

# Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

**Enginyeria Tècnica en Informàtica de Gestió**

**ASSESSOR PER LA SALUT MÒBIL**

**Memòria**

**XAVIER JUAN RAMOS  
PONENT: LINA JUAN NADAL**

PRIMAVERA 2011



TecnoCampus  
Mataró-Maresme



## **Resum**

La finalitat d'aquest projecte, ha estat desenvolupar una aplicació per a iPhone que enregistra el recorregut que fa l'usuari caminant, corrent o en bicicleta i mostra informació sobre les calories cremades, distància recorreguda i temps entre d'altres, i mostra informació sobre el pes ideal.

S'analitza perquè s'han escollit aquestes funcionalitats i es descriu com s'han desenvolupat i quins entrebancs han sorgit durant el procés. Per últim s'expliquen les característiques de la tecnologia utilitzada.

## **Resumen**

La finalidad de éste proyecto, ha sido desarrollar una aplicación para iPhone que graba el recorrido que hace el usuario andando, corriendo o en bicicleta y muestra información sobre las calorías quemadas, distancia recorrida y tiempo entre otras, y muestra información sobre el peso ideal.

Se analiza el porqué se han escogido estas funcionalidades y se describe como se han desarrollado y qué estorbos han surgido durante el proceso. Por último se explican las características de la tecnología utilizada.

## **Abstract**

The purpose of this project has been to develop an iPhone application that records the track that the user does walking, running or cycling, and shows information of calories burned, distance and time among others, and displays information about the weight.

It is analyzed why these features have been chosen and is described as have developed and what impediments have arisen during the process. Finally, explains the features of the technology used.



# Índex.

Índex de figures. ....	III
Índex de taules. ....	V
Glossari de termes.....	VII
1. Objectius. ....	1
1.1. Propòsit. ....	1
1.2. Finalitat. ....	1
1.3. Objecte. ....	1
1.4. Abast. ....	1
2. Informació sobre la realització del projecte.....	3
2.1. Objectius del projecte. ....	3
2.2. Estudi de mercat.....	4
3. Estudi de la solució. ....	7
3.1. Estudi de la tecnologia. ....	7
3.2. Estudi de les dades necessàries.....	7
3.3. Estudi de les funcionalitats de l'aplicació.....	11
4. Desenvolupament. ....	13
4.1. Entorn de programació.....	13
4.2. Anàlisi dels casos d'ús.....	15
4.3. Estudi del desenvolupament de cada cas d'ús. ....	15
4.3.1. Enregistrar la sessió.....	16
4.3.2. Historial de sessions. ....	18
4.3.3. Càlcul de l'índex de massa corporal.....	19
4.3.4. Preferències. ....	20
4.4. Desenvolupament de l'aplicació.....	20
4.5. Classes. ....	22
4.5.1. ProjecteAppDelegate.....	22
4.5.2. SessionViewController.....	23
4.5.3. HistoryViewController.....	24
4.5.4. HistoryViewListController.....	24
4.5.5. HistoryDetailViewController.....	26
4.5.6. HistoryMapViewController.....	26

4.5.7. IMCViewController.....	27
4.5.8. LocationManager.....	28
4.5.9. Session.....	29
4.5.10. GradientView.....	29
4.5.10. Preferences.....	29
4.6. Altres arxius.....	29
4.6.1. Preferències.....	29
4.6.2. Projecte-Info.plist.....	32
4.6.3 Projecte-Prefix.pch.....	32
5. Proves de l'aplicació.....	33
6. Característiques de Cocoa i de Objective-C.....	35
6.1. Estil.....	36
6.2. Gestió de la memòria.....	37
6.3. Crides a objectes.....	37
7. Llicència.....	39
8. Conclusions.....	41
9. Referències.....	43

## Índex de figures.

Fig. 4.1. Xcode 4 amb visualització d'entorn de programació.....	13
Fig. 4.2. Xcode 4 amb visualització d'entorn d'editor d'interfícies gràfiques.....	14
Fig. 4.3. Ajuda incorporada al Xcode 4.....	14
Fig. 4.4. Pantalla de sessió actual, llistat de sessions i índex de massa corporal.....	21
Fig. 4.5. Detall del nom de les imatges amb dues resolucions diferents.....	21
Fig. 4.6. Icona de l'aplicació.....	22
Fig. 4.7. Detall de la pantalla que indica que les preferències s'han d'emplenar.....	23
Fig. 4.8. Sessió actual amb les mètriques configurades a les preferències.....	24
Fig. 4.9. Detall del text d'esborrar.....	25
Fig. 4.10. Detall de la sessió amb les mètriques configurades a les preferències.....	26
Fig. 4.11. Visualització del recorregut amb els tres estils disponibles.....	27
Fig. 4.12. Índex de massa corporal amb diferents mètriques configurades.....	28
Fig. 4.13. Detall del teclat numèric sense decimals de les preferències.....	30
Fig. 4.14. Editor de l'arxiu de preferències integrat a Xcode 4.....	31
Fig. 4.15. Arxiu de preferències visualitzat amb un editor de text extern a Xcode 4.....	31
Fig. 4.16. Arxiu d'opcions visualitzat amb Xcode 4.....	32
Fig. 5.1. Primera sessió de prova de 47 segons.....	33
Fig. 5.2. Sessió de 712 segons i 7,8 kilòmetres.....	34
Fig. 5.3. Sessió de 28 minuts i 2,44 kilòmetres.....	34

Fig. 6.1. <i>Frameworks</i> que ofereix Cocoa.....	35
Fig. 6.2. <i>Frameworks</i> que ofereix Cocoa-Touch.....	35
Fig. 7.1. Preferències de signatura de l'aplicació.....	39



## Índex de taules.

Taula 3.1. Ritme cardíac segons l'edat.....	8
Taula 3.2. Factors segons velocitat de l'esport.....	9
Taula 3.3. Factors calòrics segons l'activitat.....	10
Taula 4.1. Factors de multiplicació pel pes ideal.....	19
Taula 3.3. Descripcions segons el valor de l'IMC.....	20



## Glossari de termes.

- iOS        Sistema Operatiu d'Apple® utilitzat als iPhone, iPod i iPad.
- Cocoa      Conjunt de *frameworks* que permeten el desenvolupament d'aplicacions per Mac.
- Objective-C    Llenguatge de programació orientat a objectes utilitzat per desenvolupar aplicacions mitjançant Cocoa.
- timestamp    Seqüència de caràcters numèrics que denota el nombre de segons que han passat des del 1 de gener de 1970 UTC.
- Xcode        Aplicació creada que conté totes les utilitats necessàries per desenvolupar aplicacions pels sistemes operatius d'Apple.



# **1. Objectius.**

## **1.1. Propòsit.**

Explicar com s'ha portat a terme el projecte i quins passos i decisions s'han pres durant el seu desenvolupament.

## **1.2. Finalitat.**

Mostrar al lector quin procés s'ha realitzat per desenvolupar l'aplicació i explicar quins problemes i solucions s'han trobat durant el desenvolupament.

## **1.3. Objecte.**

Documentació del Projecte Final de Carrera, consistent en tres document: memòria, estudi econòmic i annexos.

## **1.4. Abast.**

S'explica el procés que s'ha seguit per desenvolupar el projecte incloent l'estudi previ, l'explicació del flux d'informació dins de l'aplicació i l'explicació de les solucions als problemes que s'han trobat durant el desenvolupament. Resten fora de l'abast d'aquest document l'explicació dels *frameworks*, llenguatges o eines utilitzades pel seu desenvolupament, tret de la seva menció per tal d'entendre millor l'entorn de desenvolupament.



## **2. Informació sobre la realització del projecte.**

El projecte s'ha desenvolupat com a conseqüència de l'assignació del projecte proposat pel Centre de Competències d'Integració (CCI). El CCI és un centre de transferència de tecnologia del Tecnocampus, que treballa conjuntament amb la fundació TIC i Salut, i que té com a objectiu genèric integrar tots els centres de salut i les seves dades clíniques en una única xarxa nacional.

### **2.1. Objectius del projecte.**

L'objectiu del projecte és treballar la tecnologia que permet realitzar una aplicació en l'àmbit de la salut.

Tenint en compte que els objectius són molt amplis, s'han valorat diferents opcions com gestionar els aliments ingerits i proposar una dieta per tal d'arribar al pes ideal. Aquest tipus d'aplicació requereix molta intervenció per part de l'usuari, que ha d'introduir cada aliment que ingereixi en cada àpat, i requereix d'un estudi profund de dietètica per tal de mostrar resultats que siguin correctes per la salut de l'usuari. També requereix un públic que estigui disposat a interactuar molt amb el dispositiu, per tant la gent gran queda descartada potencialment.

Un altre opció és una aplicació que ajudi a fer exercici, com per exemple un entrenador personal. Aquest tipus d'aplicacions mostren una taula amb els exercicis possibles i expliquen la forma correcta de realitzar-los, però no assisteix activament sinó que només indica la forma en que s'han de fer. Aquest tipus també descarta al tipus de públic que no vulgui o pugui fer exercici i que no estigui disposat a anar seguint les instruccions d'un terminal mòbil.

Per tal de trobar un projecte més dinàmic i que no descarti cap públic potencial, s'ha pensat desenvolupar una aplicació que enregistri el recorregut de l'usuari a partir del GPS del terminal, i a partir de les dades obtingudes indiqui les calories cremades. Per tal de tenir un historial amb totes les sessions, s'enregistraran de forma que es puguin consultar i es pugui veure el recorregut en un mapa. També s'ha pensat fer una secció que indiqui quin és el pes ideal de l'usuari.

