



Anàlisi i disseny d'una aplicació gestora de curses via GPS per a dispositius mòbils amb SO Android

Salvador Triadó Ballús
Grau en Enginyeria Informàtica

Oriol Martí Girona

09/01/2015



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Anàlisi i disseny d'una aplicació gestora de curses via GPS per a dispositius mòbils amb SO Android.</i>
Nom de l'autor:	<i>Salvador Triadó Ballús</i>
Nom del consultor:	<i>Oriol Martí Girona</i>
Data de lliurament (mm/aaaa):	<i>01/2015</i>
Àrea del Treball Final:	<i>Enginyeria del Programari</i>
Titulació:	<i>Grau en Enginyeria Informàtica</i>
Resum del Treball (màxim 250 paraules):	
<p>Aquest Treball consisteix a analitzar la viabilitat d'una aplicació gestora de curses via GPS per a dispositius Android i, en cas afirmatiu, desenvolupar-ne un prototip. Per tal d'aconseguir-ho, es passen per una sèrie de fases d'estudi de l'aplicació i d'aprenentatge en l'implementació</p> <p>En primer lloc, es fa un estudi sobre la necessitat que hi pot haver en introduir una aplicació gestora de curses via GPS per a dispositius Android. Un cop s'ha justificat la possible existència d'aquesta aplicació, es fa un paral·lelisme amb el disseny de l'aplicació i els requisits que pot tenir. Així i afegint una valoració dels costos de cada requisit es demostra la viabilitat d'aquesta aplicació.</p> <p>En una segona fase el Treball consisteix en assolir un aprenentatge suficient per desenvolupar un prototip de l'aplicació estudiada. Aquest prototip es desenvolupa seguint una premissa: la qualitat del disseny per sobre de la quantitat de codi. A més, també es desenvolupa tenint en compte els requisits obtinguts a la primera fase i respectant el disseny teòric obtingut fruit de l'estudi del sistema. Amb això es pretén que no sigui necessari rectificar el codi escrit de cara a futures ampliacions del sistema.</p> <p>Per acabar, es detallen una sèrie de conclusions: sobre el disseny del prototip i l'aprenentatge assolit, sobre el sistema estudiat i un comentari com a projectista.</p>	

Abstract (in English, 250 words or less):

This Project consists in analyse the viability of an Android application which manages races using the cellular's GPS and, if affirmative, develop a prototype. With this aim, the Project goes across some application study phases and a implementation learning phase.

First of all, there is a study over the possible necessity in introducing this application for Android devices. When its existence is justified, there is a parallelism between the application design and its requirements. In this way, and estimating the possible charge of each requirement it is demonstrated the viability of this application.

In the second phase the Project consists in acquire enough learning in java development for Android applications to develop a prototype for the studied application. This prototype is developed under a premise: quality design is over the quantity of code. Furthermore, the development bears in mind the requirements obtained at first phase and respecting the theoretical design obtained from studying the system. With these constraints the aim is avoiding possible code rectifications in futures system extensions.

For finishing, there are detailed a set of conclusions: over the design of the prototype and the acquired learning, over the studied system and a comment as designer.

Paraules clau (entre 4 i 8):

Android, race, appliction, requirements specifications, use case

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball.....	5
1.3 Enfocament i mètode seguit	5
1.4 Planificació del Treball.....	8
1.5 Breu sumari de productes obtinguts	10
1.6 Breu descripció dels altres capítols de la memòria	10
2. Anàlisi del sistema.....	12
2.1 Recollida de dades	12
2.2 Anàlisi de requisits	17
2.3 Anàlisi de CU.....	20
2.4 Introducció a la GUI.....	26
3. Disseny del sistema	29
3.1 Model conceptual	29
3.2 Disseny teòric del sistema.....	32
4. Implementació del prototip	39
4.1. Aprenentatge món Android.....	39
4.2 Entorn utilitzat.....	43
4.3 Aplicació de l'aprenentatge amb app de prova.....	44
4.4 Contextualització del prototip	47
4.5 Disseny del prototip	48
4.6 Mancances del prototip i visió de futur	51
5. Conclusions.....	54
5.1 Comentari com a projectista.....	55
6. Glossari	57
7. Bibliografia.....	58
8. Annexes	60

Llista de figures

Il·lustració 1: Cronòmetre	1
Il·lustració 2: App Strava	1
Il·lustració 3: Resultats Matagalls-Montserrat.....	3
Il·lustració 4: Resultats Transgavarres	3
Il·lustració 5: Resum circuit a RaceChrono	4
Il·lustració 6: Circuit dissenyat amb RaceChrono.....	4
Il·lustració 7: Planificació del Treball	9
Il·lustració 8: Imatge explicadora del model de Kano extreta de http://baymard.com/blog/kano-model	18
Il·lustració 9: Casos d'ús	22
Il·lustració 10: Pantalla d'índex.....	26
Il·lustració 11: Pantalla de presentació per inscrits	27
Il·lustració 12: Pantalla de presentació per l'administrador	27
Il·lustració 13: Pantalla d'estat de la cursa	28
Il·lustració 14: Diagrama entitat-relació	30
Il·lustració 15: Diagrama de Seqüència pel CU Crear Cursa	34
Il·lustració 16: Diagrama de Seqüència pel CU Iniciar Cursa.....	35
Il·lustració 17: Diagrama de Seqüència pel CU Finalitzar Cursa.....	36
Il·lustració 18: Cicle d'estats i vida d'una Activity.....	40
Il·lustració 19 : Esquema d'arxius.....	42
Il·lustració 20: ActivityMain	44
Il·lustració 21: ActivityOne	45
Il·lustració 22: ActivityTwo	45
Il·lustració 23: ActivityCoordinates	45
Il·lustració 24:ActivityMap.....	46
Il·lustració 25: ActivityMain	48
Il·lustració 26: CreateActivity	49
Il·lustració 27: CreateMapActivity	50
Il·lustració 28: DisplayRaceActivity.....	51

1. Introducció

1.1 Context i justificació del Treball

L'increment i evolució de les aplicacions dels dispositius mòbils ha fet que aquestes siguin cada vegada més presents a la vida quotidiana, tant a nivell de consulta, com a nivell social o fins i tot esportiu. Tot i això, dins el món de l'esport no és fàcil trobar una aplicació que sigui capaç de gestionar curses.

