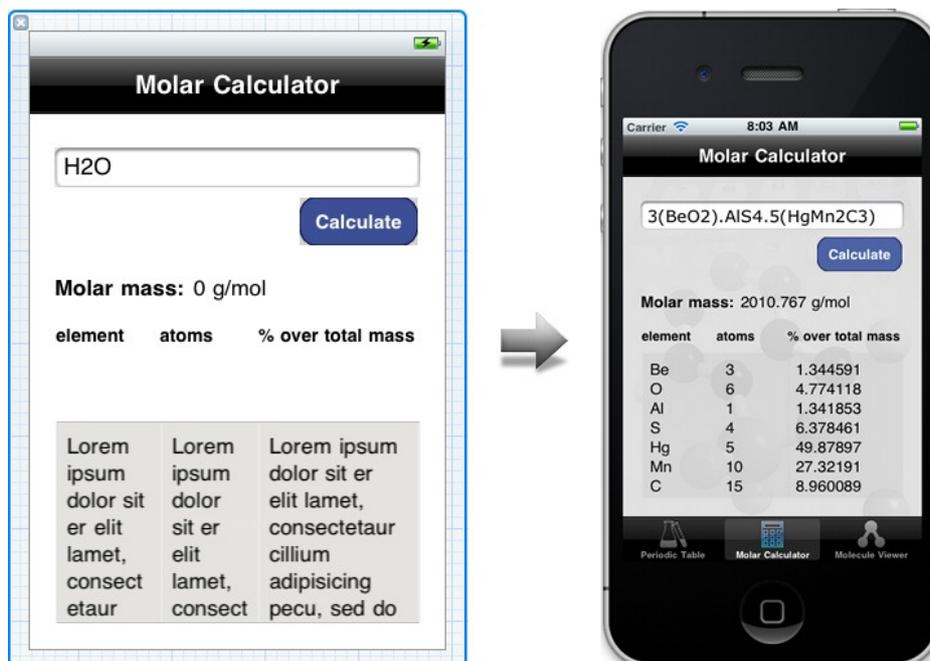
- **ICElementsTableView.** Vista que permet veure els elements disposats en el format de taula periòdica clàssic. Cada element es correspon a un botó que permet accedir, de manera anàloga a les cel·les de ICElementsView, a les propietats del mateix (a través del controlador de navegació abans implementat).



Imatge 41: ICElementsTableView, en simulador iOS.

### 3. Calculadora Molar

- **ICCalculatorView.** Vista que conté tot el necessari per obtenir la massa molar d'un compost (introduït a mà, via teclat tàctil). El resultat es mostra en dues parts: valor total, a sota del camp d'introducció de dades, i respecte a cada element (mitja part inferior). Les operacions corresponents a cada càlcul es realitzen al controlador pertinent (ICCalculatorController).



Imatge 42: ICCalculatorView. Esquerra, en I.B., dreta en Simulador iOS.

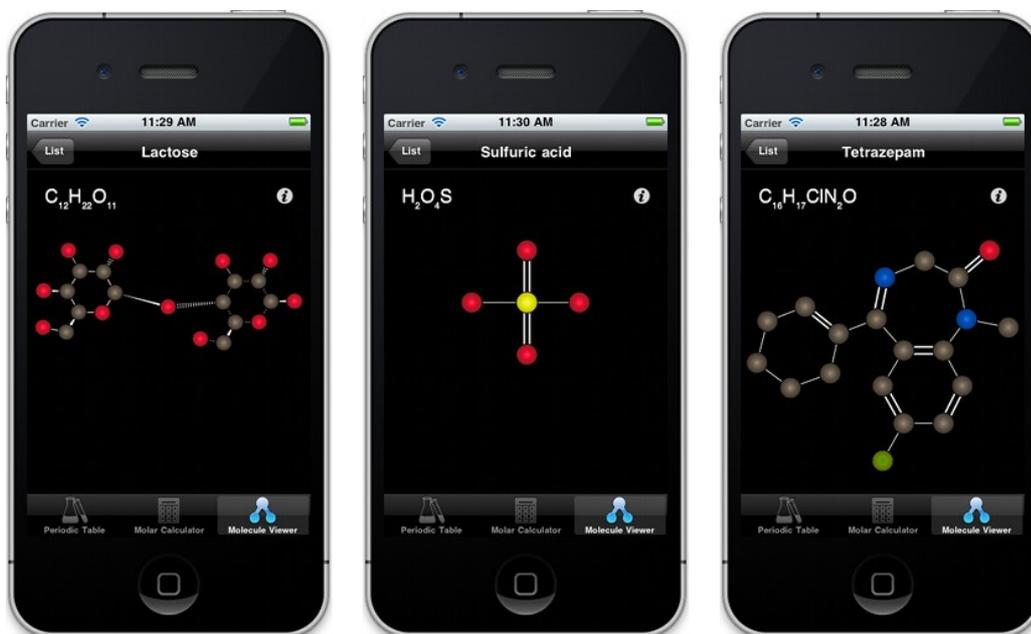
#### 4. Visor de Molècules

- ICMoleculesView.** Vista que permet veure un llistat ordenat alfabèticament de les molècules emmagatzemades (tant de les predefinides com les descarregades per l'usuari). Des d'ella es pot accedir a la visualització (clicant sobre una cel·la), o entrar en mode edició (botó 'Edit/Done' de la barra de navegació) per afegir molècules o esborrar-ne (sempre que siguin descarregades).



Imatge 43: ICMoleculesView. De dreta a esquerra: llista de molècules, mode edició amb inserció, mode edició amb supresió..

- ICSimpleMoleculeView.** Vista que conté la representació gràfica en dos dimensions (fòrmula desenvolupada millorada) de la molècula seleccionada. Conté una vista generada mitjançant codi de programació (simpleMolecule) que gestiona tota la lògica necessària per tal de dur a terme la representació gràfica requerida. De la mateixa manera que es fa amb l'àtom per la visat de propietats atòmiques dels elements, aquesta representació es fa en temps real.



Imatge 44: ICSimpleMoleculeView. Diferents representacions aconseguides..

- **ICMoleculeInfoView.** Vista modal que ofereix a l'usuari una llegenda de objectes i figures per entendre el model molecular representat per ICSimpleMoleculeController. La comunicació amb ella es fa amb el protocol propi *moleculeInfoDelegate*.



Imatge 45: ICMoleculeInfoView Esquerra, en I.B. , dreta en Simulador iOS.

- **ICMoleculesSearchView.** Vista modal que permet fer cerques de molècules *on-line* en la base de dades del *NIST*, fent ús d'un element *Search Bar*. Si la cerca dona resultats satisfactoris, és possible descarregar la molècula (botó '*Download*' de la barra inferior) per inserir-la a la base de dades d'*iCompound*, podent accedir a la seva representació quan el usuari ho desitgi. Altrament, es pot cancel·lar el procés amb el botó '*Cancel*' de la barra inferior.



Imatge 46: ICMoleculesSearchView Esquerra, en I.B. , dreta en Simulador iOS.

## 4.5 Implementació del subsistema Controller

Els controladors són classes encarregades d'oferir una sèrie de mètodes i propietats d'utilitat per interconnectar el subsistema Model amb el View; és a dir, s'encarrega de preparar les dades rescadades de la persistència per mostrar-les correctament a les vistes corresponents, i de realitzar les operacions necessàries per tal que les accions d'usuari sobre la informació tinguin la resposta adequada. Així doncs, es pot afirmar que és en aquest subsistema on es troba la lògica de negoci de l'aplicació.

En el cas de *iCompound*, existeixen diferents classes controladores, generalment lligades a les seves corresponents vistes (ja descrites en l'apartat anterior). A continuació es mostra cada una d'elles, remarcant els mètodes i propietats més importants en cada cas, així com la seva funcionalitat.