Aquest tipus d'aplicacions no requereixen de molta interacció amb l'usuari, ja que un cop configurada només cal prémer el botó d'iniciar la sessió i deixar el dispositiu a la butxaca. El fet de tenir un historial on es puguin consultar totes les sessions amb la distància recorreguda, temps emprat, velocitat mitjana i calories cremades, serveixen d'al·licient per anar a córrer o caminar. Com l'exercici recomanat per baixar el pes és caminar, s'ha pensat en crear una secció que indiqui quin és el pes ideal de l'usuari.

## 2.2. Estudi de mercat.

Pel disseny de l'aplicació es tindran en compte altres aplicacions semblats com per exemple:

- runtastic PRO
- RunKeeper
- miCoach
- Walkmeter
- endomondo
- sportypal

Totes les aplicacions tenen en comú les següents característiques:

- Utilitzen el GPS per enregistrar el recorregut i així poder mesurar la distància
- Enregistren el temps del recorregut
- Enregistren la sessió per poder consultar un històric

A partir de les dades obtingudes del GPS i del temps, i un cop havent configurat l'alçada i el pes, es poden calcular les calories cremades o el nombre de passes. L'aplicació *miCoach* utilitza obligatòriament la seva plataforma online, on es pot configurar un recorregut predefinit i quin és el calçat que s'utilitza, de forma que l'aplicació pot informar de quan cal canviar-lo segons el quilòmetres recorreguts.



Cal tenir en compte que les aplicacions consten de dues parts: una que enregistra les dades i un altre que les analitza. La part que enregistra les dades és molt similar en totes elles ja que utilitzen les funcionalitats que ofereix el dispositiu. La segona part és molt diferent entre les aplicacions, fins el punt que els resultats són diferents fent el mateix recorregut. Això és degut a que cada aplicació utilitza algorismes propis que han estat estudiats de forma diferent. L'empresa que desenvolupa *runtastic* té aplicacions dedicades a cada esport (*walktastic* – per caminar, *wintersports* – per esports d'hivern, *cyctastic* – per ciclisme).



### **3. Estudi de la solució.**

Per tal de desenvolupar l'aplicació s'ha d'escollir una plataforma de desenvolupament i cal fer un estudi de les possibilitats que ofereix cada plataforma.

#### **3.1. Estudi de la tecnologia.**

Actualment hi ha dues grans plataformes de desenvolupament al mercat. Una és Android, desenvolupada per Google, i l'altre és iOS, desenvolupada per Apple. Les altres plataformes com la de Blackberry o la de Nokia estan perdent mercat vers les dues primeres, per tant no s'han tingut en compte. Les dues plataformes escollides disposen de terminals al mercat capaços d'enregistrar la posició GPS, requisit indispensable per desenvolupar el projecte, així com d'entorns de programació documentats i simuladors de terminals per tal de poder provar l'aplicació. Com l'autor del projecte té un terminal iOS, i degut al gran èxit que té la plataforma d'Apple, es desenvoluparà sobre aquesta plataforma.

#### **3.2. Estudi de les dades necessàries.**

Per tal de definir quina serà la funcionalitat de l'aplicació i com podrà assessorar a l'usuari en l'àmbit de la salut, cal saber quines dades són necessàries i si el dispositiu és capaç d'aconseguir-les.

S'ha de tenir en compte que cada ésser humà té una condició física diferent i per tant, no hi ha una fórmula exacta per tal de poder saber amb exactitud quin és el nivell d'exigència al realitzar un esport.

La millor forma de mesurar l'estat del cos és a partir del nombre de batecs del cor. Les pulsacions indiquen quin és el nivell d'activitat al que està exposat el cos, i relaciona l'exigència de l'activitat amb la condició física.

A la següent taula es pot veure el nivell d'exigència relacionant l'edat i les pulsacions per minut:

<b>Edat</b>	<b>Ritme cardíac màxim</b>	<b>Ritme cardíac ideal (75% del màxim)</b>	<b>Zona d'objectiu (70% - 85% del màxim)</b>
20	200	150	140 a 170
25	195	146	137 a 166
30	190	142	133 a 162
35	185	139	130 a 157
40	180	135	126 a 153
45	175	131	123 a 149
50	170	127	119 a 145
55	165	124	116 a 140
60	160	120	112 a 136
65	155	116	109 a 132
70	150	112	105 a 128

Taula 3.1. Ritme cardíac segons l'edat.

L'inconvenient és que el dispositiu no té la capacitat de mesurar els batecs del cor, per tant aquesta solució no és vàlida. A l'hora de buscar mètodes alternatius, s'han cercat quins mètodes permeten relacionar les dades que el dispositiu pot mesurar amb la condició física del cos.

El dispositiu permet enregistrar les coordenades GPS en un moment determinat, per tant es pot calcular la velocitat dividint la distància entre dos punts i la diferència de temps entre ells. També existeix la possibilitat d'introduir altres dades a les preferències de l'aplicació, de forma que cal trobar mètodes relacionin la condició física amb dades com la velocitat, el temps, la distància o d'altres que es puguin incorporar a les preferències de l'aplicació.

Les calories cremades es poden calcular si es té en compte quin tipus d'exercici s'està realitzant. Es tracta de multiplicar el pes, el temps i un paràmetre que indica el nivell d'activitat.

- El pes ha d'estar en Lliures o s'ha de multiplicar per 2,2 si està en Kg.
- El temps ha d'estar en hores o s'ha de multiplicar per 60 si està en minuts.

El paràmetre que indica l'activitat s'indica a la Taula 3.3.

Els esports indicats a la taula depenen del nivell d'intensitat, pel que l'aplicació no té cap sistema que pugui detectar quin és l'esport que s'està realitzant, per tant només s'utilitzaran els esports que depenguin de la velocitat.

El factor de l'esport no respon a una fórmula lineal, ja que els factors al córrer a 13, 16 o 19,3 km/h són de 6,4, 7,5 i 7,9. De 13 a 16 km/h hi ha una diferència de 3 km/h i un increment del factor de 1,1. Segons aquestes dades, de 16 a 19,3 km/h hi ha un increment de 3,3 km/h per tant el factor hauria d'incrementar-se en 1,21 i en canvi l'increment és de 0,4.

Per tal de trobar una fórmula que permeti saber el factor a velocitats que no es troben a la taula, és necessari tractament matemàtic que requereix temps. Com el desajust entre punts no és gaire elevat i tenint en compte la precisió del GPS, s'utilitzarà una escala de valors calculada de forma lineal entre cada punt. Ja que caminar i córrer són esports complementaris que depenen de la velocitat, l'aplicació només diferenciarà entre bicicleta i córrer.

<b>Bicicleta</b>	
<b>Velocitat (km/h)</b>	<b>Factor</b>
< 3	1
9	1,6
16	3
21	4,26
40	4,66
130	10

<b>Córrer</b>	
<b>Velocitat (km/h)</b>	<b>Factor</b>
< 2	1,4
4	1,75
4,8	2
5,6	2,3
6,5	2,67
7,2	3,1
8	3,62
9,7	4,8
11,3	5,4
13	6,3
16	7,5
19,3	7,9
45	10

Taula 3.2. Factors segons velocitat per l'esport.

Valor	Activitat	Valor	Activitat	Valor	Activitat
2,65	Aeròbic Balli (Casual)	4,6	Saltar la corda (ràpid)	4,09	Pujar escales (20,3 cm)
3	Aeròbic Balli (Moderat)	0,6	Recolzat	4,42	Pujar escales (25,4 cm)
3,71	Aeròbic Balli (Intens)	5,35	Arts marciais	4,91	Pujar escales (30,5 cm)
2,7	Bàsquet (Pista completa)	4,88	Padel	0,98	Estar dempeus (oficina)
1,67	Bàsquet (Mitja pista)	3,6	Rollerblading (Casual)	0,76	Estar dempeus (quiet)
1,6	Bicicleta (9 kmph)	4,25	Rollerblading (Ràpid)	3	Nedar (Casual)
3	Bicicleta (16 kmph)	3,27	Rem màquina (Moderat)	10	Nedar (Intens)
4,26	Bicicleta (21 kmph)	4,91	Rem màquina (intens)	2,56	Tennis (Dobles)
4,66	Ciclisme (Carrera)	2,92	Córrer (6,5 kmph)	3	Tennis (Individual)
2,62	Ball (Casual)	3,62	Córrer (8 kmph)	1,39	Voleibol (Casual)
2,84	Ball (Moderat)	4,8	Córrer (9,7 kmph)	3,98	Voleibol (intens)
3,65	Ball (Intens)	5,4	Córrer (11,3 kmph)	1,4	Caminar (3,2 kmph)
3,7	Esports d'Equip (Competició)	6,3	Córrer (13 kmph)	1,75	Caminar (4 kmph)
2,75	Frisbie	7,5	Córrer (16 kmph)	2	Caminar (4,8 kmph)
2,37	Golf (Duent els pals)	7,9	Córrer (19,3 kmph)	2,3	Caminar (5,6 kmph)
2	Golf (Amb Carro)	0,76	Assegut (oficina)	2,67	Caminar (6,5 kmph)
1,96	Gimnàstica	0,6	Assegut (tranquil)	3,1	Caminar (7,2 kmph)
1,7	Labors de casa	1,91	Softball	3,2	Entrenament de peses (intens)
3,7	Saltar la corda (lent)	3,71	Pujar escales (15,2 cm)	2,35	Entrenament de peses (Normal)

Taula 3.3. Factors calòrics segons l'activitat.

Amb aquestes dades l'aplicació serà capaç de realitzar la mesura de les calories cremades.

Per tal d'assessorar a l'usuari, es pot incorporar un apartat que mostri l'índex de massa corporal. Aquest índex determina si el pes és el correcte tenint en compte el sexe, l'alçada i la complexió de l'usuari.