A dia d'avui es fan moltes curses, ja sigui a parcs motor o a recorreguts creats expressament per un esdeveniment concret.

Sempre que parlem de nivells *amateurs*, les curses encara es gestionen amb cronòmetres malgrat que només donen un temps individual per cada corredor. Tal i com es veurà a les entrevistes, aquesta eina s'utilitza sobretot a l'àmbit particular. A partir d'aquesta eina resulta molt complicat comparar diversos corredors en trams concrets de la cursa, així com l'obtenció de dades precises. D'aquesta manera, moltes marxes populars es gestionen sense tenir en compte el temps, com per exemple les de Sant Esteve de Palautordera i la de Santa Margarida. De totes maneres, un exemple de marxa gestionada amb cronòmetre és la de Sant Esteve de Llagostera (<http://cursasantesteve.wordpress.com/>), on es pot apreciar la manca d'exactitud de les dades preses.



Il·lustració 1: Cronòmetre



Il·lustració 2: App Strava

Per solucionar aquest problema l'aplicació més utilitzada és Strava, una aplicació esportiva que actua com a xarxa social, on es pot comparar el rendiment d'un usuari amb el d'altres usuaris. A més, també ofereix una divisió del recorregut per segments (trams de cursa) per comparar el rendiment a trams concrets. L'abast d'ús d'aquesta eina es podrà contrastar a les entrevistes, a fòrums com a www.forociclista.com i també a articles d'opinió, així com els comentaris als fòrums en relació a l'article, com es pot veure a pàgines com <http://montenbaik.com/2012/05/como-sacarle-el-maximo-provecho-a-strava/>, <http://www.velocidadabsurda.com/enganchado-al-strava/> i a <http://www.gadgetsparacorrer.com/2013/01/29/strava/>.

Tal i com es pot observar, Strava és una aplicació extensament utilitzada, tot i que no és l'única. De fet, tal i com es pot contrastar als links anteriors, recentment s'ha començat a popularitzar l'aplicació Endomondo, que també està molt ben valorada.

Tot i les crítiques favorables, gestionar curses amb aquestes eines no és adequat ja que no contempla la possibilitat de crear trams de competició per un temps determinat (sincronitzat) i per uns usuaris concrets, fent que el concepte de "cursa" no encaixi dins les seves possibilitats. També cal afegir que només està dissenyat per corredors i ciclistes, deixant de banda, a més, el món del motor.

Deixant de banda les competicions personals i les petites competicions, quan es tracta de maratons o qualsevol altre tipus de cursa llarga amb una afluència alta de participants, el que s'utilitza és un sistema de xips. Aquest sistema consisteix a col·locar un xip a cada competidor registrant el temps que es tarda des de la sortida fins a l'arribada. A més, també incorpora punts de control durant la cursa. Tot i que a primer cop d'ull sembla un sistema molt pràctic, té 2 inconvenients importants: el cost econòmic i el retard de publicació de resultats.

Tal i com es pot veure consultant el reglament de les curses més populars de Catalunya, el xip és el sistema més utilitzat. Això es pot corroborar consultant el reglament de curses com la Marató del Montseny (<http://maratomontseny.com/>) o la Transgabarres (<http://www.cursademuntanyadegirona.com/>). Consultant el reglament i la inscripció, es pot confirmar que els competidors han d'utilitzar el xip i a més el preu d'inscripció és bastant elevat.

Com a detall afegit, es pot consultar un exemple de resultat que ofereix el xip per algunes curses:

 **Buscador Mm14**

NOM O COG. :

GO 

 **Controls Mm14**

DORSAL

GO 

Informació Mm 14 Controls i Temps Total

2116 ~~XXXXXXXXXXXX~~

Sortida-0Km	Arribada-84,8Km
14 de setembre a les 17:09:13	15 de setembre a les 12:13:04

Control 1	Control 2	Control 3	Control 4	Control 5	Control 6	Control 7	Control 8	Control 9	Control 10
7,9Km	20,4Km	28,8Km	37Km	51,3Km	59,2Km	66,5Km	71,2Km	78,9Km	81,6Km
18:28:34	20:47:18	22:40:56	00:49:51	NO MARCA	05:49:17	07:44:57	08:50:47	10:33:13	11:14:25

II-lustració 3: Resultats Matagalls-Montserrat

tretze sports

TRANSGAVARRES CMG 2013

~~XXXXXXXXXXXX~~

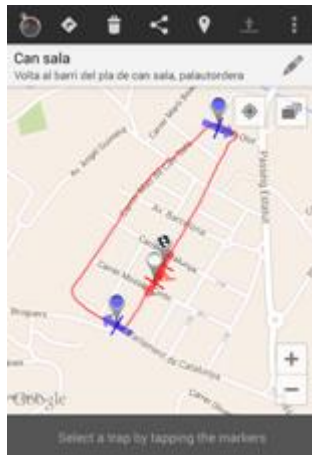
Dorsal:	14	Posició	
Club:	C.A. ROSES	2	Posició Parcial 1 :
Categoria:	ABSM	Posició Categoria	Temps Parcial 1 :
Sexe:	Home	2	Posició Parcial 2 :
Temps Oficial :	05:21:14	Posició Sexe	Temps Parcial 2 :
Temps Real :		2	Posició Parcial 3 :
Marca Km :	00:05:22		Temps Parcial 3 :

II-lustració 4: Resultats Transgavarres

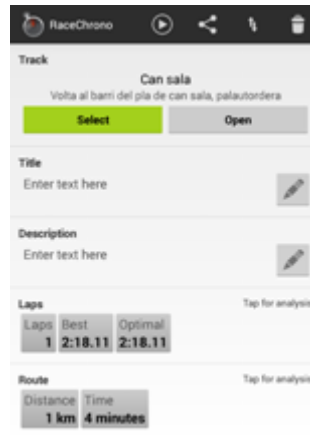
En aquests exemples es pot visualitzar:

1. La impossibilitat de confrontar diversos corredors.
2. La gran quantitat de quilòmetres sense control.

A part de les dificultats analitzades, cal afegir que quan es vol analitzar el rendiment a un tram de la cursa no només és important el temps de cada corredor, sinó que també és important saber quin corredor va davant i amb quin avantatge. Per gestionar aquest aspecte trobem un buit molt important a la Google Play Store.