### 4.5.1 Mòdul d'elements

Tots els controladors d'aquest mòdul es troben sota la carpeta *classes/elements*. És en ells on es pot trobar tota la lògica de negoci que involucra a la interacció de l'usuari amb les dades dels elements de la taula periòdica en els diferents formats establerts.

#### 4.5.1.1 ICElementsController

Classe controladora encarregada de tractar les dades referents als elements i mostrar-les en la taula. Subclasse de *UIViewController*, permet la gestió de navegació entre les diferents vistes associades. A més, implementa els protocols per gestió dades i accions associades a taules *UITableViewDataSource* i *UITableViewDelegate*, i el protocol propi *orderCriteriaDelegate*, que garanteix la correcta visualització segons criteri d'ordenació triat.

Les principals propietats són:

- ***eDelegate***. Objecte del tipus *iCompoundDelegate* que permet accedir al delegat de l'aplicació per obtenir la informació requerida en cada moment.
- ***elements***. Objecte del tipus *NSMutableArray* que conté el llista d'elements, ordenats segons el criteri establert per l'usuari. D'utilitat per omplir la taula, així com per obtenir les referències necessàries per accedir a les propietats segons la selecció d'element feta.

Remarcar que hi han d'altres propietats d'utilitat a l'hora d'emmagatzemar dades sobre els grups en que es classifica cada cel·la de la taula (*sectionKeys*, *sectionContents*, *contents*) i per obtenir el criteri d'ordenació a aplicar (*criteria*).

Dintre dels mètodes, es poden trobar els més comuns en la gestió de taules (*cellForRowAtIndexPath*, *numberOfRowsInSection*, *didSelectRowAtIndexPath*, *titleForHeaderInSection*, *numberOfSectionsInTableView*). No cal entrar en detalls sobre aquests, ja que la seva funcionalitat es prou coneguda i està ben definida. Cal destacar, però, d'altres mètodes implementats:

- ***goToTableView***. Encarregat d'accedir a la vista de taula periòdica clàssica i gestionar el pas entre ella i la vista de taula, a més d'associar-hi la barra de navegació.

```
-(IBAction)goToTableView:(id)sender {
    UIBarButtonItem *backButton =
        [[UIBarButtonItem alloc]
         initWithTitle:@"List View"
         style:UIBarButtonItemStyleBordered target:nil action:nil];
    self.navigationItem.backBarButtonItem = backButton;
}
```

```
ICElementsTable *elementsTable =  
    [[ICElementsTable alloc] initWithNibName:@"ICElementsTable" bundle:nil];  
UINavigationController *navController = self.navigationController;  
[navController popViewControllerAnimated:NO];  
navController.hidesBottomBarWhenPushed = YES;  
[navController pushViewController:elementsTable animated:YES];  
}
```

- **goToOrderView.** Gestiona l'accés a la finestra modal de selecció de criteri d'ordenació. Per aquesta tasca fa ús del protocol orderCriteriaDelegate, que permet definir les opcions d'usuari.

```
-(IBAction)goToOrderView:(id)sender {  
    ICElementsOrderCriteria *varOrderCriteriaViewController =  
        [[ICElementsOrderCriteria alloc] init];  
    varOrderCriteriaViewController.OCDelegate = self;  
    [self.navigationController presentViewController:  
        varOrderCriteriaViewController animated:YES];  
}
```

- **didReceiveOrderOption.** Gestiona com la taula de la vista associada ha de mostrar-se, en funció del criteri d'ordenació. Mitjançant el protocol orderCriteriaDelegate rep la informació recollida en la vista ICElementsOrderCriteriaView, per comunicar-se amb el delegate de l'aplicació (iCompoundDelegate) i determinar la visualització a aplicar.

```
-(void)didReceiveOrderOption:(NSString *)selectedCriteria  
{  
    ICompoundDelegate *eDelegate =  
        (ICompoundDelegate *)[[UIApplication sharedApplication] delegate];  
  
    if (selectedCriteria != nil) {  
        [eDelegate didReceiveOrder:selectedCriteria];  
        elements = nil;  
        sectionKeys = eDelegate.sectionKeys;  
        sectionContents = eDelegate.sectionContents;  
        [self.elementsTableView reloadData];  
    }  
}
```

#### 4.5.1.2 ICElementsOrderCriteria

Classe encarregada de fer efectius els canvis de visualització de la llista d'elements en funció de diferents

paràmetres seleccionables per l'usuari. Destacar que defineix la vista associada (*ICElementsOrderCriteriaView*) com modal, al ser un punt on l'usuari ha de prendre una decisió que afecta a part del funcionament. Al igual que *ICElementsController*, implementa els protocols *UITableViewDataSource* i *UITableViewDelegate*.

Les propietats principals són:

- ***eDelegate***.
- ***OCDelegate***. Permet, a través del protocol *orderCriteriaDelegate*, accedir els mètodes associats al mateix per determinar l'ordre de visualització dels elements a la vista *ICElementsView*.
- ***ElementsOrderTableView***. Subclasse d'*UITableView*, corresponent a la taula amb les opcions d'ordenació preestablertes.

Els mètodes que encapsulen les funcionalitats primordials d'aquest controlador són, al igual que en *ICElementsController* els propis per gestionar la taula que conté, les dades associades i els comportaments. Destacar que la taula i el seu contingut es genera via codi Objective-C. Cal destacar el mètode:

- ***dismissView***. Encarregat de retornar el mètode d'ordenació a la vista principal *ICElementsView*, fent servir el mètode *didReceiveOrderOption* del protocol *orderCriteriaDelegate*. A més, s'encarrega de tancar la vista modal associada, *ICElementsOrderCriteriaView*.

```
- (IBAction)dismissView:(id)sender {
    [OCDelegate didReceiveOrderOption:rowTitle];
    [self dismissModalViewControllerAnimated:YES];
}
```

#### 4.5.1.3 *ICElementsTable*

Classe controladora (subclasse de *UIViewController*) que gestiona el comportament de la taula periòdica (format clàssic) per mostrar les dades requerides per la vista associada (*ICElementsTableView*), així com el comportament davant les accions dutes a terme per l'usuari. Permet accedir a les propietats de l'element seleccionat o tornar al format de visualització de taula, com a funcionalitats més rellevants.

Les principals propietats són:

- ***eDelegate***.
- ***elementsDictionary***. Aquest objecte, del tipus *NSMutableDictionary* permet associar una

referència de cada element al botó en la vista ICElementsTableView.

Quant als mètodes, només destacar un:

- **selectedElement.** Implementa la lògica necessària per tal que la selecció feta per l'usuari en la vista associada ICElementsTableView (al prémer un dels botons corresponents a cada element), permeti obtenir les dades associades al mateix i garantir la navegació entre les seves propietats (bàsiques, vista ICElementsBasicView).

```
- (IBAction)selectedElement:(id)sender {
    ICompoundDelegate *delegate =
        (ICompoundDelegate *)[[UIApplication sharedApplication] delegate];
    ICElementsBasicController *basicInformation = [[ICElementsBasicController alloc]
                                                    init];
    UIButton *elementButton = (UIButton *)sender;
    basicInformation.element =
        [elementsDictionary objectForKey:[elementButton currentTitle]];
    [delegate.INavController pushViewController:basicInformation animated:YES];
    [basicInformation release];
}
```

#### 4.5.1.4 ICElementsBasicController

Controlador (subclasse de UIViewController) que implementa la lògica necessària per que la informació bàsica referent a l'element sigui mostrada en el format correcte en la vista associada ICElementsBasicView. Tanmateix, encapsula les funcionalitats de navegació entre vistes, necessària per tornar al llistat general (ICElementsView) o anar a les propietats atòmiques (ICElementsAtomicView), pas gestionat pel mètode *viewAtomicProperties*. Destacar d'aquest últim que fa ús de la propietat *element* (objecte de la classe bàsica *Element*) per passar informació sobre el mateix.

```
-(IBAction) viewAtomicProperties : (id) sender {
    ICElementsAtomicController *varAtomicViewController =
        [ [ICElementsAtomicController alloc] initWithIndexPath:index];
    varAtomicViewController.element = element;
    [self.navigationController pushViewController:varAtomicViewController
        animated:YES];
    [varAtomicViewController release];
}
```

#### 4.5.1.5 ICElementsAtomicController

Controlador (subclasse de `UIViewController`) que implementa la lògica necessària per que la informació atòmica referent a l'element sigui mostrada en el format correcte en la vista associada `ICElementsAtomicView`. Tanmateix, encapsula les funcionalitats de navegació entre vistes, necessària per tornar a la vista de propietats bàsiques (`ICElementsBasicView`), o cap a la de propietats avançades (`ICElementsAdvancedView`). De manera anàloga a l'anterior controlador descrit, es fa ús d'un mètode que gestiona l'accés a les propietats avançades a petició de l'usuari, sota el nom de `viewAdvancedProperties`.