### **3.3. Estudi de les funcionalitats de l'aplicació.**

Sabent quines dades es poden aconseguir cal especificar quines funcionalitats ha de proporcionar l'aplicació. Una funcionalitat ha de ser enregistrar el recorregut i ha de mostrar informació sobre les calories cremades. Una altra funcionalitat ha de ser poder recuperar o eliminar una sessió enregistrada. L'altre funcionalitat ha d'oferir informació sobre l'índex de massa corporal segons les dades configurades.





## 4. Desenvolupament.

Per desenvolupar l'aplicació cal conèixer l'entorn de programació que requereix Apple.

### 4.1. Entorn de programació.

Apple ofereix el seu entorn de programació de forma gratuïta a l'hora de fer la compra del sistema operatiu. De forma addicional es pot comprar a través de la seva plataforma App Store o descarregar-se des del seu web <http://developer.apple.com>

Xcode 4 és una eina que integra l'editor de codi, l'editor d'interfícies gràfiques i l'ajuda, de forma que no cal cap eina més per poder desenvolupar l'aplicació.

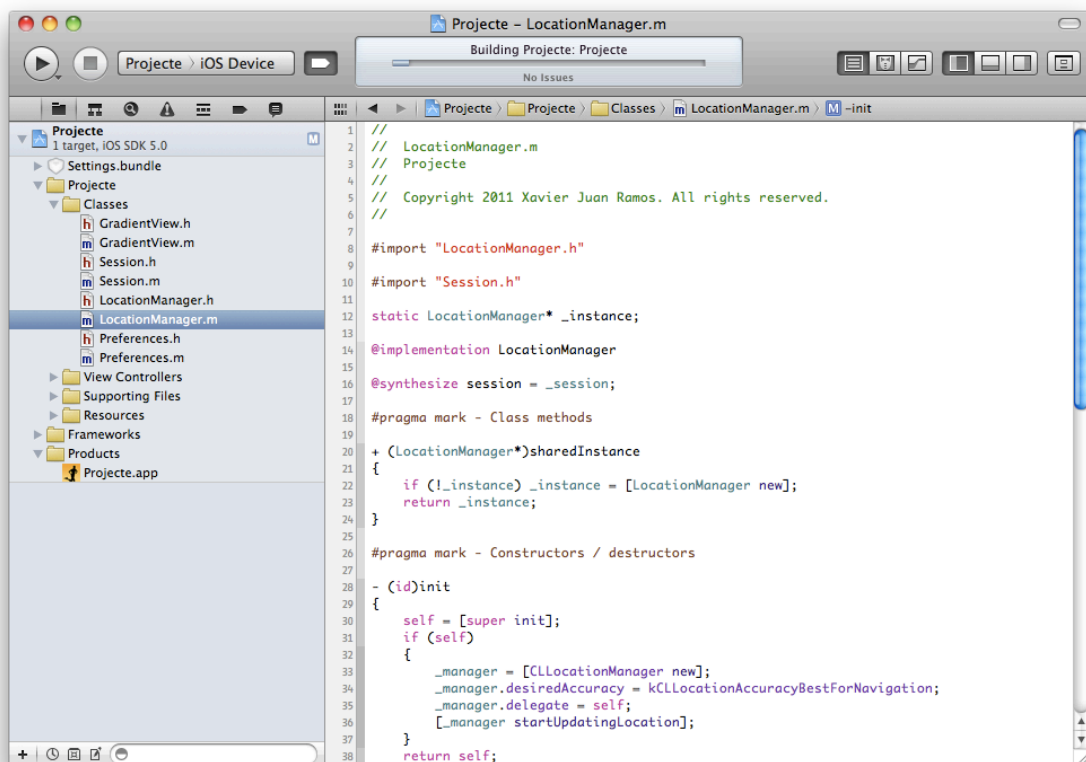


Fig. 4.1. Xcode 4 amb visualització d'entorn de programació.

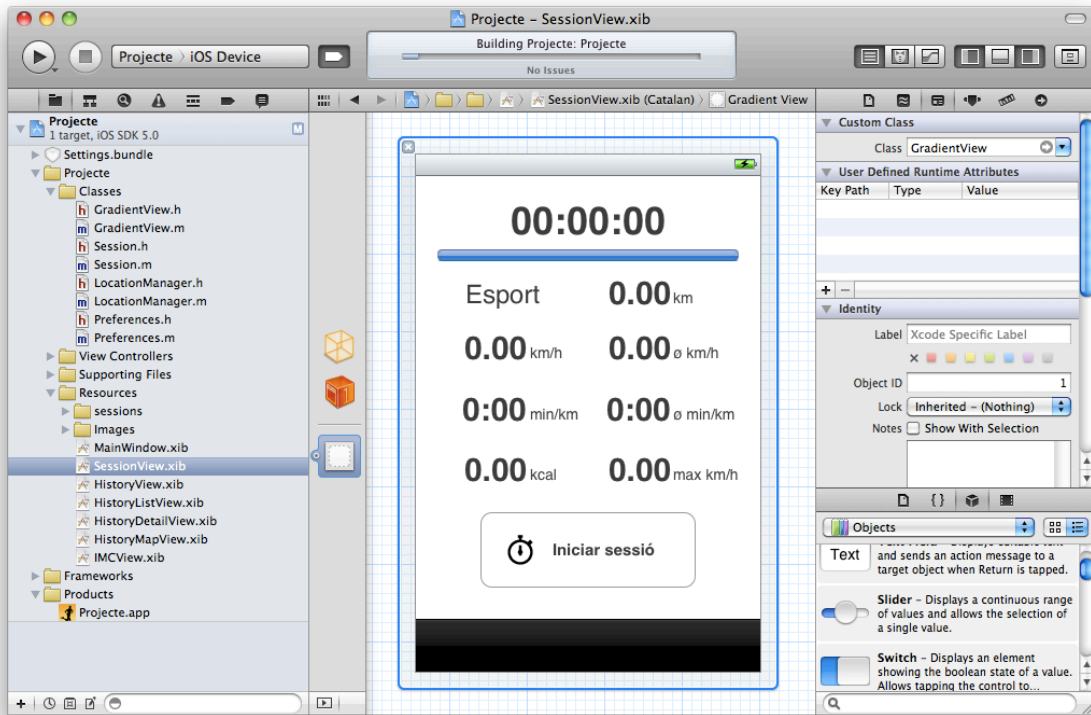


Fig. 4.2. Xcode 4 amb visualització d'entorn d'editor d'interfícies gràfiques.

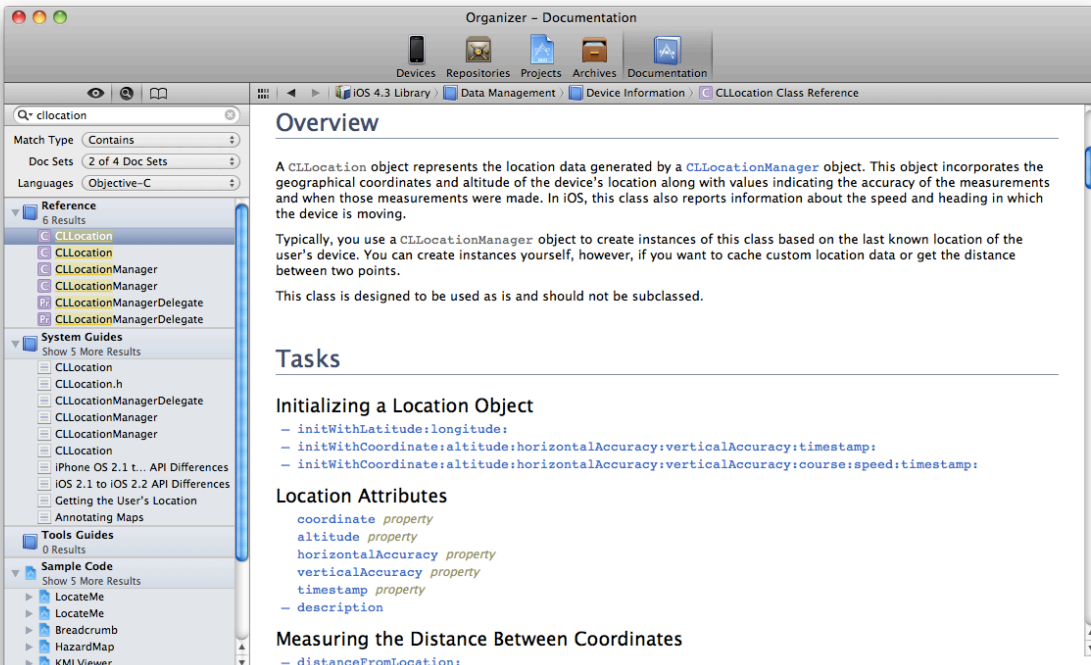


Fig. 4.3. Ajuda incorporada al Xcode 4.

No tota la família de dispositius iOS disposa de GPS, pel que queda descartada la família de iPod i iPad i únicament es desenvoluparà la versió per iPhone. Donat que existeixen 4 models diferents d'iPhone al mercat, i la primera versió no té GPS, el desenvolupament es centrarà en iPhone 3G, iPhone 3GS i iPhone 4.

L'aplicació es desenvoluparà utilitzant l'última versió disponible de iOS, que en aquest moment és la 4.3.

## **4.2. Anàlisi dels casos d'ús**

El cas d'ús principal d'aquest tipus d'aplicacions és enregistrar la sessió. Aquest cas enregistrarà els canvis de posició i analitzarà les dades per oferir informació sobre les calories cremades, velocitat actual i mitjana de la sessió, temps que porta la sessió activa, distància recorreguda, minuts per kilòmetre, mitjana de minuts per kilòmetre i quin és l'esport que està configurat a les preferències.