Il·lustració 6: Circuit dissenyat amb RaceChrono



Il·lustració 5: Resum circuit a RaceChrono

Si focalitzem l'anàlisi del context a aquest buit de Google Play, trobem una aplicació de curses: RaceChrono. Aquesta s'acosta més al que busquem ja que permet dissenyar fàcilment recorreguts. Tot i això, la mancança principal d'aquesta aplicació és que està dissenyada per un ús individual, cosa que impossibilita fer curses contra tercers.

Per aquest motiu considero necessari crear una aplicació específicament dissenyada per gestionar curses d'àmbit genèric, englobant el món del motor, el ciclisme i les curses a peu, i en la qual puguin competir un grup significatiu de persones alhora.

Aquesta aplicació que procedeix a analitzar i dissenyar estarà connectada amb els GPS dels dispositius mòbils on s'hagi instal·lat, oferint pel final de la cursa un vídeo a vista de planta i un arxiu amb una comparativa de resultats.

1.2 Objectius del Treball

El programari per a dispositius mòbils està enfocat o bé individualment o bé a la competició asíncrona, deixant un buit a la competició sincronitzada (curses). Per aquest motiu aquest projecte pretén establir la base per una posterior implementació d'una aplicació gestora de curses principalment per motor, però essent prou oberta per no tancar la porta a cap tipus de cursa.

El projecte d'anàlisi i disseny d'una aplicació gestora de curses via GPS per a dispositius mòbils amb SO Android pretén fer un anàlisi complet per dissenyar l'aplicació:

- Conèixer què esperen els possibles usuaris finals de l'aplicació.
- Trobar i avaluar els requisits del sistema.
- Identificar totes les entitats i rols que intervindran a l'aplicació.
- Fer un disseny de l'aplicació que faciliti la seva posterior implementació.
- Fer-ho tot amb un equilibri adequat entre la intel·ligibilitat del projecte i la completesa del mateix.

A nivell de projectista, he escollit aquest Treball per diversos motius:

- Entendre què significa elaborar una aplicació mitjana però molt completa partint de zero.
- Assimilar tots els conceptes apresos durant el Grau en referència a l'anàlisi i disseny d'aplicacions, deixant l'aplicació a punt per ser implementada sense dificultats a nivell teòric.
- Ser capaç de plantejar una planificació coherent, de manera que el seu desenvolupament sigui viable.
- I iniciar-me en la programació d'aplicacions per dispositius Android, ja que és un aspecte que m'ha mancat al Grau.

1.3 Enfocament i mètode seguit

El Projecte de Final de Grau "Anàlisi i disseny d'una aplicació gestora de curses via GPS per a dispositius mòbils amb SO Android" té com a objectiu omplir un buit dins el ventall d'aplicacions Android. A més, tractant-se d'un anàlisi i disseny de la possible aplicació, no d'una implementació, el Treball durà a terme el desenvolupament d'un producte nou.

D'aquesta manera, l'estratègia per completar aquest Treball consisteix en els següents punts:

1. Recollida de dades: es tracta de, a través d'una enquesta a diversos *stakeholders*, obtenir diferents punts de vista i saber què n'espera cada un d'una aplicació per gestionar curses. L'objectiu d'aquest punt és no treballar només amb els requisits que com a desenvolupador considero essencials, sinó obtenir una quantitat d'opinions suficient com per poder avaluar els requisits d'una manera objectiva.

Un punt a tenir en compte és que, a més de la quantitat d'opinions a recollir, també és important la qualitat d'aquestes opinions, per això es buscaran entrevistats propers al món de les curses.

2. Anàlisi de requisits: a partir de les dades recollides es detectaran tots els requisits que hi haurà implícits a les entrevistes. Posteriorment es treballaran per identificar els diferents tipus de requisits, s'analitzarà la seva viabilitat, el cost i el desig per part dels enquestats perquè aquests requisits es compleixin o no.

La clau d'aquesta fase és trobar un equilibri adequat entre el fet de descartar requisits amb un descontentament mínim per part dels usuaris finals. Igualment, també s'intentarà trobar el màxim de requisits desitjables a complir amb un cost mínim.

3. Identificar els casos d'ús de cada *stakeholder*: aquesta fase consisteix a identificar i analitzar tots els casos d'ús que intervindran a l'aplicació. Un cop identificats, caldrà elaborar un diagrama de fluxos d'esdeveniments per tal de dissenyar un comportament adequat i coherent per part de l'aplicació. A aquest punt només faltaria elaborar un diagrama complet d'estats de la interfície gràfica. Amb això, caldrà establir un paral·lelisme entre els fluxos d'esdeveniments i un diagrama d'estats de la interfície gràfica.

El més important d'aquesta fase és aconseguir una coherència entre el diagrama de fluxos d'esdeveniments i el diagrama d'estats de la interfície gràfica. D'aquesta manera la implementació de l'aplicació quedarà definida d'una manera clara, reduint la possibilitat de posteriors rectificacions.

4. Elaboració d'un model conceptual del sistema: amb l'aplicació ben definida a nivell d'expectatives i comportament, cal fer un anàlisi a nivell de d'entitats i atributs del sistema.

És important, a més de d'identificar i definir correctament totes les entitats que intervindran a l'aplicació, definir les restriccions d'integritat que s'hauran de contemplar per garantir la integritat del sistema.

5. Disseny teòric del sistema: arribats a aquest punt ja està fet tot l'anàlisi del sistema. Ara només quedaria fer el disseny. A aquesta fase es dissenyarà el diagrama de computació complet del sistema per tal de deixar el Treball a punt per la implementació.

Amb això, degut a que l'última fase requereix tot un aprenentatge sobre la tecnologia de dispositius Android, la fase del disseny teòric del sistema podria reduir-se a només la part que afecti al prototip. D'aquesta manera evitaria subestimar el que pot comportar aquest aprenentatge.

6. Implementació d'un prototip del sistema: de cara a fer una prova del sistema, s'implementarà una part de l'aplicació a nivell de demostració de la funcionalitat que pretén dur a terme aquest Projecte.