La propietat `atomView` instancia un objecte de la classe `AtomView` (subclasse de `UIView`), permetent que la vista associada a `ICElementsAtomicView` quedi lligada a aquesta, fet que possibilita la creació del diagrama de l'àtom en temps real.

#### 4.5.1.6 ICAtomViewController

Classe de control que implementa la lògica necessària per generar la vista associada a `ICElementsAtomicView`. Subclasse de `UIView`, la seva finalitat és generar en temps real la vista de l'àtom de l'element en funció de les seves característiques.

Per poder dur a terme aquesta tasca utilitza les funcionalitats de representació gràfica en 2D proporcionada pel framework Core Graphics i Quartz2D. L'ús d'aquests queda palès en els mètodes:

- **`drawAtomCore`**. Mètode encarregat de representar el nucli de l'àtom en el lloc pertinent i amb les propietats especificades.

```
-(void)drawAtomCore:(CGContextRef)context
{
    CGGradientRef myGradient;
    CGColorSpaceRef myColorspace;
    size_t num_locations = 3;
    CGContextSetFlatness(context,1);

    CGFloat locations[4] = { 0, 0.5, 1.0 };
    CGFloat components[12] = {
        0.968, 0.368, 0.160, 1.0,
        0.553, 0.209, 0.090, 1.0,
        0.200, 0.078, 0.035, 1.0,
    };

    myColorspace = CGColorSpaceCreateDeviceRGB();
    myGradient = CGGradientCreateWithColorComponents (myColorspace, components,
```

```

        locations, num_locations);
CGContextDrawRadialGradient (context, myGradient, myStartPoint, myStartRadius,
        myEndPoint, myEndRadius, 0);
CGContextBeginPath(context);
CGContextAddArc(context, myStartPoint.x, myStartPoint.y, myEndRadius,
        0, 2*M_PI, 0);
CGContextStrokePath(context);
}

```

- **drawAtomOrbits.** Mètode encarregat de representar les diferents òrbites amb els corresponents electrons, segons les especificacions donades i una sèrie de regles químiques bàsiques (que s'escapen de l'objectiu principal d'aquest informe).

```

-(void)drawAtomOrbits:(CGContextRef)context
{
    int n,m;
    int numberOfElectrons;
    int coreAtomRadius = 30;
    CGPoint currentPoint;

    for( n=1; n <= [electronsPerLevel count]; n++ ) {
        numberOfElectrons = [[electronsPerLevel objectAtIndex:n-1] intValue];
        for( m=0; m<numberOfElectrons; m++ ) {
            CGContextBeginPath(context);
            CGContextAddArc(context, myStartPoint.x, myStartPoint.y,
                coreAtomRadius + 12*n, 2*m*M_PI/numberOfElectrons+M_PI/2,
                2*(m+1)*M_PI/numberOfElectrons+M_PI/2, 0);
            currentPoint = CGContextGetPathCurrentPoint(context);
            CGContextStrokePath(context);
            CGFloat cs2[] = {0.0, 0.196, 0.455, 1.0-(0.08*n)};
            CGColorSpaceRef myCs = CGColorSpaceCreateDeviceRGB();
            CGColorRef color2 = CGColorCreate(myCs, cs2);
            CGContextSetFillColorWithColor(context, color2);
            CGContextBeginPath(context);
            CGContextAddArc(context, currentPoint.x, currentPoint.y,
                4, 0, 2*M_PI, 0);
            CGContextFillPath(context);
        }
    }
}

```

- **drawRect.** Prepara el context per poder representar l'àtom, cridant després de certes inicialitzacions als mètodes anteriorment citats.

```

- (void)drawRect:(CGRect)rect
{
    [super drawRect:rect];
    CGContextRef ctx = UIGraphicsGetCurrentContext();

    CGFloat cs[] = {1.0, 1.0, 1.0, 1.0};
    CGColorSpaceRef myCs = CGColorSpaceCreateDeviceRGB();
    CGColorRef color = CGColorCreate(myCs, cs);
    CGContextSetFillColorWithColor(ctx, color);
    CGContextBeginPath(ctx);
    CGContextMoveToPoint(ctx, 0, 0);
    CGContextAddLineToPoint(ctx, CGRectGetMaxX(self.bounds), 0);
    CGContextAddLineToPoint(ctx, CGRectGetMaxX(self.bounds),
        CGRectGetMaxY(self.bounds));
    CGContextAddLineToPoint(ctx, 0, CGRectGetMaxY(self.bounds));
    CGContextClosePath(ctx);
    CGContextFillPath(ctx);
    myStartPoint.x = CGRectGetMaxX(self.bounds)/2;
    myStartPoint.y = 155;
    myEndPoint.x = myStartPoint.x;
    myEndPoint.y = myStartPoint.y;
    myStartRadius = 0;
    myEndRadius = 30;

    [self drawAtomCore: ctx];
    [self drawAtomOrbits :ctx];
}

```

Com a principal propietat, esmentar *electronsPerLevel*, un llistat del tipus `NSMutableArray` que conté la informació necessària per poder dur a terme les operacions del mètode *drawAtomOrbits*.

#### 4.5.1.7 ICElementsAdvancedController

Controlador (subclasse de `UIViewController`) que implementa la lògica necessària per que la informació avançada referent a l'element es mostri en el format correcte en *ICElementsAdvancedView*. Tanmateix, encapsula les funcionalitats de navegació entre vistes, necessària per tornar a la vista prèvia (*ICElementsAtomicView*).

## 4.5.2 Mòdul de Calculadora Molar

Tots els controladors d'aquest mòdul es troben sota la carpeta *classes/calculator*. Encapsulen les funcionalitats inherents al càlcul de massa molar del compostos introduïts per l'usuari.

### 4.5.2.1 ICCalculatorController

Únic controlador del mòdul, que gestiona l'enllaç entre les dades extretes de la persistència i la vista associada *ICCalculatorView*. Per poder dur a terme les tasques descrites anteriorment, fa ús d'un seguit de propietats de les quals cal destacar:

- ***elements***. Diccionari on es poden consultar les dades necessàries referents al pes atòmic de cada un dels elements implicats en el càlcul.
- ***elementsDetected***. Diccionari on s'emmagatzemen temporalment tots i cada un dels elements implicats en el càlcul. Proporciona informació d'utilitat per mostrar les estadístiques parcials.
- ***elementsDetectedIndex***. Array utilitzat en la descodificació de *elementsDetected*.