Un altre cas d'ús ha de permetre gestionar les sessions enregistrades, mostrant una llista amb les sessions i en quina data i hora es van enregistrar. Al la pantalla del detall s'han de visualitzar les mateixes dades que ofereix el cas d'ús principal, tret de la velocitat i minuts per kilòmetre actuals. Com a cas d'ús addicional, es pot mostrar un mapa per mostrar el recorregut enregistrat. Aquest mapa ha de permetre escollir el tipus de visualització, permetent escollir entre vista de satèl·lit, vista de mapa i vista híbrida, i ha de permetre centrar la vista a la zona del recorregut.

Un altre cas d'ús ha d'oferir informació sobre l'índex de massa corporal, usant la informació que l'usuari hagi configurat a les preferències. La informació que ha de mostrar és el sexe, la complexió, l'alçada, el pes, l'índex de massa corporal i el pes ideal.

L'últim cas d'ús ha de permetre configurar les preferències de l'aplicació.

## **4.3. Estudi del desenvolupament de cada cas d'ús.**

Cada cas d'ús necessita un estudi per veure com es pot implementar, analitzant les eines que ofereix Apple.

### 4.3.1. Enregistrar la sessió.

Apple ofereix eines per detectar els canvis de posició a través del *framework Core Location*. Aquest *framework* ofereix informació a la classe que implementi el protocol *CLLocationManagerDelegate* i disposa de dos mètodes diferents per detectar el canvi.

El primer sistema enregistra els canvis de posició que compleixin les condicions que es configuren, com poden ser la precisió mínima o la distància mínima. Per tal de saber quins paràmetres configurar, s'ha desenvolupat una aplicació de prova i s'han analitzat els resultats utilitzant cada una de les opcions que permeten els paràmetres. La conclusió de l'estudi és que les opcions que venen per defecte són les més apropiades ja que les altres opcions permeten enregistrar canvis de posició molt grans de forma que es pot estalviar energia, però no perceben de forma correcta els canvis de posició quan la velocitat és petita.

El segon sistema enregistra els canvis de posició quan el dispositiu entra en una zona definida. Aquest mètode queda descartat ja que no es pot definir una zona en aquest tipus de projectes.

Un cop enregistrat el canvi de posició s'han de realitzar els càlculs pertinents per aconseguir la informació necessària. El canvi de posició informa de les coordenades, l'alçada, precisió horitzontal i vertical, data i velocitat. Al analitzar les dades obtingudes en diferents sessions, s'ha detectat que la velocitat no ofereix informació real, ja que només informa de la velocitat quan el canvi de posició és molt ràpid, com és el cas d'anar en un vehicle. Com la velocitat és un paràmetre necessari, cal fer un estudi per tal de determinar la millor forma d'aconseguir el valor. La notificació del canvi de posició informa de quin és el punt anterior respecte la posició actual, per tant es pot calcular la velocitat dividint la distància entre els dos punts entre la diferència de distància entre ells.

Aquesta solució ofereix una informació més acurada de la velocitat, però la distància entre els punts no sempre s'actualitza amb la mateixa regularitat, pel que hi ha ocasions en les que es rep una notificació de canvi de posició i durant els següents 20 segons es reben notificacions de canvi de posició amb les mateixes coordenades (pel que el resultat de la divisió és 0) ; passats els 20 segons es rep una notificació amb una localització a més de 200 metres de distància de l'última, pel que el càlcul utilitzant l'última posició és que s'han

recorregut 200 metres en un segon. Per tal d'evitar aquest inconvenient, s'enregistrarà l'última posició que interessa i no s'utilitzarà la que ofereix la notificació. Utilitzant el mateix exemple, al primer segon s'enregistra la posició com a última interessant, i les notificacions dels propers 20 segons sense canvis de posició no s'enregistren, de forma que quan arriba la notificació del canvi de posició de 200 metres respecte l'última posició interessant, el temps és de 20 segons i el resultat del càlcul és de 10 m/s.

Un altre aspecte a tenir en compte és la falta de cobertura del sistema GPS. *Core Location* ofereix les notificacions del canvi de posició amb tots els sistemes disponibles, per tant no utilitza només el GPS, sinó que utilitza a-GPS i posicionament per IP. El sistema a-GPS assisteix la informació que ofereix el GPS triangulant la posició amb les antenes de telefonia properes, de forma que ràpidament posiciona un punt amb una precisió baixa, fins que la senyal de GPS permet millorar la precisió. El posicionament per IP intenta posicionar el terminal mitjançant la IP en el cas que estigui connectat a Internet. Aquest sistema interfereix en el càlcul de la velocitat tal i com han demostrat les proves, ja que en cas d'estar connectat a Internet, la primera posició que ofereix el sistema és el de la centraleta de telefonia, i a continuació el canvi de posició és el que ofereix el GPS, pel que la interpretació de les dades és d'un canvi de posició de més de 2 km en un segon. Aquest cas només es dona a l'inici de la sessió, ja que és quan el sistema de posicionament encara està cercant els satèl·lits del GPS. Per evita-ho, i tenint en compte que per tal de poder començar a realitzar els càlculs es necessiten dos punts, s'elimina la última posició interessant si la distància dels dos punts inicials és superior als 100 metres i es torna a començar el procés per aconseguir els dos punts necessaris.

Per últim, al fer els anàlisis s'ha detectat que els canvis de cobertura influeixen a la lectura de la velocitat tot i aplicant les mesures explicades. Per tal de millorar el sistema i tenint en compte que un ésser humà no és capaç de realitzar canvis de posició molt ràpids a no ser que es desplacin en un vehicle, s'ha limitat la velocitat màxima i s'ha implementat un sistema que enregistra les últimes 6 velocitats i calcula la mitjana, de forma que el canvi de velocitat és més regular. Degut a la combinació dels dos sistemes anteriors, resulta impossible saber quan s'atura l'usuari, ja que al aturar-se no canvia de posició, i per tant no s'enregistra cap canvi. Per detectar-ho, l'aplicació afegeix velocitats 0 a la llista de velocitats quan han passat 6 segons sense cap canvi, de forma que quan han passat 12 segons la mitjana de les velocitats és 0. Per tal que la velocitat sigui correcte, s'han

configurat velocitats màximes per cada esport, que són 45 km/h en cas de córrer i 130 km/h en cas de la bicicleta.

Per calcular les calories cremades s'utilitza la velocitat mitjana de la sessió i les dades de les taules 2 i 3 depenent de l'esport configurat. Calcular les calories cremades d'aquesta forma té in inconvenient a primera vista, i és que si l'usuari corre a 30 km/h durant 1 minut i s'està 1 minut parat abans d'aturar la sessió, el càlcul es realitzarà com si hagués corregut 2 minuts a 15 km/h. S'han fet proves utilitzant un sistema incremental enlloc d'utilitzar la mitjana, però els resultats obtinguts no han estat correctes amb comparació amb altres aplicacions, de forma que el primer sistema ha demostrat ser el més correcte.

Com les dades obtingudes requereixen conversió de mètrica, s'ha incorporat una preferència per poder escollir entre kilòmetres o milles.

La sessió s'enregistrerà en format serialitzable, contenint totes les localitzacions enregistrades i les dades bàsiques necessàries per tal de poder tornar a realitzar els càlculs quan es carregui de nou. Es desarà al directori de documents habilitat dins del paquet de l'aplicació.

#### **4.3.2. Historial de sessions.**

Per tal de mostrar una llista amb totes les sessions enregistrades, es mostra un llistat de tots els arxius del directori documents del paquet de l'aplicació. Per mostrar aquest llistat es carreguen les sessions i es mostra la data en què es van realitzar. Cocoa no permet des-serialitzar un arxiu de forma selectiva, sinó que el carrega sencer. Aquest fet es veu penalitzat per la velocitat lenta del processador, de forma que carregar cada una de les sessions amb totes les localitzacions (s'enregistra una localització per segon aproximadament) és molt lent. Per evitar aquest temps de càrrega, les sessions s'enregistren en un arxiu on el nom és la data i hora del moment (*timestamp*). D'aquesta forma es pot mostrar la data i hora del moment d'enregistrament de la sessió sense haver de carregar l'arxiu.

Per eliminar la sessió, s'ha seguit l'estil d'Apple, i enlloc de mostrar un botó, cal desplaçar horitzontalment el dit sobre l'ítem de la llista que es vol eliminar. Al fer-ho apareix un botó per eliminar la sessió.

Al seleccionar una fila de la llista es navega a la següent secció on es mostra el resum de la sessió. En aquest moment és on es des-serialitza tot l'arxiu i es carrega la sessió que torna a realitzar els càlculs per mostrar el resum. Aquest formulari té un botó que permet navegar a la següent secció on es dibuixa un polígon amb totes les localitzacions en un mapa.

Per tal de facilitar la visió del recorregut, s'analitzen quins són els punts extrems del recorregut, i es centra el mapa per veure el recorregut total.

### 4.3.3. Càlcul de l'índex de massa corporal.

En aquesta secció es calcula quin és el pes ideal segons l'alçada, i la complexió física configurades i indica el valor actual segons el pes configurat.

L'índex de massa corporal es calcula multiplicant el pes en kg i l'alçada al quadrat en metres. El pes ideal es calcula multiplicant l'alçada en metres al quadrat i un factor.

Els càlculs es realitzen tenint en compte les xifres de les taules Taula 4.1. i les descripcions es mostren segons el valor de l'índex de massa corporal calculat tal i com s'indica a la Taula 4.2.

	Complexió petita	Complexió Mitjana	Complexió Robusta
Home	23	24,3	26,4
Dona	21,4	22,8	25

Taula 4.1. Factors de multiplicació pel pes ideal.

Valor IMC	Descripció
< 18,5	Pes baix
18,5 - 25	Normal
25 - 30	Sobrepès
30 - 35	Obesitat risc alt
35 - 40	Obesitat risc molt alt
> 40	Obesitat risc extrem

Taula 4.2. Descripcions segons el valor de l'IMC.