Aquesta part també inclourà una fase d'aprenentatge d'implementació d'aplicacions per dispositius mòbils amb SO Android. De fet, el compliment d'aquesta fase encaixa amb un dels objectius com a projectista: iniciar-me en la programació d'aplicacions per dispositius Android.

El fet d'haver de passar per totes aquestes fases ordenadament abans d'implementar el sistema garantirà una evolució ordenada del que inicialment és una idea fins a ser un projecte concretament definit. La idea d'eliminar alguna d'aquestes fases per començar més aviat amb la implementació no s'ha contemplat ja que tot aquest procés és necessari per tenir el projecte definit amb concreció. Qualsevol omissió d'alguna de les tasques podria provocar errors a la fase d'implementació o bé un sistema inconsistent, causant que un inicial estalvi de temps comporti una elevació dels costos d'implementació.

Un altre motiu pel que s'ha considerat adequat definir el sistema d'una manera molt concreta és per permetre fer un codi el màxim estable, evitant haver de fer grans modificacions a l'hora de fer ampliacions o millores del sistema.

1.4 Planificació del Treball

Per tal de dur a terme aquest Treball, els recursos que necessitaré seran:

- Temaris de totes les assignatures cursades al grau d'Enginyeria Informàtica.
- Alguns voluntaris per la fase de presa de dades i detecció de requisits.
- Una gravadora per poder disposar de l'entrevista *a posteriori*. S'utilitzarà Super Voice Recorder.
- Skype per fer algunes entrevistes a distància més la gravadora *Amolto* per enregistrar les converses fetes via Skype.
- Una llicència per MagicDraw: degut al volum d'informació dels diagrames, la llicència necessària és superior a les que s'ofereixen gratuïtament i a la que ofereix la UOC als estudiants. Per garantir un disseny professional seria convenient comprar una llicència, per tal d'evitar duplicitats al declarar rols, entitats i components. Tot i això, s'ha optat per duplicar informació i separar els diagrames en documents independents ja que cap diagrama per si sol supera el límit establert per les llicències de prova.
- Un manual de MagicDraw per poder plasmar certs aspectes més complicats del disseny. En la majoria dels casos s'ha recorregut a consultes ja existents a l'apartat de suport de la web de l'empresa No Magic¹, tot i que ocasionalment sí que s'han consultat manuals de MagicDraw.
- L'entorn Eclipse adaptat per poder desenvolupar i compilar aplicacions Android.
- Una màquina virtual Android.
- Veient lentitud amb la que opera la màquina virtual Android, l'accelerador HAX, que serà descarregat des del propi SDK Manager.
- Un *plug-in* per connectar el Nexus 4 i el HTC One al PC. Al link <http://developer.android.com/tools/extras/oem-usb.html> s'hi poden trobar de tots els dispositius.
- Les llibreries de Google Play Services per poder treballar amb l'API de Google Maps.
- Una clau gratuïta que em proporcionarà Google per poder executar l'API de Google Maps.
- Una aplicació per gravar la imatge de la pantalla del mòbil, per poder reproduir l'execució de les aplicacions desenvolupades. S'utilitzarà A-Z Screen Recorder.
- Un programa per PC per modificar vídeo a fi d'adaptar la seva amida a la seva finalitat. S'utilitzarà el programa online a <http://video.online-convert.com/>.

¹ No Magic és una empresa amb web www.nomagic.com fundadora de MagicDraw.

ID	NOM TASCA	DURADA (setmanes)	SETMANA 1	SETMANA 2	SETMANA 3	SETMANA 4	SETMANA 5	SETMANA 6	SETMANA 7	SETMANA 8	SETMANA 9	SETMANA 10	SETMANA 11	SETMANA 12	SETMANA 13	SETMANA 14	SETMANA 15	SETMANA 16	
			22/09 a 28/09	29/09 a 05/10	06/10 a 12/10	13/10 a 19/10	20/10 a 26/10	27/10 a 02/11	03/11 a 09/11	10/11 a 16/11	17/11 a 23/11	24/11 a 30/11	01/12 a 07/12	08/12 a 14/12	15/12 a 21/12	22/12 a 28/12	29/12 a 04/01	05/01 a 09/01	
PAC1	PAC 1. Planificació del treball	1,5																	
PAC1.0	Planificació del treball	1,5																	
PAC2	PAC 2. Anàlisi del sistema	5																	
PAC2.1	Recollida de dades	1																	
PAC2.2	Anàlisi de requisits	1,5																	
PAC2.2a	- Detecció i classificació de requisits	0,5																	
PAC2.2b	- Avaluació i tria de requisits	1																	
PAC2.3	Anàlisi de casos d'ús (CUs)	1,5																	
PAC2.3a	- Identificació de CUs	0,5																	
PAC2.3b	- Esdeveniments per CU	1																	
PAC2.3c	- Esdeveniments segons GUI	1																	
PAC2.fi	Revisió anàlisi	1																	
PAC3	PAC 3. Disseny del sistema	6																	
PAC3.4	Model Conceptual	1																	
PAC3.5	Disseny teòric del sistema	1																	
PAC3.fi1	Revisió disseny	1																	
PAC3.6	Implementació prototip	4																	
PAC3.6a	Aprentatge món Android	1																	
PAC3.6b	Aplicació de l'aprenentatge amb apps de prova	1																	
PAC3.6c	Disseny prototip	2																	
PAC3.fi2	Revisió implementació	1																	
PAC4	PAC 4. Memòria i presentació	3,5																	
PAC4.7	Fi memòria	1,5																	
PAC4.8	Presentació	2																	

II-lustració 7: Planificació del Treball

1.5 Breu sumari de productes obtinguts

Àudios:

- [enquesta_josep_parc_motor.mp3](#)
- [enquesta_toni_pages.mp3](#)
- [enquesta_josep_vilaro.mp3](#)
- [enquesta_maite_garcia.mp3](#)

Fulls de càlcul:

- [avaluacio_requisits.xlsx](#)

Imatges:

- [SequenceDiagramCreateRace.jpg](#)
- [SequenceDiagramStartRace.jpg](#)
- [SequenceDiagramFinishRace.jpg](#)

Vídeos:

- [app_helloWorld.mp4](#)
- [app_proRace_prototype.mp4](#)

Arxius de codi:

- [HelloWorld.zip](#)
- [ProRace_prototype.zip](#)

1.6 Breu descripció dels altres capítols de la memòria

Al segon capítol de la memòria s'hi inclou un estudi sobre com ha de ser el sistema. Aquest estudi es fa primer fent una recollida de dades per obtenir els requisits del sistema segons els possibles usuaris finals i posteriorment es confronten amb els casos d'ús necessaris per aconseguir que el sistema tingui el comportament esperat. D'aquesta manera s'aconsegueix completar una anàlisi en el que els casos d'ús respectin i alhora garanteixin el compliment dels requisits detectats.