Pel que fa a mètodes representatius, s'han de tenir en compte:

- ***performOperation***. Conté tota la lògica de negoci necessària per poder realitzar els càlculs de massa molar a partir dels elements, de la seva repetibilitat i de la seva massa. El procés de càlcul, així com el format de les expressions a tractar responen a patrons definits en química, i que s'escapen de l'abast d'aquest informe. Les dades necessàries es capturen via *ICCalculatorView*, amb la interacció de l'usuari. En aquest cas, i per complexitat del codi, no s'adjunta una part del mateix; no obstant, es pot consultar als annexos.
- ***showResults***. Processa, en els formats especificats, la informació resultant de realitzar les operacions pertinents sobre la informació proporcionada per l'usuari, per poder mostrar-la en *ICCalculatorView*. Aquesta quedarà desglossada en dos blocs: el del resultat general i els parcials estadístics.

```
-(void)showResults:(float)result {
    elementListView.text=@"";
    atomNumberView.text=@"";
    weigthPercentualView.text=@"";
    for(int i=0; i < [elementsDetectedIndex count]; i++) {
        elementListView.text = [[elementListView.text stringByAppendingString:
            [elementsDetectedIndex objectAtIndex:i]] stringByAppendingString:@"\n"];
        atomNumberView.text = [[atomNumberView.text stringByAppendingString:
```

```

        [[elementsDetected objectForKey:[elementsDetectedIndex objectAtIndex:i]]
         stringValue]] stringByAppendingString:@"\n"];
    float percentualWeight = ([[elementsDetected objectForKey:[elementsDetectedIndex
        objectAtIndex:i]] intValue]*[[[elements objectForKey:
            [elementsDetectedIndex objectAtIndex:i]] eAtom] atomicWeight]
            floatValue])*100/result;
    weighPercentualView.text = [[weighPercentualView.text stringByAppendingString:
        [[NSNumber numberWithFloat:percentualWeight] stringValue]]
        stringByAppendingString:@"\n"];
}
resultLabel.text = [[[NSNumber numberWithFloat:result] stringValue]
    stringByAppendingString:@" g/mol"];
}

```

- **dismissKeyboard.** Mètode que gestiona la informació introduïda via teclat tàctil en ICCalculatorView, passant-la als mètodes anteriorment descrits, processant així la informació i tornant-la degudament formatada a la vista associada.

```

-(IBAction)dismissKeyboard: (id)sender {
    [formulaField resignFirstResponder];
    [self showResults:[self performOperation]];
}

```

### 4.5.3 Mòdul de Molècules

Tots els controladors d'aquest mòdul es troben sota la carpeta *classes/molecules*. S'encarreguen de definir la lògica de negoci lligada a les diferents representacions de compostos que l'usuari pot requerir.

#### 4.5.3.1 ICMoleculesController

Controlador (subclasse de UIViewController) que implementa la lògica necessària per poder gestionar (navegar, editar) la taula de les molècules disponibles continguda en *ICMoleculesView*. A més, implementa els protocols per gestió dades i accions associades a taules UITableViewDataSource i UITableViewDelegate.

Les principals propietats són:

- **eDelegate.** Objecte del tipus iCompoundDelegate que permet accedir al delegat de l'aplicació per obtenir, modificar i/o emmagatzemar la informació requerida en cada moment.
- **molecules.** Objecte del tipus NSMutableArray que conté el llista de les molècules. D'utilitat per omplir la taula de molècules disponibles, per obtenir les referències necessàries i accedir a

les visualitzacions gràfiques de la molècula seleccionada. També permet iniciar el procés d'emmagatzematge de noves, cas que sigui necessari.

Respecte als mètodes, cal remarcar alguns dels ja esmentats que permeten la gestió de la taula, que en el cas que ens ocupa queden lleugerament modificats per tal de poder actuar davant les opcions d'inserció/esborrat de noves molècules.

- ***cellForRowAtIndexPath.*** Encarregat de determinar la informació a la cel·la assenyalada per *indexPath*. Discrimina entre el que és una cel·la contenidora d'una molècula del que és la designada per accedir a la inserció de noves (col·locada al final de la taula).

```
- (UITableViewCell *)tableView:(UITableView *)tv cellForRowAtIndexPath:
    (NSIndexPath *)indexPath {
    UITableViewCell *cell = [tv dequeueReusableCellWithIdentifier:@"cell"];
    if(nil == cell) {
        cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleValue1
            reuseIdentifier:@"cell"] autorelease];
    }
    if(indexPath.row < molecules.count) { // Cell contains an existing molecule
        Molecule *thisMolecule = [molecules objectAtIndex:indexPath.row];
        cell.textLabel.text = thisMolecule.moleculeName;
        cell.textLabel.textColor = [UIColor blackColor];
        cell.editingAccessoryType = UITableViewCellAccessoryNone;
        if(self.editing) {
            [cell setSelectedStyle:UITableViewCellStyleNone];
        }
        cell.detailTextLabel.text = thisMolecule.formule;
        cell.detailTextLabel.font = [UIFont systemFontOfSize:14];
    }
    else { // Cell for add new molecule (temporally created for that task)
        cell.textLabel.text = @"Add new molecule";
        cell.textLabel.textColor = [UIColor blueColor];
        cell.editingAccessoryType = UITableViewCellAccessoryDisclosureIndicator;
        cell.detailTextLabel.text = @"";
    }
    return cell;
}
```

- ***didSelectRowAtIndexPath.*** Encarregat de determinar l'acció a prendre al seleccionar una de les cel·les de la taula. Si s'està en mode d'edició es garanteix l'accés a la vista de cerca i inserció; en cas contrari, s'accedeix a la visualització en 2D de la molècula seleccionada.

```
- (void)tableView:(UITableView *)tv didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
```

```

if(indexPath.row < molecules.count && !self.editing) {
    // Is the selected row an existing molecule? Then go to view its representation
    ICSimpleMoleculeController *moleculeController = [[ICSimpleMoleculeController
        alloc] initWithIndexPath:indexPath];
    moleculeController.selectedMolecule = [molecules objectAtIndex:indexPath.row];
    [eDelegate.ICMoleculesNavController pushViewController:moleculeController
        animated:YES];
    [moleculeController release];
}
if(indexPath.row == molecules.count && self.editing) {
    // If user wants to add a new molecule, the let it go to molecule finder
    ICMoleculesSearchController *searchController = [[ICMoleculesSearchController
        alloc] init];
    [eDelegate.ICMoleculesNavController presentViewController:
        searchController animated:YES];
    [searchController release];
}
[tableView deselectRowAtIndexPath:indexPath animated:YES];
}

```

- **setEditing.** Encarregat de gestionar l'entrada/sortida de la taula en mode edició.
- **editingStyleForRowAtIndexPath.** Encarregat de determinar l'estil de la cel·la especificada en el mode edició. Aquestes poden ser de dos tipus: afegir un nou ítem o esborrar-lo, i la seva elecció marca les accions a prendre.
- **commitEditingStyle.** Encarregat de establir les accions associades a cada estil dintre del mode edició. Centrat en l'esborrat de molècules (la inserció es gestiona en d'altres mètodes tal i com s'ha especificat), verifica tant l'existència d'aquesta a la base de dades de l'aplicació com del fitxer d'extensió .mol corresponent a les dades de representació gràfica. Si tot és correcte, l'esborrat es fa efectiu (eliminació de la cel·la, de la BBDD i de fitxers associats); altrament, informa de l'error en el procés.

```

// Tasks to do on edit mode. More exactly, deleting cells tasks!!!
-(void) tableView:(UITableView *)tv commitEditingStyle:
    (UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:
    (NSIndexPath *)indexPath {
    if (editingStyle == UITableViewCellEditingStyleDelete) {
        Molecule *molecule = [molecules objectAtIndex:indexPath.row];
        // For error information
        NSError *error;
        // Create file manager
        NSFileManager *fileMgr = [NSFileManager defaultManager];
        NSArray *paths =