#### 4.3.4. Preferències.

Per tal de poder configurar els paràmetres necessaris s'ha creat un apartat de preferències dins de la secció d'ajustos generals del terminal. Aquest sistema permet escollir quines preferències s'han de mostrar i quins són els valors que l'aplicació ha de gestionar per tal de comprendre-les. Totes les preferències han estat definides excepte les d'alçada i pes, per les que s'ha optat per mostrar un teclat numèric. Aquest teclat no permet la introducció de decimals pel que cal indicar un pes aproximat en Kg i els centímetres de l'alçada.

Com hi ha càlculs que depenen d'aquestes preferències, s'ha desenvolupat un sistema que comprova que les dades introduïdes siguin correctes, de forma que l'aplicació mostrarà una pantalla on s'explica com arribar a les preferències. Aquesta pantalla desapareixerà al tornar de les preferències si totes són correctes.

#### 4.4. Desenvolupament de l'aplicació.

La implementació dels casos d'ús es pot realitzar de manera independent, de forma que es poden tractar com mòduls separats, però és necessari un entorn on l'usuari pugui navegar entre tots els casos d'ús.

Per seguir les línies d'estil de les aplicacions d'Apple, s'utilitza un sistema de pestanyes per navegar entre ells. Aquestes pestanyes contenen els títols i les icones corresponents.





Fig. 4.4. Pantalla de sessió actual, llistat de sessions i índex de massa corporal.

Totes les imatges de l'aplicació es troben en format PNG (Portable Network Graphics), i es troben en dues resolucions. Això és degut a que la resolució de la pantalla de l'iPhone 4 és el doble que la de l'iPhone 3G o 3GS. Apple ofereix un sistema per no haver de gestionar el canvi de resolució i es tracta de fer la mateixa imatge en la resolució desitjada i un altre amb el doble de resolució, on els noms han de ser els mateixos i la gran ha d'anar acabada amb @2x.

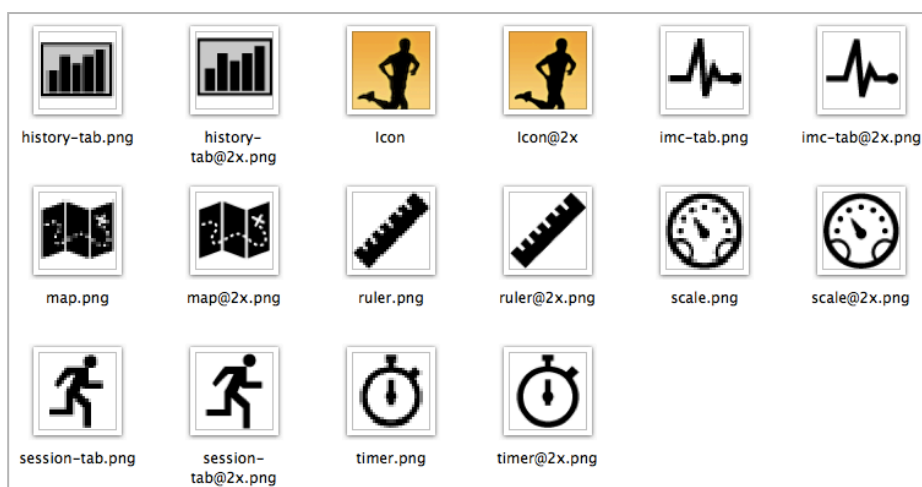


Fig. 4.5. Detall del nom de les imatges amb dues resolucions diferents.

Com el sistema de posicionament és lent fins aconseguir una precisió usable, s'inicia al obrir l'aplicació, però només enregistren les dades al iniciar una sessió.

Per tal de poder usar l'aplicació de forma agradable, s'ha implementat un sistema que evita que la pantalla s'apagui quan hi ha una sessió activa, i s'ha configurat de forma que pugui córrer en segon pla, permetent anar enregistrant la sessió alhora que es fan altres tasques amb el terminal.

La icona s'ha realitzat amb el format especificat per Apple (512px x 512px) amb un fons degradat taronja per fer referència al Tecnocampus.



Fig. 4.6. Icona de l'aplicació.

## 4.5. Classes.

El codi està distribuït en diferents classes.

### 4.5.1. `ProjecteAppDelegate`.

Hereta de *NSObject* i implementa els protocols *UIApplicationDelegate* i *UITabBarControllerDelegate*. És la classe delegada de la gestió de l'aplicació i de la gestió del menú de pestanyes inferior.

Quan l'aplicació torna del segon pla o s'inicia per primer cop, carrega les preferències i comprova que siguin les correctes. En cas negatiu mostra un missatge indicant que cal emplenar totes les preferències per tal que l'aplicació funcioni correctament. També indica al controlador de la vista corresponent a la pestanya que es carregarà, que està apunt d'aparèixer.

Per últim crea la instància de `LocationManager`.



Fig. 4.7. Detall de la pantalla que indica que les preferències s'han d'omplir.

#### 4.5.2. `SessionViewController`.

Hereta de `UIViewController`. Dóna la funcionalitat a la pantalla de sessió, controlant les dades que es visualitzen per pantalla i gestionant el botó d'iniciar o parar la sessió.

Quan s'inicia la sessió, desactiva el temporitzador que posa la pantalla en repòs i en crea un que es dispara cada segon per tal d'actualitzar la informació que es veu per pantalla. També indica al singletó `LocationManager` que comenci a enregistrar una sessió.

Consulta al singletó `Preferences` per tal de mostrar els camps en la mètrica configurada a les preferències.



Fig. 4.8. Sessió actual amb les mètriques configurades a les preferències.

### 4.5.3. HistoryViewController.

Hereta de *UIViewController* i implementa el protocol *UINavigationControllerDelegate*. Configura com es realitzarà la navegació al fer clic als ítems de la llista i mostra o amaga la barra de navegació depenent de si es mostra el primer controlador. Afegeix com a primer controlador de la navegació al *HistoryViewListController*.

### 4.5.4. HistoryViewListController.

Hereta de *UIViewController* i implementa els protocols de *UITableViewDataSource* i *UITableViewDelegate*. Al aparèixer, configura l'estil de la taula de la que n'és delegat i carrega un llistat de tots els arxius que es troben al directori de documents de l'aplicació, que tenen extensió *.plist* i que tenen una longitud igual o superior als 14 caràcters. Aquest filtre es realitza amb un predicat.

Com a delegat de la taula, gestiona la selecció dels ítems creant una instància de *HistoryDetailViewController*, l'afegeix a la navegació i li indica quin és el nom de l'arxiu de la sessió que ha de carregar. També indica quin text posar al desplaçar horitzontalment el dit sobre un ítem de la llista, tal com es pot veure a la Fig. 4.9.

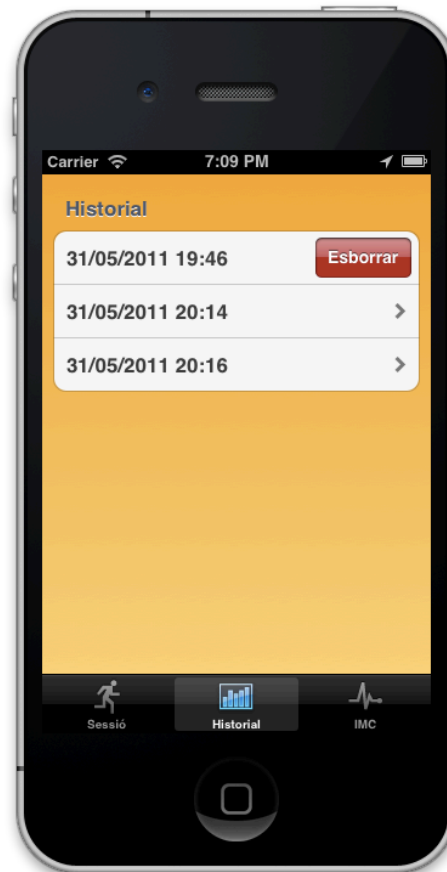


Fig. 4.9. Detall del text d'esborrar.

Com a font de dades de la taula, implementa un mètode per saber quants ítems conté la taula, i un altre que retorna l'ítem d'una posició en concret.

Per retornar la cel·la d'una posició concreta, es crea una i s'aprofita per cada posició. Aquesta cel·la es crea a partir d'una *UITableViewCellStyleDefault* que ja està definida i que conté un camp de text que s'emplenarà amb la data de la sessió. La data de la sessió s'obté del nom de l'arxiu on es troba enregistrada.

En cas d'eliminar la sessió s'esborra l'arxiu corresponent i es mostra un missatge en cas d'error.

#### 4.5.5. HistoryDetailViewController.

Hereta de *UIViewController*. Gestiona la informació que es veu per pantalla i carrega l'arxiu de sessió que li indica el *HistoryViewListController*. Al veure la ruta, crea una instància de *HistoryMapViewController*, li indica quines localitzacions ha de carregar i l'afegeix a la navegació.

Consulta al singletó *Preferences* per tal de mostrar els camps en la mètrica configurada a les preferències.



Fig. 4.10. Detall de la sessió amb les mètriques configurades a les preferències.

#### 4.5.6. HistoryMapViewController.

Hereta de *UIViewController* i implementa el protocol *MKMapViewDelegate*. Crea una taula de *CLLocationCoordinate2D* amb les coordenades de la sessió per tal de crear un

*MKPolyline* i mostrar-lo per sobre del mapa. També analitza quins són els marges del recorregut per quadrar la vista i no haver de fer zoom per veure-la completa.

Gestiona els botons de mètodes de visualització del mapa i el botó per centrar el mapa a la posició original.



Fig. 4.11. Visualització del recorregut amb els tres estils disponibles.