Al tercer capítol es fa el disseny del sistema. Primer de tot s'identifiquen i defineixen totes les entitats que intervenen al sistema, els atributs que les defineixen i les restriccions d'integritat i consistència que presenten. Com a segona part d'aquest capítol es concreta com ha de ser el sistema, posant unes bases de cara a la implementació i definint com ha d'actuar cada cas d'ús dins del sistema, deixant les bases clares sobre com s'ha d'encarar la implementació.

El quart capítol inicialment trenca amb els capítols anteriors, aparquant tot el treball obtingut per entrar a una fase d'aprenentatge, sense la qual no es podria iniciar el desenvolupament de l'aplicació. En aquest aprenentatge es prioritzen els elements que influeixen directament en l'aplicació a desenvolupar, com és el cas de la ubicació via GPS i l'API de Google Maps.

Un cop finalitzada aquesta fase d'aprenentatge, aquest quart capítol combina l'aprenentatge fet amb el sistema estudiat anteriorment, resultant amb un prototip de l'aplicació que s'anomenarà ProRace.

Per finalitzar, el cinquè capítol recupera els objectius marcats a aquest primer capítol i en plasma les conclusions extretes en dos punts de vista: el punt de vista del sistema i un punt de vista més personal que n'extrec com a projectista.

2. Anàlisi del sistema

A l'hora de desenvolupar un projecte és primordial analitzar el treball que això comporta.

A aquest punt el primer a fer serà una recollida de dades per tal de saber què s'espera del sistema que estic dissenyant. Tot i que en tinc una idea bastant clara com a desenvolupador i també com a afeccionat a les curses, cal tenir present que la diversitat d'opinions podrà ajudar a valorar el disseny del sistema de manera objectiva.

Un cop es disposi d'una quantitat adequada de dades de *stakeholders*, caldrà analitzar la informació rebuda i tractar-la de manera objectiva. Aquí els coneixements com a desenvolupador agafen un pes considerable ja que és molt important valorar la complexitat de cada característica que s'afegeix. Malauradament, serà possible trobar-me amb característiques desitjades que no es podran afegir per significar un cost excessiu. Tot i això, des d'un punt de vista comercial, és important equilibrar la balança de la satisfacció de l'usuari afegint el màxim nombre de característiques desitjades que impliquin un cost reduït. Per això val la pena valorar cada requisit que es plantegi, ja que la no presència d'un requisit concret pot no implicar ni sorpresa ni descontentament però, en canvi, fer-la present pot comportar un grau de satisfacció important.

Amb la tria de requisits feta, caldrà especificar quants *stakeholders* diferents hi haurà i quines accions podran dur a terme. A més, també serà necessari desenvolupar a nivell teòric totes les accions que s'hagin considerat i fer-ne un paral·lelisme amb una interfície gràfica (també a nivell teòric).

2.1 Recollida de dades

La recollida de dades és un pas primordial per poder dur a terme un anàlisi del sistema en profunditat i objectivitat.

Hi ha molts mètodes aptes documentats per recollir dades d'una manera eficient. En el meu cas, tenint en compte que es tracta d'un treball individual, el mètode que seguiré serà entrevistar a usuaris potencials del sistema. D'aquesta manera podré conèixer què esperen més enllà de la idea pròpia que tinc del sistema.

Cal dir que també s'ha pensat en altres mètodes com en un *brainstorming* general de requisits o en un *brainstorming* classificat segons els possibles rols dels *stakeholders*. Aquests mètodes han estat descartats perquè al tractar-se d'un únic analista les opinions mai haurien estat suficientment variades. Per tal d'aconseguir més diversitat, també s'hauria pogut recórrer a preguntar-ho directament a possibles *stakeholders*, però prèviament hauria calgut posar-los en context. Per tant, ja que recórrer als *stakeholders* hauria estat igualment necessari,

ha pres més força el mètode de l'enquesta, on s'ha traslladat una pregunta oberta per recollir també un petit *brainstorming*.

El mètode d'observació i prototipatge hauria estat l'encertat per aquesta situació, ja que com a afeccionat a les curses que sóc hauria pogut dissenyar un prototip de sistema que servís per recollir més informació i finalment elaborar l'aplicació definitiva. Malauradament aquest mètode no és possible ja que la meua manca d'experiència i coneixements en programació per dispositius mòbils farien que l'elaboració del prototip s'allargués massa en el temps i provoqués que el projecte no es pugui finalitzar dins el termini establert.

Seguint alguns consells d'un dels documents publicats al web www.monografias.com, l'entrevista estarà dissenyada per ser breu, però alhora molt clara. La intenció és que els entrevistats no triguin gaire a finalitzar però que tinguin molt clar el motiu de l'entrevista i, per tant, puguin ser més clars i amb les seves respostes.

Així doncs, l'entrevista començarà amb una breu explicació de la motivació d'aquesta. A continuació, se li indicarà perquè se l'ha triat i què s'espera d'ell.

Un cop l'entrevistat estigui dins del context, pretendré conèixer-lo una mica amb la finalitat de posteriorment, a la fase d'anàlisi de requisits, saber si les seves respostes han pogut estar basades en intuïcions o bé tenen fonament. Per justificar aquesta afirmació podria posar com a exemple que no és el mateix algú que corre curses habitualment que algú que les corre molt ocasionalment o bé algú que professionalment té aquest món molt a prop.

Amb totes aquestes dades, però abans de començar les preguntes de pes, també és interessant conèixer si el participant és usuari de la tecnologia Android i si la utilitza (o voldria utilitzar) en alguna competició.