```

```

NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
    NSUserDomainMask, YES);
NSString *documentsPath = [paths objectAtIndex:0];
NSString *bundlePath = [[NSBundle mainBundle] pathForResource:[molecule
    fileId] ofType:@"mol"];
NSString *fh = [NSString stringWithContentsOfFile:bundlePath encoding:
    NSUTF8StringEncoding error:nil];
// Molecules were associated to representation files on .mol format.
// If exists on the bundle path of application, user can't delete it.
// If exists on user documents path, they're downloaded molecules, and
// its deletion is permitted
if(fh==nil) {
    // Delete selected cell
    fh = [NSString stringWithContentsOfFile: [[[documentsPath
        stringByAppendingString:@"/"] stringByAppendingString:
        [molecule fileId] stringByAppendingString:@"mol"]
        encoding:NSUTF8StringEncoding error:nil];
    if ([fileMgr removeItemAtPath:[[[documentsPath
        stringByAppendingString:@"/"] stringByAppendingString:
        [molecule fileId] stringByAppendingString:@"mol"]
        error:&error] != YES) {
        }
    else {
        // and molecule and associated file if deletion is possible
        [eDelegate dropMoleculeFromDatabase:molecule];
        [molecules removeObjectAtIndex:indexPath.row];
        [tv deleteRowsAtIndexPaths:[NSArray arrayWithObject:
            indexPath] withRowAnimation:
            UITableViewRowAnimationLeft];
    }
} else {
    UIAlertView *alert =[[UIAlertView alloc] initWithTitle:
        @"Error deleting molecule"
        message: @"Predefined molecules can't be removed."
        delegate: self
        cancelButtonTitle: @"OK"
        otherButtonTitles: nil];
    [alert show];
    [alert release];
    return;
}
}
}
}

```

### 4.5.3.2 *ICSimpleMoleculeController*

Controlador (subclasse de `UIViewController`) que implementa la lògica necessària per poder mostrar la representació gràfica en 2D de la molècula seleccionada. Implementa el protocol *moleculeInfoDelegate*, que permet la comunicació amb la vista d'informació de la representació *ICMoleculeInfoController*.

Com a principals propietats destacar *molecule* (instància de la classe *simpleMolecule*, generadora de la representació) i *selectedMolecule* (del tipus *simpleMolecule*, que proporciona informació sobre la molècula a visualitzar).

Quant a mètodes, el més destacable és *goToInformationView*, que s'encarrega de garantir l'accés a la informació relativa a la representació molecular (llegenda de símbols).

```
-(IBAction)goToInformationView:(id)sender {
    ICMoleculeInfoController *varInfoViewController =
    [[ICMoleculeInfoController alloc] init];
    varInfoViewController.MIDelegate = self;
    [self.navigationController presentViewController:varInfoViewController
     animated:YES];
}
```

### 4.5.3.3 *simpleMolecule*

Subclasse de `UIView`, generada directament amb codi Objective-C, que conté tota la lògica necessària per tal de generar les representacions en dues dimensions de les molècules dins la vista *ICSimpleMoleculeView*. Aquesta representació es fa utilitzant funcions de de l'api *Quartz 2D*, inclosa en el framework *Core Graphics*.

Algunes de les propietats destacades són comunes a d'altres vistes ja analitzades (*eDelegate*, *elements*, etc), i la resta són específiques per poder realitzar tots els càlculs necessaris per poder situar i dibuixar en pantalla cada una de les molècules i els corresponents enllaços.

Quant als mètodes (dels quals es pot trobar informació detallada als annexos degut a la seva complexitat) es troben:

- ***drawRect***. Mètode principal, encarregat d'inicialitzar tant l'entorn gràfic com les propietats necessàries per aquests efectes. També fa la crida als mètodes implicats en la preparació i representació de les dades de la molècula seleccionada. A continuació es mostra una part del codi, encarregat de preparar la pantalla i cridar els mètodes implicats en el procés.

```

...
// Init graphics & draw initial (base) screen
CGContextRef ctx = UIGraphicsGetCurrentContext();
CGFloat cs[] = {0.0, 0.0, 0.0, 1.0};
CGColorSpaceRef myCs = CGColorSpaceCreateDeviceRGB();
CGColorRef color = CGColorCreate(myCs, cs);
CGContextSetFillColorWithColor(ctx, color);
CGContextBeginPath(ctx);
CGContextMoveToPoint(ctx, 0, 0);
CGContextAddLineToPoint(ctx, CGRectGetMaxX(self.bounds), 0);
CGContextAddLineToPoint(ctx, CGRectGetMaxX(self.bounds),
    CGRectGetMaxY(self.bounds));
CGContextAddLineToPoint(ctx, 0, CGRectGetMaxY(self.bounds));
CGContextClosePath(ctx);
CGContextFillPath(ctx);
...
// Get data to represent the molecule
[self readMolFile ];
// Draw molecule
[self drawMolecule: ctx];
...

```

- **readMolFile.** Encarregat d'obtenir la informació gràfica emmagatzemada en el fitxer .mol corresponent a la molècula que es vol representar. Extretes les dades, prepara les variables implicades en la representació, a més de realitzar els ajustos pertinents per garantir la correcta ubicació i grandària de la molècula, obtenint una perfecta adaptació a la pantalla del dispositiu.
- **drawMolecule.** Processa tota la informació sobre la representació per tal que cada àtom i enllaç es mostri a la posició que toca, amb les característiques especificades en cada cas.

#### 4.5.3.4 ICMoleculeInfoController

Controlador (subclasse de UIViewController) que s'encarrega de mostrar per pantalla, gràcies a la vista associada *ICMoleculeInfoView* informació relativa a les entitats mostrades en la representació gràfica de la molècula (el que es coneix com a llegenda o mapa de símbols). Per facilitar l'experiència i comoditat d'usuari fa ús de un element *UIScrollView*, que permet desplaçar-se per la vista amb controls tàctils.

Destacable és la utilització del protocol *moleculeInfoDelegate*, que agilitza la comunicació de la vista amb d'altres relacionades (tal i com s'ha apuntat abans).

### 4.5.3.5 *ICMoleculesSearchController*

Controlador (subclasse de *UIViewController*) que implementa la lògica necessària per realitzar la cerca i descàrrega d'una nova molècula, fent ús de la vista *ICMoleculesSearchView*. Tanmateix, implementa els protocols per gestió dades i accions associades a taules *UITableViewDataSource* i *UITableViewDelegate*, així com els necessaris per gestionar navegació web *UIWebViewDelegate* i cerques amb objectes gràfics *UISearchBarDelegate*.

Pel que es refereix a les propietats, dir que la gran majoria són específiques per poder gestionar la connexió i cerca en Internet, gestió de la taula de resultats o de la barra de cerca, a més de la ja esmentada *eDelegate*, d'utilitat per dur a terme les tasques relacionades amb la interacció amb la base de dades de l'aplicació.

Quant a mètodes, a banda dels típics necessaris per gestionar taules (amb seccions), val la pena destacar:

- ***searchBarTextDidBeginEditing***. Prepara l'entorn per poder introduir al cercador el nom de la molècula a descarregar. Mostra el teclat alfanumèric per permetre a l'usuari interactuar, a més de configurar els botons necessaris per dur a terme les tasques relacionades (cercar, esborrar, cancel·lar, etc).
- ***searchBarCancelButtonClicked***. Controla la selecció de l'usuari de cancel·lar la cerca activa, amagant el teclat i tornant a la vista de descàrrega.
- ***searchBarSearchButtonClicked***. Detecta l'ordre d'usuari de realitzar la cerca segons les dades especificades. Intenta connectar amb la base de dades del NIST a través d'Internet, i intenta recuperar la informació relacionada amb la molècula demanada. Si aquesta es troba, es prepara tot per tal que l'usuari pugui descarregar-la al seu dispositiu. Altrament informa a l'usuari de l'impossibilitat d'obtenir resultats per la cerca feta.
- ***moleculeDownload***. S'encarrega de gestionar tot el procés de descàrrega de la molècula cercada. Això implica la descàrrega del fitxer de dades de representació (format .mol) al dispositiu, a més de la inserció de les dades associades a la base de *iCompound*.