#### 4.5.7. IMCViewController.

Hereta de *UIViewController*. Calcula l'índex de massa corporal i el pes ideal a partir de les dades configurades a les preferències i mostra la informació per pantalla.

Consulta al singletó *Preferences* per tal de mostrar els camps en la mètrica configurada a les preferències i poder fer els càlculs amb la mètrica adient.



Fig. 4.12. Índex de massa corporal amb diferents mètriques configurades.

#### 4.5.8. LocationManager.

Hereta de *NSObject* i implementa el protocol *CLLocationManagerDelegate*. És singletona pel que només pot existir una instància en temps d'execució. Al crear-se la instància crea un *CLLocationManager* i se'n fa el delegat.

Rep les coordenades de les localitzacions que el dispositiu detecta i les filtra de forma que només gestiona les que tenen menys de 5 segons de demora i la seva precisió horitzontal no és negativa.

Quan l'objecte *SessionViewController* li indica que ha d'enregistrar una sessió, crea un objecte *Session* i li va afegint localitzacions a mesura que li van arribant. Quan ha de deixar d'enregistrar-la, li indica que es desi i resta activa per tal d'enregistrar-ne una de nova.



### 4.5.9. Session.

Hereta de *NSObject*. Conté una taula amb totes les localitzacions del recorregut i analitza les dades per tal d'aconseguir la velocitat, el temps, la distància, la velocitat mitjana, la velocitat màxima, els minuts per kilòmetre o per milla, la mitjana de minuts per kilòmetre o per milla i les kilocalories cremades.

És capaç de crear un arxiu amb les dades necessàries per poder carregar-les i poder calcular els valors que es mostren al resum de la sessió.

Al instanciar-se crea un temporitzador que es dispara un cop per segon i que serveix per gestionar la velocitat.

### 4.5.10. GradientView.

Hereta de *UIView*. Aquesta classe genera un degradat a la vista que l'instanciï. Totes les vistes de l'aplicació l'instancien per tal de mostrar el degradat taronja.

### 4.5.10. Preferences.

Hereta de *NSObject*. És singletona pel que només pot existir una instància en temps d'execució. Té un mètode per comprovar que les preferències s'han configurat correctament. També retorna l'adreça del directori on es troben les sessions i té mètodes que retornen els textos de les mètriques configurades per tal de mostrar-les per pantalla.

## 4.6. Altres arxius.

Hi ha arxius que no són classes però que són importants per l'aplicació.

### 4.6.1. Preferències.

Les preferències es troben juntament a les del sistema, gràcies a un paquet que incorpora l'aplicació. Aquest paquet es diu *Settings.bundle* i conté un arxiu *Root.plist*.

Xcode 4 permet editar aquest arxiu de forma visual, sense haver de mirar la documentació per tal de saber quines claus s'han d'especificar. Les preferències es guarden com enters,

de forma que és la classe *Preferences* la que fa la conversió per tal de poder treballar còmodament. També es pot especificar el tipus de teclat en els camps d'entrada de text, però no existeix un teclat numèric que permeti decimals, per això el text especifica que sigui “aproximat” tal com es pot veure a la Fig. 4.13.



Fig. 4.13. Detall del teclat numèric sense decimals a les preferències.

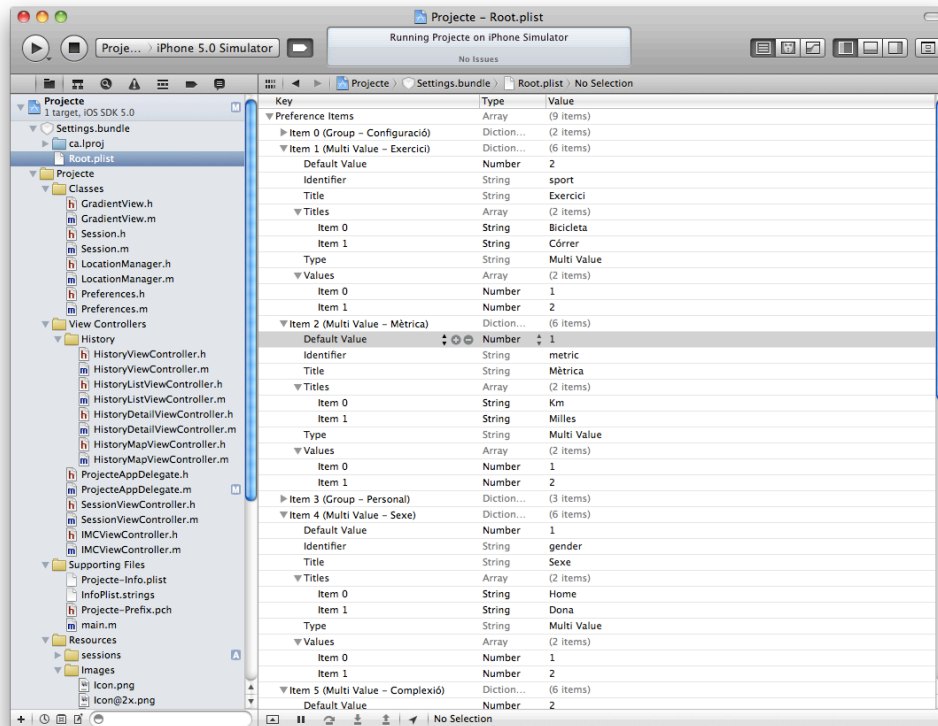


Fig. 4.14. Editor de l'arxiu de preferències integrat a Xcode 4.

```

1 |<?xml version="1.0" encoding="UTF-8"?>
2 |<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 |<plist version="1.0">
4 |<dict>
5 |   <key>PreferenceSpecifiers</key>
6 |   <array>
7 |     <dict>
8 |       <key>Title</key>
9 |       <string>Configuració</string>
10 |      <key>Type</key>
11 |      <string>P5GroupSpecifier</string>
12 |     </dict>
13 |     <dict>
14 |       <key>DefaultValue</key>
15 |       <integer>2</integer>
16 |       <key>Key</key>
17 |       <string>sport</string>
18 |       <key>Title</key>
19 |       <string>Exercici</string>
20 |       <key>Titles</key>
21 |       <array>
22 |         <string>Bicicleta</string>
23 |         <string>Córrer</string>
24 |       </array>
25 |       <key>Type</key>
26 |       <string>P5MultiValueSpecifier</string>
27 |       <key>Values</key>
28 |       <array>
29 |         <integer>1</integer>
30 |         <integer>2</integer>
31 |       </array>
32 |     </dict>
33 |     <dict>
34 |       <key>DefaultValue</key>
35 |       <integer>1</integer>
36 |       <key>Key</key>
37 |       <string>metric</string>
38 |       <key>Title</key>
39 |       <string>Mètrica</string>
40 |       <key>Titles</key>
41 |       <array>
42 |         <string>Km</string>
43 |         <string>Milles</string>
44 |       </array>
45 |       <key>Type</key>
46 |       <string>P5MultiValueSpecifier</string>
47 |       <key>Values</key>
48 |       <array>
49 |         <integer>1</integer>
50 |         <integer>2</integer>
51 |       </array>
52 |     </dict>

```

Fig. 4.15. Arxiu de preferències visualitzat amb un editor de text extern a Xcode 4.

### 4.6.2. Projecte-Info.plist.

Aquest arxiu especifica les opcions de llançament de l'aplicació i altres opcions pel sistema del dispositiu.

Igual que l'arxiu de preferències, aquest arxiu es pot editar de forma visual utilitzant les eines que incorpora Xcode 4. Per aquesta aplicació s'ha especificat que la barra superior on s'indica l'hora, cobertura i bateria sigui negra des del llançament, que la icona ja té efecte brillant, que l'idioma de desenvolupament és el català, que el nom de l'aplicació és RunSensor i que l'aplicació corre en segon pla enregistrant canvis de localització.