A la secció de les preguntes directament vinculades amb l'aplicació, hi haurà un ventall de possibilitats suficientment ampli com per contemplar qualsevol resposta que pugui rebre. Igualment, totes les preguntes estaran orientades amb la mateixa finalitat: saber si l'enquestat trobaria interessant una aplicació d'aquest estil en front del mètode que utilitzi al moment de l'entrevista i guiar el curs de les preguntes cap a la part del formulari on les seves respostes estaran directament relacionades amb les característiques de l'aplicació.

Per estructurar les preguntes relacionades amb les característiques de l'aplicació s'ha fet en dos estils diferents: unes buscant una resposta imaginativa i oberta i altres determinant la importància que es donaria a certes funcionalitats ja concretades.

Doncs seguint totes aquestes premisses, l'entrevista elaborada és la següent:

Com a estudiant de la UOC estic analitzant el desenvolupament d'una aplicació per gestionar curses per mòbils Android. Com que he detectat que a Google Play hi ha un buit en aquest sentit, m'agradaria que em contestessis unes preguntes que em podran fer servei per estudiar com hauria de ser aquesta aplicació.

Abans de començar, dir que el teu perfil de *[definir perfil]* fa que pugui ser molt constructiu conèixer les teves opinions. Doncs comencem:

- Tens Smartphone?
 - o *[Sí]* És Android?
 - *[No]* Quin sistema operatiu té?
- Quan fas curses, com les gestiones? O com les gestionen en cas que ho faci algú altre.
 - o *[Si la resposta no és una app]:*
 - *Què ofereix que t'agradi?*
 - *I algun aspecte en especial?*
 - *Què li trobes a faltar?*
 - *Canviaries o milloraries algun aspecte?*
 - *Per quin motiu no fas servir una aplicació per mòbil?*
 - o *[Si la resposta és una app que no és específica de curses]* Quina aplicació utilitzes? Descriu-la.
 - *Què ofereix que t'agradi?*
 - *I algun aspecte en especial?*
 - *Què li trobes a faltar?*
 - *Canviaries o milloraries algun aspecte?*
 - *Per quin motiu no fas servir una aplicació específica per curses?*
 - o *[Si la resposta no és amb cap app específica de curses]* T'agradaria la idea de poder-la gestionar a través d'una app?
 - *[Sí]* Descriu com te la imagines.
 - Una part important serien els resultats. Què creus que seria important que reflectissin més enllà de la classificació i el temps de cadascú?
 - Valora, si us plau, d'una escala de l'1 al 5, essent 1 gens important i 5 molt important. Consideraries important poder veure l'evolució de la cursa un cop acabada com si es tractés d'un vídeo?
 - També de l'1 al 5. Consideraries important que t'oferís analitzar els resultats dins d'un tram concret de la cursa?
 - També de l'1 al 5. Consideraries important una funcionalitat de contrarellotge?
 - També de l'1 al 5. Consideraries important poder penjar els resultats a alguna xarxa social com ara Facebook?
 - T'imagines fent-la servir? *[Cas afirmatiu]* Amb quina freqüència creus que la utilitzaries?
 - Ara que potser hi has pensat una mica més, què no toleraries que li faltés a una app així?

Bé, això és tot. Moltes gràcies per la col·laboració.

Una vegada fetes les entrevistes, procediré a identificar els requisits que hi ha implícits i classificar-los segons el tipus de requisit.

Cal dir que els requisits detectats són en la seva immensa majoria requisits funcionals, i és que els enquestats quan et parlen del sistema que ells s'imaginen ho fan en referència al funcionament i comportament d'aquest.

Així doncs, treballant amb els requisits funcionals, els classificaré separant si són de funcionalitat (del comportament de l'aplicació) o bé de dades (dades que ha de conèixer el sistema).

- Requisits de funcionalitat:
 - La pantalla a mostrar consistiria en un mapa del recorregut amb un indicador en forma de punt que indiqui on està cada participant (Toni Pagès).
 - Resultats instantanis (Toni Pagès i Josep parc motor). Tots dos volen tenir els resultats al moment en comptes d'esperar a que la organització de la cursa els publiqui.
 - Resultat de la cursa en forma de vídeo (Toni Pagès: 5; Josep parc motor: 4; Josep Vilaró: 4; Maite Garcia: 5).
 - Poder comparar els temps a certs trams (Toni Pagès 5; Josep parc motor: 5; Josep Vilaró: 3-4 però fa notar que ara mateix ja hi treballa; Maite Garcia: 5).
 - Contrarellotge (Toni Pagès: 3; Josep parc motor: 5; Josep Vilaró: textualment "no pinta res" ; Maite Garcia: 4 però remarca que no li dona gens d'importància).
 - Possibilitat de penjar resultats a una xarxa social (Toni Pagès: 5; Josep parc motor: 2; Josep Vilaró: 2; Maite Garcia: 5 tot i que remarca que no ho considera important).
- Requisits de dades:
 - Conèixer l'ubicació dels altres participants durant la cursa (Toni Pagès).
 - L'aplicació hauria de guardar el temps de trams concrets (Toni Pagès, Josep parc motor, Maite Garcia): en Josep del parc motor i en Toni Pagès ho valoren molt positivament, mentre que la Maite, a més de valorar-ho com a irrenunciable, remarca que ja utilitza un sistema per xips que li ofereix aquesta dada.
 - L'aplicació hauria de poder mostrar els temps parcials de cada competidor durant la mateixa cursa (Toni Pagès). En Toni vol poder saber quanta estona d'avantatge li treu un competidor.
 - Fiabilitat de les dades (Josep parc motor). En Josep vol millorar la precisió que li ofereix un cronòmetre.
 - Calcular diferències de temps entre competidors (Josep parc motor).
 - L'aplicació ha de poder mostrar el temps per volta, a més de a certs trams (Josep parc motor).
 - Ha de ser capaç de gestionar el temps sense necessitat d'aturar-se (Josep Vilaró). Està acostumat a haver de parar per tal que li marquin el temps.
 - Càlcul de pulsacions (Maite Garcia). No el destaca per ella però ho considera per altra gent.
 - Càlcul de calories cremades (Maite Garcia). Li faria molta il·lusió una aplicació que comptés les calories cremades. Considera que és molt oportú pel públic femení.
 - Gestió correcta dels desnivells (Maite Garcia). Considera que això no pot faltar.