Cal destacar que el fet que una molècula estigui disponible per la descàrrega no implica que aquesta es dugui a terme. De fet, només és pot afegir sempre i quan no existeixi prèviament en l'aplicació (fet que es verifica, tant en el cas que es tracti d'una molècula predefinida com d'una descarregada per l'usuari). D'aquesta manera s'eviten tant duplicitats com ocupació d'espai innecessaris.

El codi font detallat de les funcions descrites no es mostra per la seva complexitat, però és possible veure'l a l'apartat d'annexos.

## 5 Integració i proves

Un cop es té el producte dissenyat i implementat, és el moment de verificar el correcte funcionament, la adequació als estàndards d'usabilitats establerts, i realitzar les accions necessàries per garantir que la qualitat de l'aplicació s'ajusta a les expectatives inicials.

Per això s'ha treballat en diferents fases:

1. Verificació de funcionament.
2. Avaluació de l'aplicació.
3. Aplicació de mesures correctores en funció dels resultats de l'avaluació.

El procés hauria de ser iteratiu, per tal de poder garantir una millora més acurada. Desafortunadament, el temps disponible per a la realització de tot el procés és molt limitat, i per tant, només es pot fer una vegada, i prenent accions molt específiques. Així doncs, i per simplificar, cada verificació aplica immediatament la correcció oportuna, per evitar arrossegar errors en el procés.

### 5.1 Verificació de funcionament

El primer punt en el procés d'integració i proves consisteix en garantir que les especificacions de funcionament fetes en estadis inicials del disseny es compleixen. Això implica revisar, punt per punt, cada part de l'aplicació en busca d'errades per poder fer una correcció acurada.

Aquesta verificació s'ha dut a terme en dos fases. La primera, sobre el simulador que incorpora X-Code, i altre, sobre un dispositiu iPod Touch (4<sup>a</sup> generació). Aquest procés s'ha focalitzat en la cerca de funcionaments anòmals (gestió ineficient de memòria, errors en el codi, etc) i incompliments de les especificacions inicials.

#### 5.1.1 Consideracions prèvies

És important destacar el fet que durant el procés d'implementació ja s'ha tingut en compte la detecció de comportaments que no s'ajusten a l'especificació, i per tant, molts dels possibles problemes ja han estat corregits.

Un dels principals punts delicats és la gestió de la memòria que els sistemes mòbils basats en iOS realitzen, tal i com s'ha apuntat en l'apartat 5.1.2. Durant el procés de verificació s'ha revisat que el cicle de creació i alliberament quedi balancejat, garantint en la mesura del possible una gestió encertada.

### 5.1.2 Detecció de problemes sobre simulador

No es van detectar problemes de memòria en la versió considerada final, però en canvi si certs comportaments que no es corresponien amb les especificacions donades:

- Per certes molècules, la seva representació no s'ajustava bé al tamany de la pantalla, fet que s'ha corregit millorant l'algorisme d'adaptació del renderitzat.
- La vista de propietats atòmiques no s'ajustava bé per àtoms d'elements amb tots els orbitals complets. Corregit adaptant la vista (posició de cada element).
- La calculadora molar no reaccionava bé a fórmules definides amb parèntesis, ja que no es respectava la precedència d'operands en casos puntuals. Solucionat ajustant l'algorisme de càlcul (als annexos es pot trobar les bases necessàries emprades per dur a terme aquestes tasques).
- L'esborrat de molècules no quedava restringit a les afegides via descàrrega. Una correcció en l'algorisme d'esborrat va eliminar el problema.

### 5.1.3 Detecció de problemes sobre iPod Touch

L'execució de *iCompound* sobre un dispositiu Apple és possible gràcies a les eines de depuració de X-Code, concretament a *Organizer* (que facilita de manera notable la integració d'aplicacions en fase de test en dispositius reals). D'aquesta manera, és possible poder verificar certs aspectes que es poden passar per alt al treballar sobre el simulador, a més de poder donar l'oportunitat a un/s usuari/s de fer servir l'eina directament en un dispositiu físic.

Així doncs, la primera verificació a fer era veure si hi havien problemes amb la gestió de la memòria. Es van trobar dues anomalies al respecte:

- L'aplicació s'avortava sobtadament quan es feien canvis en l'ordenació del llistat d'elements i posteriorment es realitzaven operacions de navegació sobre les propietats. El problema radicava en fer més d'un alliberament d'objectes de manera incorrecta, fet que desreferenciava part de la informació entre vistes. La solució va passar per fer una correcció en el cicle de creació i alliberament a la vista `ICElementsBasicView`.
- L'aplicació quedava penjada en intentar eliminar una molècula del llista de disponibles. El problema, més que de cicle de gestió de memòria, radicava en un ordre incorrecte del mateix. Organitzant-lo bé va quedar arreglat.

D'altra banda, hi havia un comportament anòmal respecte a l'estipulat en el disseny. De fet, més que un error d'implementació era un error de localització de la base de dades consultada, la qual depèn de l'idioma

del navegador/dispositiu. La solució, tenint en compte que no s'ha implementat localització per *iCompound* (tot i que entra en els plans de millores futures), passava per readaptar l'algorisme de tractament de la informació sobre les molècules consultades.

## 5.2 Avaluació de l'aplicació

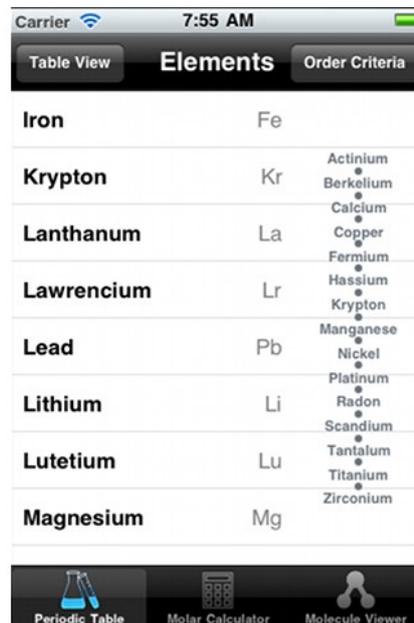
Per poder determinar si els estàndards mínims d'usabilitat establerts inicialment es compleixen, el millor es fer un estudi de les opinions del usuaris, basant-se en els perfils definits en el procés de definició i anàlisi de requeriments (punt 3.3.2). Òbviament, el temps és un factor determinant a l'hora de poder realitzar totes les tasques que se suposen inherents al procés de disseny i implementació d'una aplicació, i en el cas que ens ocupa, és massa limitat per fer un estudi en profunditat. Així doncs, i com a mostra, s'han fet dues proves d'us sobre un dispositiu físic (iPod Touch) a dos persones que encaixen dintre dels perfils d'usuari tipus estudiant universitari (avançat) i aficionat a la química (bàsic).

A continuació es detallen algunes de les problemàtiques comuns assenyalades pels mateixos respecte a la satisfacció en l'experiència d'usuari, així com els aspectes positius globals.