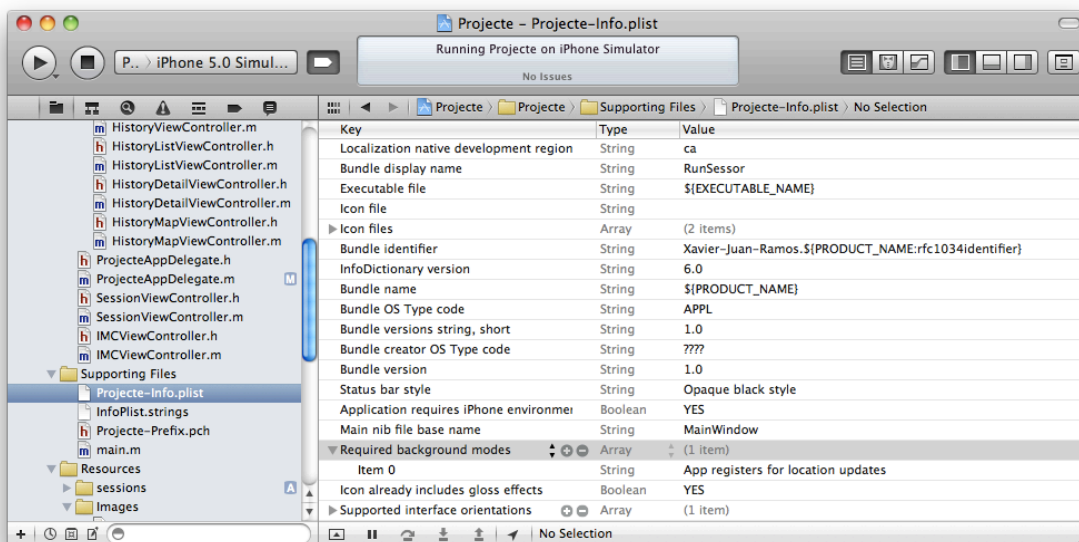


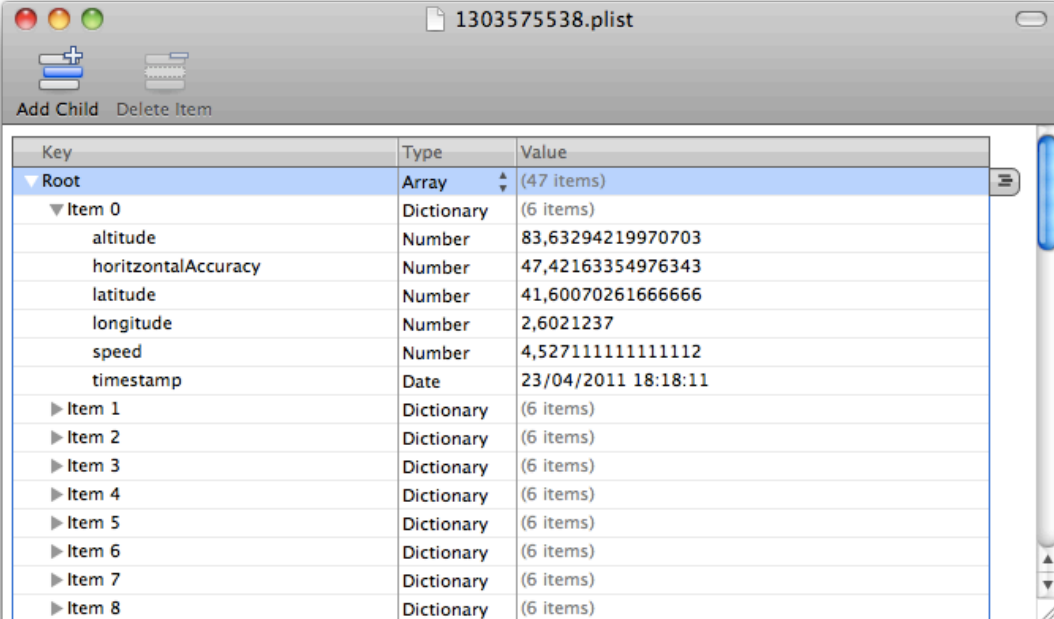
Fig. 4.16. Arxiu d'opcions visualitzat amb Xcode 4.

### 4.6.3 Projecte-Prefix.pch.

Aquest arxiu és l'arxiu capçalera de tots els arxius del projecte. En aquest cas s'ha afegit un *include* per tal d'incloure la definició del *framework CoreLocation*.

## 5. Proves de l'aplicació.

Per tal de provar l'aplicació s'han anat fent proves durant el desenvolupament. Per analitzar les dades, s'ha fet que la sessió enregistri els valors que es mostren per pantalla de forma que el propi arxiu conté el registre de tot el que ha calculat l'objecte *Session*. Com la sessió s'enregistra en un arxiu .plist, es pot visualitzar amb l'eina que incorpora Xcode 4.



Key	Type	Value
Root	Array	(47 items)
Item 0	Dictionary	(6 items)
altitude	Number	83,63294219970703
horizontalAccuracy	Number	47,42163354976343
latitude	Number	41,60070261666666
longitude	Number	2,6021237
speed	Number	4,527111111111112
timestamp	Date	23/04/2011 18:18:11
Item 1	Dictionary	(6 items)
Item 2	Dictionary	(6 items)
Item 3	Dictionary	(6 items)
Item 4	Dictionary	(6 items)
Item 5	Dictionary	(6 items)
Item 6	Dictionary	(6 items)
Item 7	Dictionary	(6 items)
Item 8	Dictionary	(6 items)

Fig. 5.1. Primera sessió de prova de 47 segons.

La primera sessió va permetre veure quines dades es podien aconseguir de l'objecte *CLLocation* que arriba a la classe *LocationManager*.

S'han realitzat sessions d'unes poques localitzacions per tal de comprovar si els càlculs eren correctes, i d'altres llargues per tal d'analitzar una sessió real. En aquestes sessions s'enregistraven tots els camps possibles.

The screenshot shows a window titled '1303802217.plist' with a table of keys and values. The table has three columns: Key, Type, and Value. The keys include 'Item 704' through 'Item 712', and various metrics like 'altitude', 'calculated\_avgspeed', 'calculated\_distance', 'calculated\_speed', 'horizontalAccuracy', 'latitude', 'longitude', 'screen\_distance', 'screen\_minKm', 'screen\_minKmAvg', 'screen\_speed', 'screen\_speedAvg', 'screen\_speedMax', 'speed', and 'timestamp'.

Key	Type	Value
Item 704	Dictionary	(15 items)
Item 705	Dictionary	(15 items)
Item 706	Dictionary	(15 items)
Item 707	Dictionary	(15 items)
Item 708	Dictionary	(15 items)
Item 709	Dictionary	(15 items)
Item 710	Dictionary	(15 items)
Item 711	Dictionary	(15 items)
Item 712	Dictionary	(15 items)
altitude	Number	49,60695266723633
calculated_avgspeed	Number	11,02930361478644
calculated_distance	Number	7852,951514932172
calculated_speed	Number	5,320103539015294
horizontalAccuracy	Number	9,112091691513054
latitude	Number	41,57913123333333
longitude	Number	2,546027466666667
screen_distance	String	7.85
screen_minKm	String	0.00
screen_minKmAvg	String	-0.66
screen_speed	String	0.00
screen_speedAvg	String	-39.77
screen_speedMax	String	76.12
speed	Number	0
timestamp	Date	26/04/2011 09:16:57

Fig. 5.2. Sessió de 712 segons i 7,8 kilòmetres.

Finalment, el format de les dades enregistrades va canviar per tal de poder carregar totes les dades necessàries per poder fer tots els càlculs, i un camp de localitzacions que registra la taula de localitzacions.

The screenshot shows a window titled '1306773070.plist' with a table of keys and values. The table has three columns: Key, Type, and Value. The keys include 'Root', 'distance', 'kcal', 'locations', 'minKmAvg', 'speedAvg', 'speedMax', 'sport', and 'time'.

Key	Type	Value
Root	Dictionary	(8 items)
distance	String	2.44
kcal	String	181.24
locations	Array	(381 items)
minKmAvg	String	0.09
speedAvg	String	5.18
speedMax	String	12.32
sport	Number	2
time	String	00:28:15

Fig. 5.3. Sessió de 28 minuts i 2,44 kilòmetres.

## 6. Característiques de Cocoa i de Objective-C.

Cocoa i Cocoa Touch consisteixen en llibreries i APIs que permeten programar per les plataformes d'Apple. Utilitzen el patró de disseny Model-Vista-Controlador i s'utilitzen amb el llenguatge Objective-C. Els *frameworks* que ofereixen són els que es poden veure a la Fig. 6.1. i la Fig. 6.2.

<b>Audio and Video</b>	<b>Graphics and Animation</b>	<b>Cocoa Bridges to Scripting Languages</b>
Core Audio	Core Animation	AppleScript
Core MIDI	Core Image	Python
Core Video	OpenGL	Ruby
	Quartz	
<b>Data Management</b>	QuickTime	<b>User Applications</b>
Core Data	UIKit	Address Book
		Calendar Store
<b>Networking and Internet</b>		Instant Message
Bonjour		
Directory Services		
Kerberos		

Fig. 6.1. *Frameworks* que ofereix Cocoa.

<b>Audio and Video</b>	<b>Graphics and Animation</b>	<b>User Applications</b>
Core Audio	Core Animation	Address Book
OpenAL	OpenGL ES	Core Location
Media Library	Quartz 2D	Map Kit
AV Foundation		Store Kit
<b>Data Management</b>	<b>Networking and Internet</b>	
Core Data	Bonjour	
SQLite	WebKit	
	BSD Sockets	

Fig. 6.2. *Frameworks* que ofereix Cocoa-Touch.

En aquest cas, degut a que l'aplicació s'ha desenvolupat per iPhone, s'ha utilitzat Cocoa Touch i s'ha utilitzat el *framework* de *Map Kit* i *Core Location*.

Objective-C té particularitats que el diferencien d'altres llenguatges com el Java.

## 6.1. Estil.

L'estil de programació és el primer que es veu diferent al veure un codi programat en Java o en Objective-C. En Java les funcions tenen un nom i els paràmetres es defineixen entre parèntesi. Objective-C és molt més descriptiu, ja que el nom de les funcions inclou descripcions pels paràmetres. Per exemple per fer un mètode de classe que retorni un color i que se li hagi d'indicar per paràmetres la quantitat de vermell, verd, blau i el nivell de transparència.

En Java el mètode es defineix com:

```
public function Color colorWithRGB(double red, double green, double blue, double alpha);
```

I la crida és:

```
Color color = Color.colorWithRGB(1.0, 1.0, 1.0, 1.0);
```

En Objective-C el mètode es defineix com:

```
+ (UIColor *)colorWithRed:(CGFloat)red green:(CGFloat)green blue:(CGFloat)blue alpha:(CGFloat)alpha;
```

I la crida és:

```
UIColor *color = [UIColor colorWithRed:1.0 green:1.0 blue:1.0 alpha:1.0];
```

El codi de Objective-C és més llarg però més descriptiu. Les crides es realitzen entre claudàtors.

En Java els mètodes de classe es declaren com a *static*, en canvi en Objective-C es declaren amb un signe positiu (+) a l'inici del mètode. Si són d'instància es declaren amb un signe negatiu (-).

Per conveni, els mètodes de les classes de Cocoa que tenen diferents constructors comencen amb la paraula *initWith*. Els mètodes que retornin l'objecte instanciat com les factories comencen amb la mateixa paraula que la classe. Per exemple la classe *UIColor* pot tenir un constructor *initWithAlpha* o una factoria *colorWithAlpha*.