Amb els requisits funcionals ja definits queda bastant clar com ha de ser l'aplicació a nivell de funcionament, però cal tenir en compte que hi ha altres requisits a tenir en compte que no tenen a veure amb la manera com ha de funcionar el sistema, però que afecten directament a la seva qualitat i a l'acceptació que pot tenir. Aquests són els requisits no funcionals:

- De presentació: **la interfície ha de ser clara i amb poques funcionalitats a cada pantalla**, ja que s'ha d'executar des d'un dispositiu mòbil i les pantalles solen ser inferiors a 5". A la presentació també caldrà tenir en compte una **diferenciació entre disciplines** per tal de mostrar les dades pertinents per cada disciplina. Per exemple, en el cas de les calories, no és una dada rellevant per un perfil de motorista.
- D'usabilitat i humanitat: **la interfície ha de ser fàcilment intel·ligible** sense cap tipus de guia ni instruccions ja que l'aplicació està pensada per tot tipus d'usuaris, independentment de les seves habilitats o formació.
- De compliment: l'execució de l'aplicació no haurà de dependre de la xarxa mòbil ja que durant la cursa els competidors poden passar per trams sense cobertura. La seva **execució normal no s'ha de veure interrompuda mentre no falli el GPS**.
- Operacional i d'entorn: fent referència al requisit de presentació, ha d'estar dissenyada per funcionar amb una pantalla de com a màxim 5". A més a més, l'aplicació **no ha de ser pesant** ja que molts dispositius mòbils tenen poca capacitat de *hardware*.
- De manteniment i suport: s'hauran **d'enumerar tots els possibles errors i elaborar una funcionalitat d'enviament d'informació per tal de detectar els errors** i corregir-los a posteriors actualitzacions. A més, també es pot elaborar un **formulari de contacte per establir una comunicació a nivell de suport amb l'usuari**.
- De seguretat: no s'han considerat requisits de seguretat ja que no caldrà fer *login* per accedir a l'aplicació ja que s'ha encarat amb un accés lliure.
- Culturals i polítics: l'aplicació estarà disponible en els idiomes dels països on més probablement hi ha afició a les curses, siguin del tipus que siguin. Suposant que els països on es disputen Ironman² són una referència de l'afició de la seva població a les curses, a l'hora del disseny, es tindran en compte els idiomes: **català, castellà, francès i anglès**. En aquesta situació, s'ha decidit obviar l'alemany i no el francès degut al seu coneixement de l'anglès, segons el web d'*Education First* <http://www.ef.com.es/epi/>.
- Legals: s'inclourà una **advertència** fent referència a la perillositat de les curses si no són part d'un esdeveniment organitzat i de la necessitat d'una llicència federativa en el cas del motor.

² Competició a nivell mundial que barreja disciplines de natació, cursa a peu i ciclisme. Informació extreta del web www.ironman.com

Amb tots aquests requisits ja només quedaria comentar que els requisits de procés quedarien inclosos a l'apartat de la planificació. Una elaboració més detallada dels requisits de procés correspondria quan es treballés amb una implementació completa del sistema, cosa que es faria només tenint en compte la possible acceptació de l'aplicació per part dels *stakeholders*.

Tenint en compte el desenvolupament de les entrevistes, puc considerar que l'aplicació tindria una acceptació notable a nivells *amateurs*. Tot i això, a nivells d'alta competició s'hauria d'estudiar amb més profunditat ja que, pel que s'ha pogut observar a les entrevistes, els enquestats no són capaços d'imaginar-s'ho ja que utilitzen un sistema centralitzat, completament estàtic i alhora obsolet des de fa molts anys.

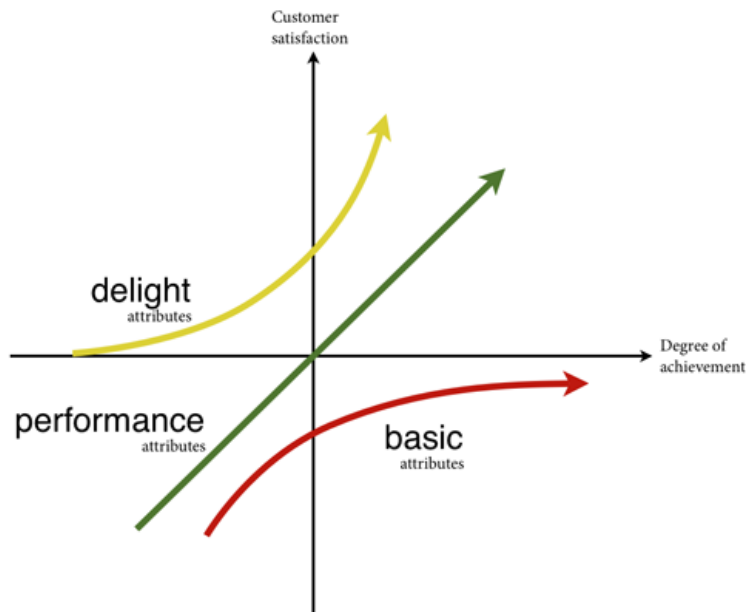
2.2 Anàlisi de requisits

Una vegada completada la fase d'obtenció de requisits tenim un ampli ventall de característiques esperades per usuaris. Però el fet que un requisit concret sigui molt desitjat per un usuari no el fa imprescindible perquè pot ser que sigui poc necessari o gens convenient per d'altres.

Arribats a aquest punt la següent fase és analitzar el valor que aporta cada requisit i, en cas que es tracti d'un requisit desitjable, estudiar què suposa garantir-ne el compliment.

Un mètode àmpliament conegut i molt utilitzat és el mètode de Kano, en el qual es pregunta als enquestats sobre cada requisit de manera funcional o disfuncional. És a dir, preguntant sobre què els semblaria el sistema en cas que un requisit X hi fos (funcional) i també què els semblaria si el mateix requisit X no hi fos (disfuncional).