### 5.2.1 Deficiències detectades i solucions proposades

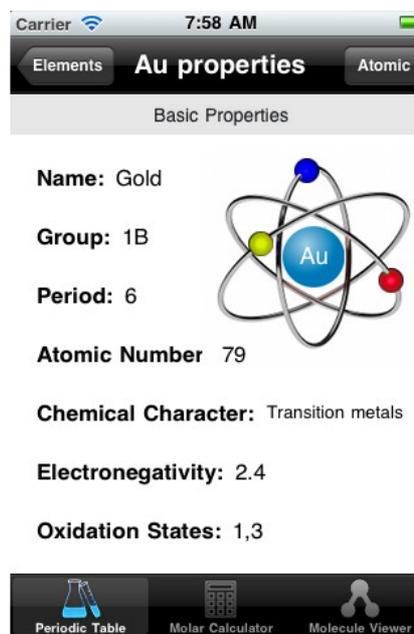
Les proves sobre utilització de *iCompound* han permès detectar certes deficiències en el que es refereix a experiència d'usuari:

- La configuració inicial de la llista d'elements, formada per una llista agrupada (segons els criteris de cerca) en la que cada cel·la tenia el color del caràcter químic de l'element representat no resultava del tot agradable. Així doncs, es va optar per fer un format de llista no agrupat i de colors clars, que resultava molt més entenedor. La figura 47 mostra com era inicialment i com qued després de les correccions apuntades.
- La navegació entre els elements resultava certament farragosa, ja que l'alt nombre de cel·les (més de cent) feia complicat arribar al final de la llista. Com a solució, elegant i efectiva, s'opta per introduir un índex lateral que permet anar, segons grups, a les cel·les més allunyades de manera senzilla (veure figura 47).
- Els botons de la barra inferior per commutar amb la vista de taula i la vista de ordenació s'integren a la barra de navegació superior, com a resposta a les anotacions fetes pels usuaris respecte a incrementar de part de a pantalla utilitzada pel llistat d'elements (veure figura 47).



Imatge 47: Vista inicial del llistat d'elements (esquerra) i vista millorada (dreta).

- La visualització de les propietats bàsiques, en que el nom atòmic quedava dintre d'un quadre degradat no resultava atractiva. Es va decidir cercar una imatge més adient, que a més, aprofita el concepte de color segons caràcter químic per donar més informació sobre l'element que s'esta analitzant (aplicat al nucli de l'àtom).



Imatge 48: Vista inicial de propietats bàsiques d'element (esquerra) i vista millorada (dreta).

- Es van detectar errors en la llegenda mostrada en la visualització de les molècules, que podrien induir a errors, especialment a usuaris no experts (com va succeir en un dels dos casos analitzats). Una revisió i correcció del problema va corregir-los.

### 5.2.2 Aspectes destacables

Un cop tingudes en compte les opinions dels usuaris, s'ha tornat a fer proves d'usabilitat. L'increment de satisfacció respecte a les primeres proves ha estat notable, podent destacar del producte final:

- Senzillesa en la utilització.
- Claredat en la informació mostrada.
- L'aplicació resulta altament intuïtiva i precisa.
- Tant l'aspecte gràfic com la navegació contribueixen a un alt grau d'amigabilitat.
- Organització de continguts ben estructurada.

## 5.3 Resultat final

El producte final es pot veure tant en els vídeos que s'adjunten amb aquesta memòria, com en el paquet de codi font adjunt, preparat per posar en marxa *iCompound* i verificar tot el que s'ha explicat al llarg d'aquesta memòria.

És possible afirmar que, amb força precisió, s'han assolit les expectatives creades a l'inici del projecte, tot i haver hagut de fer certes modificacions per adaptar el producte final tant als processos a realitzar com al temps real del que s'ha disposat per dur-los a terme.

## 5.4 Possibles millores i futur de l'aplicació

Dintre de l'apartat millores cal recordar el fet de no haver pogut realitzar la representació en tres dimensions de les molècules, degut a que la corba d'aprenentatge d'Open GL ES és elevada, i l'assimilació del coneixement necessari no era possible tenint en compte la limitació temporal que es tenia. D'altra banda seria interessant:

- Implementar un índex pel llistat de molècules, tal i com s'ha fet pel llistat d'elements. S'ha de pensar que l'usuari pot arribar a descarregar un alt nombre de compostos i, per tant, la navegació resultaria insatisfactòria tal i com està actualment.

- Optimitzar el codi font en certs algorismes (concretament, els de la calculadora i descàrrega) per tal de dotar de més agilitat a l'aplicació. Algunes d'aquestes mesures podrien passar per l'ús de funcions relacionades amb expressions regulars, que millorarien la cerca d'informació.
- Oferir més opcions al cercador de molècules. Aquestes poden anar des de permetre diferents criteris de cerca (actualment només funciona la cerca per nom), a més de implementar l'accés a més servidors especialitzats, fet que incrementaria el nombre de molècules disponibles per descarregar.
- Permetre, tal i com s'havia pensat inicialment, en oferir opcions de zoom sobre la vista de taula periòdica clàssica, per fer més enriquidora l'experiència d'usuari.
- Adaptar l'aplicació per córrer també sobre iPad. En principi, pensada per iPhone i iPod, aquesta adaptació podria resultar interessant, a més de permetre afegir alguna nova funcionalitat aprofitant les dimensions i propietats del tablet d'Apple.
- Aplicar tècniques d'internacionalització, per permetre que un públic més ampli pugui treballar de manera nativa en el seu idioma. En principi, a idea és que *iCompound* pugui estar disponible en anglès (llengua per defecte), espanyol, català i francès.

Un cop aplicades aquestes millores, i potser alguna més que pugui plantejar-se, la idea és no deixar el projecte aparcad, sinó millorar-lo i publicar-lo a *Apple Store* per veure quina sortida té en un mercat real, on la competència pot ser dura però enriquidora. Aquest pas final és, sense cap mena de dubte, la prova de foc per *iCompound*.

## 6 Conclusions

A l'inici d'aquest projecte es plantejaven diferents objectius a assolir. En aquest punt, en que ja s'ha arribat al final, és possible afirmar que la gran majoria d'expectatives s'han cobert de manera satisfactòria.

El principal objectiu, que era dissenyar i implementar una aplicació sobre iOS com és *iCompound* ha suposat un repte important. No només per tot el procés d'adquisició de coneixements sobre l'entorn a treballar, sinó també per la dificultat afegida d'haver de tornar a recordar conceptes relacionats amb la química (anteriorment adquirits, però oblidats). Així doncs, és possible afirmar que l'enriquiment cultural que ha suposat la consecució d'aquest projecte ha estat important.

A més, el fet d'haver participat en un procés de creació des de zero d'una aplicació, passant per tots (o gairebé tots) els estadis del cicle de vida d'un producte informàtic, ha proporcionat una visió més clarificadora de les dificultats que implica. Ha quedat totalment palesa la importància de cada un dels passos a seguir per a poder acabar obtenint un producte de qualitat i ajustat a les necessitats de l'usuari. Destacar d'aquest procés la importància de certs aspectes:

- Planificació. Estructurar bé les tasques a fer i fer una distribució temporal acurada és la base per garantir l'èxit del procés. En el cas d'aquest projecte, tot i que hi hagut certs desfases entre la realitat i el planificat (especialment en la part d'implementació degut a la previsió no acomplerta de la visualització en 3D), es pot afirmar que part de l'èxit ha radicat en haver-lo seguit de manera fidedigna.
- Iteració i interacció. Es considera que, en un disseny enfocat a garantir la satisfacció d'usuari, és una màxima que sempre s'ha de seguir, ja que permet detectar deficiències d'una manera senzilla i natural, contribuint a la consecució d'un producte de qualitat millorada.
- L'usuari és el centre del desenvolupament. Una eina que ha de ser d'utilitat per un usuari ha de tenir-lo sempre en compte. Conèixer les seves preferències i necessitats en tot moment pot ser un factor determinant per garantir l'èxit de l'aplicació.

Cal destacar la potència, elegància i eficiència del llenguatge Objective-C. Aprendre la seva idiosincràsia no ha estat pas complicada, més tenint en compte el bagatge adquirit (tant en la vida professional com en la docent) en d'altres llenguatges com C o C++. També es remarcable la senzillesa i potència de les eines ofertes per Apple (X-Code i Interface Builder) utilitzat per dur a terme aquest projecte.