## 6.2. Gestió de la memòria.

En Java la gestió de la memòria la realitza un recol·lector d'objectes orfes. Cocoa permet la opció d'utilitzar un recol·lector però la gestió de memòria original, i amb la que s'ha desenvolupat aquest projecte, és la manual. La gestió d'objectes orfes es realitza de forma diferent a altres llenguatges. En Cocoa existeix un comptador intern del nombre de referències a un objecte, de forma que quan l'objecte no té cap referència s'elimina. Al crear un objecte el comptador s'incrementa en una unitat, i al fer *release* disminueix en una unitat, però no té perquè eliminar-se.

Els constructors retornen l'objecte amb un increment en el nombre de referències, pel que l'objecte que ha fet la crida al constructor ha de tenir la cura de disminuir el nombre de referències quan no necessiti l'objecte. Les factories retornen l'objecte amb un increment al comptador, però gestionen la disminució mitjançant el mètode *autorelease*. Aquest mètode assegura que l'objecte s'elimini quan ja s'ha executat el bloc de codi on es troba la crida.

## 6.3. Crides a objectes.

En Java no es poden fer crides a mètodes a punters nuls. Cocoa permet fer crides a punters nuls en temps d'execució sense que hi hagi cap inconvenient. Si el punter és nul (*nil* en Objective-C) la crida no es realitzarà mai. Si el punter senyala a un objecte instanciat, la crida al mètode es realitzarà tant si l'objecte implementa el mètode com si no.

En cas que l'objecte no implementi el mètode, es realitzarà una crida al seu pare. Si cap objecte de la jerarquia implementa el mètode es genera una excepció. Xcode 4 informa que l'objecte no implementa el mètode en temps d'edició, però ho indica informant que el retorn de la funció serà tractat com a *id*.

Cocoa és un llenguatge que gestiona els tipus de les variables, de forma que cada objecte té un tipus. L'objecte *id* és un punter, i per tant pot ser qualsevol tipus.



## 7. Llicència.

Apple permet desenvolupar aplicacions amb Xcode i executar-les sota el simulador incorporat amb les diferents versions del sistema operatiu iOS. Per tal de poder executar una aplicació al dispositiu o publicar-la a l'AppStore, és necessària una llicència que té un cost de 99\$ anuals.

Apple permet vincular fins a 100 dispositius amb una llicència, i Xcode 4 incorpora la gestió de dispositius, de manera que un cop vinculat s'ha de compilar l'aplicació amb la clau privada per tal que el dispositiu pugui descriptar l'aplicació amb la clau pública corresponent.



Fig. 7.1. Preferències de signatura de l'aplicació.

Per tal de poder distribuir l'aplicació a l'AppStore cal signar-la amb la clau dedicada a la publicació, i cal enviar l'aplicació a revisió de forma que Apple revisi que no s'utilitzen APIs no documentades. Un cop Apple ha revisat l'aplicació, es publica a la botiga.

Apple només permet l'ús de classes i mètodes que es trobin publicats a la documentació que ofereixen, de forma que, degut a la característica de Cocoa de poder fer crides a mètodes no especificats, es poden trobar mètodes amb funcionalitats no documentades.



## 8. Conclusions.

Apple s'ha convertit en l'empresa referent en quan a comerç de dispositiu mòbils gràcies al sistema de distribució. El fet d'integrar una botiga on trobar totes les aplicacions i que es puguin instal·lar amb un sol clic fa que sigui un mercat amb molt moviment i amb ingressos molt elevats si la idea és bona. S'han donat casos d'aplicacions que han tingut benefici de 2 milions de dòlars en un cap de setmana. Apple es queda amb el 30% de la venda de forma que li interessa que les aplicacions siguin bones.

En la botiga hi ha més de 350.000 aplicacions, el que també vol dir que hi ha molta competència. Aquest projecte ha estudiat altres solucions ja existents a la botiga, i tots els programes tenen més opcions, pel que el preu i temps de desenvolupament segurament han estat molt superiors. El preu estrella de les aplicacions és de 79 cèntims d'euro, pel que calen moltes vendes per aconseguir que el projecte sigui rentable. En un mercat tant actiu, cal innovar i oferir solucions que la competència no pugui oferir, de forma que aquesta aplicació aporta el càlcul de l'índex de massa corporal que les altres aplicacions no ofereixen.

Gràcies a les eines que Apple ofereix, el temps de desenvolupament no és elevat tenint en compte que gran part ha estat fer proves i provar diferents sistemes i possibles solucions. El fet de programar amb Cocoa i Xcode 4 fa que sigui molt còmode trobar l'ajuda integrada i exemples a la pàgina de suport.

Hi ha diferents aspectes que es poden millorar o afegir a l'aplicació. El càlcul de les calories cremades és un punt que requereix d'un estudi per tal d'analitzar si es poden tenir en compte paràmetres com l'alçada, el pes o la constitució física per tal d'aconseguir un factor més acotat a cada usuari. També es pot millorar la llista de les sessions, mostrant un resum a la pròpia llista.



## 9. Referències.

- [1] [www.vitonica.com/carrera/cuantas-calorias-quemo-corriendo-dos-formulas-para-calcularlas](http://www.vitonica.com/carrera/cuantas-calorias-quemo-corriendo-dos-formulas-para-calcularlas), Eunice, 27 de desembre de 2008
- [2] [www.altorendimiento.net/nutricion-y-gasto-energetico/calculador-del-gasto-energetico-2.html](http://www.altorendimiento.net/nutricion-y-gasto-energetico/calculador-del-gasto-energetico-2.html)
- [3] [www.nhlbi.nih.gov/guidelines/obesity/prctgd\\_b.pdf](http://www.nhlbi.nih.gov/guidelines/obesity/prctgd_b.pdf), NIH Publication No. 00-4084, Octubre 2000





# Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

**Enginyeria Tècnica en Informàtica de Gestió**

**ASSESSOR PER LA SALUT MÒBIL**

**Estudi econòmic**

**XAVIER JUAN RAMOS  
PONENT: LINA JUAN NADAL**

PRIMAVERA 2011



TecnoCampus  
Mataró-Maresme



## **Índex.**

1. Cost de desenvolupament. ....	1
1.1. Temps de dedicació. ....	1
1.2. Pressupost de mà d'obra. ....	1
2. Cost de producció. ....	3



## **1. Cost de desenvolupament.**

Ja que Xcode només es troba disponible per plataforma Mac OSX, cal un ordinador Apple, que en aquest cas ha estat un Macbook Pro. Per tal de poder fer proves en un entorn real és necessari un dispositiu mòbil, que en aquest cas ha estat un iPhone 3GS.

### **1.1. Temps de dedicació.**

El cost de desenvolupament no ha requerit aprenentatge del llenguatge ni de l'entorn en aquest projecte, ja que l'autor del projecte desenvolupa aplicacions en l'entorn d'Apple.

El temps d'anàlisi de les dades obtingudes ha estat el que més dedicació ha requerit, ja que el càlcul de les calories cremades depèn de la velocitat i la gestió de la velocitat ha estat la més complicada.

La disposició de les dades a la pantalla s'ha fet en base a altres aplicacions del mercat, pel que no s'ha estudiat l'accessibilitat. El disseny visual sí que ha estat pensat per ser semblat al del Tecnocampus. La recerca i adaptació de les icones també ha estat estudiada per tal de ser definitòries per si mateixes.

### **1.2. Pressupost de mà d'obra.**

Per tal de poder pressupostar el desenvolupament de l'aplicació, s'especifiquen preus per hora diferents depenent del tipus de dedicació. Aquest preu es multiplica per les hores de dedicació a la tasca i s'aconsegueix el subtotal de la tasca. El cost de mà d'obra resulta de la suma de tots els subtotals.

Tipus de dedicació	Tasca	Hores de dedicació	Subtotal
<b>Investigació i recerca</b> 30€ l'hora	Anàlisi de la competència	6	180€
	Recerca dels càlculs de les calories i IMC	12	360€
	Estudi de les dades disponibles	2	60€
	Estudi de la gestió de la velocitat	35	1.050€
	Recerca de les icones	3	90€
	Estudi de les conversions mètriques	4	120€
	<b>Total investigació</b>	<b>62</b>	<b>1.860€</b>
<b>Desenvolupament i proves</b> 60€ l'hora	Disseny de classes	6	360€
	Implementació del codi	40	2.400€
	Disseny de les interfícies gràfiques	6	360€
	Creació i adaptació de les icones	3	180€
	<b>Total desenvolupament</b>	<b>61</b>	<b>3.660€</b>
<b>Total</b>		<b>123</b>	<b>5.520€</b>

Pressupost de mà d'obra.

## 2. Cost de producció.

La imputació al projecte s'ha realitzat tenint en compte 12 projecte per any, i la vida útil de l'ordinador s'ha pensat en 7 anys i del dispositiu mòbil en 5 anys.

<b>Element</b>	<b>Cost</b>	<b>Imputació al projecte</b>
Investigació i recerca	1.860€	1.860€
Desenvolupament i proves	3.660€	3.660€
Llicència de desenvolupament	69€ al canvi	5,75€
Macbook Pro utilitzat	1.800€	21,42€
iPhone 3GS	650€	10.83€
<b>Total</b>	<b>8.039€</b>	<b>5.558€</b>

Cost de producció.





# Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

**Enginyeria Tècnica en Informàtica de Gestió**

**ASSESSOR PER LA SALUT MÒBIL**

**Annexos**

**XAVIER JUAN RAMOS  
PONENT: LINA JUAN NADAL**

PRIMAVERA 2011



**TecnoCampus  
Mataró-Maresme**



## **Índex.**

Annex I. Contingut del CD-ROM. ....	1
-------------------------------------	---



## **Annex I. Contingut del CD-ROM.**

- Documentació del projecte.
- Projecte de Xcode.
- Document psd amb la icona del programa.