Finalment, dir que tot i no haver pogut assolir el 100% de la idea inicial sobre les característiques d'*iCompound*, s'ha aconseguit fer un disseny funcional i ajustat a tots els paràmetres sobre DCU que una aplicació com aquesta ha de tenir. En un futur, que comença amb la finalització d'aquest treball d'investigació, s'han d'assolir els objectius no acomplerts per tal de publicar aquesta eina a Apple Store i comprovar fins a quin punt les bones sensacions tenen raó de ser. Francament, i tenint en compte que el projecte era ambiciós, la realitat és que resulta un orgull haver arribat als resultats obtinguts.

## 7 Glossari

<b>APPLE INC</b>	Multinacional dels EEUU que té com activitats empresarials el disseny i producció de programari / maquinari informàtic.
<b>TABLET</b>	Ordinador portàtil de dimensions reduïdes gestionat amb controls tàctils.
<b>IPHONE</b>	Terminal telefònic intel·ligent, multimèdia i tàctil dissenyat, produït i comercialitzat per Apple inc.
<b>IPOD</b>	Dispositiu multimèdia (tàctil en la versió Touch) dissenyat, produït i comercialitzat per Apple inc.
<b>IPAD</b>	Tablet multimèdia dissenyat, produït i comercialitzat per Apple inc.
<b>IOS</b>	Sistema operatiu propietat d'Apple inc, utilitzat en els seus dispositius portàtils.
<b>OBJECTIVE-C</b>	Llenguatge de programació orienta a objectes basat en C, però similar a SmallTalk, utilitzat per MacOSX.
<b>COCOA-TOUCH</b>	Interfície de programació d'aplicacions (API) que proporciona una capa d'abstracció del sistema per a la creació de programes sota iOS.
<b>OPEN GL ES</b>	Variant de l'API gràfica Open GL per dispositius portàtils. És una especificació estàndard destinada a produir gràfics en dos i tres dimensions.
<b>CORE GRAPHICS</b>	API de Quarz 2D que ofereix funcions per creació i renderització de grafics en dos dimensions.
<b>X-CODE</b>	Entorn de desenvolupament gràfic (IDE) creat per Apple, destinat al disseny d'aplicacions per dispositius de la mateixa empresa.
<b>INTERFACE BUILDER</b>	Part de X-Code per dissenyar les interfícies dels programes d'Apple.
<b>IPO</b>	Interacció persona-ordinador. Estudi de la comunicació entre persones i computadors.
<b>DCU</b>	Disseny centrat en l'usuari. Filosofia basada en crear productes que

esdevinguin eines útils per a les persones, tot garantint la seva satisfacció en l'experiència d'usuari.

**HUMAN INTERFACE GUIDELINES** Documentació destinada a desenvolupadors que ofereixen recomanacions orientades a millorar l'experiència d'usuari.

**MVC** Model Vista Controlador. Patró d'arquitectura de programari que separa les dades de la interfície i de la lògica de negoci, tractant a cada un com peces diferents del programa.

**WIREFRAME** Representació esquemàtica associada al disseny d'una aplicació, que permet definir de manera gràfica part de les interfícies i comportaments.

**RENDER** Procés per generar imatges en tres dimensions a partir de models establerts en dues dimensions.

**SERIALITZACIÓ** Codificació d'un objecte en un format sensible de ser emmagatzemat o tramés com a agrupació de bytes.

**DELEGATE** Patró de programació que permet crear objectes que realitzen transaccions entre d'altres, amb la missió de minimitzar l'acoblament.

**MODAL VIEW CONTROLLER** Controlador que permet interrompre el flux de treball normal per realitzar una tasca concreta i acotada, mostrant una nova finestra.

**GARBAGE COLLECTOR** Mecanisme inclòs en diversos llenguatges de programació destinat a facilitar la gestió de memòria, al alliberar de manera automàtica recursos.

**IUPAC** International Union of Pure and Applied Chemistry

**NIST** National Institute of Standards and Technology

## 8 Bibliografia

- [1] Allan A. (2010). *Learning iPhone Programming*. United States of America: O'Reilly Media, Inc. [Consulta: Juliol-Octubre 2011].
- [2] Buck, E. M.; Yacktman, D. A. (2010). *Cocoa Design Patterns*. United States of America: Pearson Education, Inc. [Consulta: Octubre 2011].
- [3] Mark, D.; Nutting, J.; LaMarche, J. (2011). *Beginning iPhone 4 Development*. Apress. [Consulta: Octubre 2011].
- [4] Kochan, S. G. (2011). *Programming in objective-C (third edition)*. United States of America: Pearson Education, Inc. [Consulta: Octubre 2011].
- [5] Apple Inc. (2010). *iOS Developer Library*. [en línia].  
<http://developer.apple.com/library/ios/navigation/> [Consulta: Setembre-Octubre 2011]
- [6] Apple Inc. (2010). *iOS Human Interface Guidelines*. [en línia].  
<http://developer.apple.com/library/IOS/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html> [Consulta: Setembre-Octubre 2011]
- [7] Johanna W. (2009). *New research on the demographics and behavioral characteristics of iPhone And iPod touch users from AdMob and comScore*. [en línia]. <http://blog.admob.com/2009/06/16/new-research-on-the-demographics-and-behavioral-characteristics-of-iphone-and-ipod-touch-users-from-admob-and-comscore/> [Consulta: Octubre 2011]
- [8] Chan, Raymon. (2002). *Química General (7ª edición)*. Colombia: McGraw Hill Interamericana. [Consulta: Juliol-Octubre 2011].
- [9] Helmetine, Anne Marie. (2011). *Atom diagrams: Diagrams of Atoms and Isotopes*. [en línia]. <http://chemistry.about.com/od/elementfacts/ig/Atom-Diagrams/> [Consulta: Octubre-Novembre 2011]
- [10] McGraw Hill Interamericana de España. (2011). *Tabla periódica de los elementos*. [en línia]. [http://www.mcgraw-hill.es/bcv/tabla\\_periodica/mc.html](http://www.mcgraw-hill.es/bcv/tabla_periodica/mc.html) [Consulta: Setembre- Noviembre 2011]
- [11] IUPAC [en línia]. [http://old.iupac.org/dhtml\\_home.html](http://old.iupac.org/dhtml_home.html) [Consulta :Setembre-Diciembre 2011]
- [12] NIST. *Libro del Web de Química*. [en línia]. <http://webbook.nist.gov/chemistry/> [Consulta: Noviembre-Diciembre 2011]
- [13] Wikipedia. (2011). *Chemical file format*. [en línia]. [http://en.wikipedia.org/wiki/Chemical\\_file\\_format](http://en.wikipedia.org/wiki/Chemical_file_format) [Consulta: Setembre 2011]

- [14] Accorn NMR inc. (2006). *Displaying molfiles*. [en línia].  
<http://www.acornnmr.com/NutsHelp/molfiles.htm> [Consulta: Setembre-Novembre 2011]
- [15] Sunset Lake Software. (2008). *Lessons from Molecules: OpenGL ES*. [en línia].  
<http://www.sunsetlakesoftware.com/2008/08/05/lessons-molecules-opengl-es> [Consulta: Novembre 2011]
- [16] Ray Wenderlich. (2011). *OpenGL ES 2.0 for iPhone Tutorial*. [en línia].  
<http://www.raywenderlich.com/3664/opengl-es-2-0-for-iphone-tutorial> [Consulta: Octubre-Novembre 2011]
- [17] Matt Gallagher. (2010). *Cocoa with love: 5 ways to draw a 2D shape with a hole in CoreGraphics*. [en línia]. <http://cocoawithlove.com/2010/05/5-ways-to-draw-2d-shape-with-hole-in.html> [Consulta: Novembre-Desembre 2011]