El model de Kano respondria perfectament al tipus de model que es necessita ja que detecta els requisits bàsics, els apassionants i els lineals. L'objectiu de separar els requisits en aquests 3 tipus és detectar la necessitat real que el sistema compleixi aquell requisit, alhora de detectar requisits concrets amb els que s'aconseguiria que la satisfacció de l'usuari fos el màxim elevada possible. De manera visual, el model de Kano consisteix en el següent:



Il·lustració 8: Imatge explicadora del model de Kano extreta de <http://baymard.com/blog/kano-model>

Aquest mètode, tot i semblar idoni per aquest projecte, s'ha descartat per 2 motius:

- L'excés de temps que hauria pogut durar la nova enquesta. Tal i com s'ha explicat anteriorment, és molt important que l'enquesta sigui breu per tal de mantenir l'enquestat atent durant tota l'enquesta. Aquest projecte té un ventall de requisits bastant ampli i, el fet de preguntar sobre tots ells en 2 sentits diferents, hauria prolongat massa el temps. A més, cal tenir present que en aquest projecte s'està explorant un tipus d'aplicació concreta, descobrint els requisits que hauria de tenir. Per tant, el correcte hauria estat primer fer l'entrevista que ja s'ha fet, després identificar els requisits tal i com s'ha fet i, per acabar, tornar a entrevistar als enquestats amb una nova entrevista que es basi en el model de Kano. Però aquí entra el segon motiu pel qual s'ha descartat aquest model.
- La impossibilitat de tornar a contactar amb alguns enquestats. En aquest sentit la solució hauria estat buscar nous enquestats, però veient les dificultats general que hi ha per entendre aquesta aplicació –ja que ara mateix no existeix– s'ha descartat. I aquest punt coincideix amb el punt anterior en el sentit que s'hauria prolongat massa el temps d'enquesta.

A part d'aquests motius, també cal tenir present que el model de Kano no avalua la complexitat del compliment dels requisits. Aquest és un pas que s'hauria de fer *a posteriori*.

En referència a altres tècniques de prioritització de requisits apreses a l'assignatura d'Enginyeria de Requisits, també s'ha estudiat la possibilitat de treballar amb la tècnica dels 100 dòlars, però ha estat descartada ja que el nombre de requisits a estudiar és bastant elevat, causant una

dificultat segurament difícil d'acceptar pels enquestats. I si a aquesta dificultat s'hi afegeix el fet que la dificultat per entendre el sistema és elevada, la conclusió rau en descartar aquest sistema.

Seguint amb els mètodes apresos a Enginyeria de Requisits, s'ha descartat la tècnica de prioritats limitades perquè, tot i ajustar-se bastant a la necessitat actual, no avalua la complexitat, fet que, tractant-se d'una iniciació al desenvolupament per a Android, ho he considerat imprescindible.

Per acabar, la tècnica d'A/B tèsting s'ha descartat per no ajustar-se al projecte. El fet que aquest model doni dues alternatives no s'ajusta perquè aquest és un projecte que, tot i que parteix de zero, hi ha tota una sèrie de requisits que només s'ha d'avaluar si es compleixen o no. En cas que no es compleixin, no hi ha una alternativa, simplement hi haurà una funcionalitat que no s'implementarà.

Amb aquest estudi previ sobre quin mètode utilitzar per avaluar els requisits, la tècnica que s'utilitzarà serà una barreja entre la tècnica de prioritats limitades i l'avaluació de la complexitat que cada requisit concret implica. Aquesta tècnica consistirà en el següent:

- S'elaborarà una taula on cada fila representi un requisit. Per tal de facilitar la posterior identificació, cada requisit tindrà un codi identificatiu.
- Les primeres columnes mostraran l'acceptació o rebuig dels usuaris per trobar la necessitat real. Aquesta acceptació es plasmarà inspirant-se en la proposta MoSCoW³ que fa el consorci DSDM⁴. Però per tal d'ajustar-ho més a l'opinió de l'enquestat s'anotará la resposta en les mateixes paraules que dona l'enquestat. Fins i tot, en algun cas s'afegirà una anotació referent a l'èmfasi o a l'entonació de la resposta.

La voluntat de mantenir els factors èmfasi i entonació és el que ha fet optar per mantenir les entrevistes en format d'àudio, d'aquesta manera l'observació de les respostes ha estat més objectiva. En aquest sentit, la voluntat ha estat preservar el "què diuen" a més del "com ho diuen".

- El resultat de l'acceptació dels usuaris es trobarà a partir de confrontar les respostes individuals i analitzar si són excloents entre elles o bé si són complementàries.

³ Acrònim de les possibilitats que proposa: Must have, Should have, Could have i Won't have (this time). Cada possibilitat pretén definir la necessitat del requisit en qüestió. És una tècnica proposada pel consorci DSDM.

⁴ Dynamic Systems Development Method. Es tracta d'un mètode que té com a objectiu desenvolupar els sistemes seguint els principis del manifest 'The Manifesto for Agile Software Development' (veure més a www.agilemanifesto.org). La pretensió és poder arribar a una solució amb agilitat.

- A continuació es plantejarà columna que avalui el cost del requisit. a aquesta columna, a més, també es considerarà el cost tenint en compte que la consecució de determinats requisits redueix el cost d'altres requisits.
- Per acabar, hi haurà una conclusió on es decidirà si s'implementa o no el requisit i perquè. Aquesta conclusió es basarà en comparar la necessitat segons els enquestats amb el suposat cost d'implementació.

Amb aquesta tècnica s'obté la taula de l'arxiu [avaluacio_requisits.xlsx](#). Degut a la seva envergadura no s'adjunta a aquest document.

Tal i com es pot observar, al resultat de l'avaluació hi ha uns requisits amb aplicació "a estudiar". Això es deu a que la complexitat de dur-los a terme depèn de la consecució d'altres requisits. En cas que els requisits vinculats no es compleixin, aquests tampoc es compliran ja que el seu cost serà massa elevat en comparació al benefici que es pot tenir. En canvi, en el cas que els requisits vinculats sí que es compleixin, el requisit afectat també s'haurà de complir.

2.3 Anàlisi de CU

Havent finalitzat l'anàlisi de requisits queda bastant més clar amb què ha de complir el sistema. D'aquesta manera, ja es pot concretar "què" ha de fer. Però abans d'identificar el "què", cal identificar "qui", és a dir, els *stakeholders* o actors que donaran ordres.

Per identificar els actors del sistema, cal tenir en compte les següents premisses:

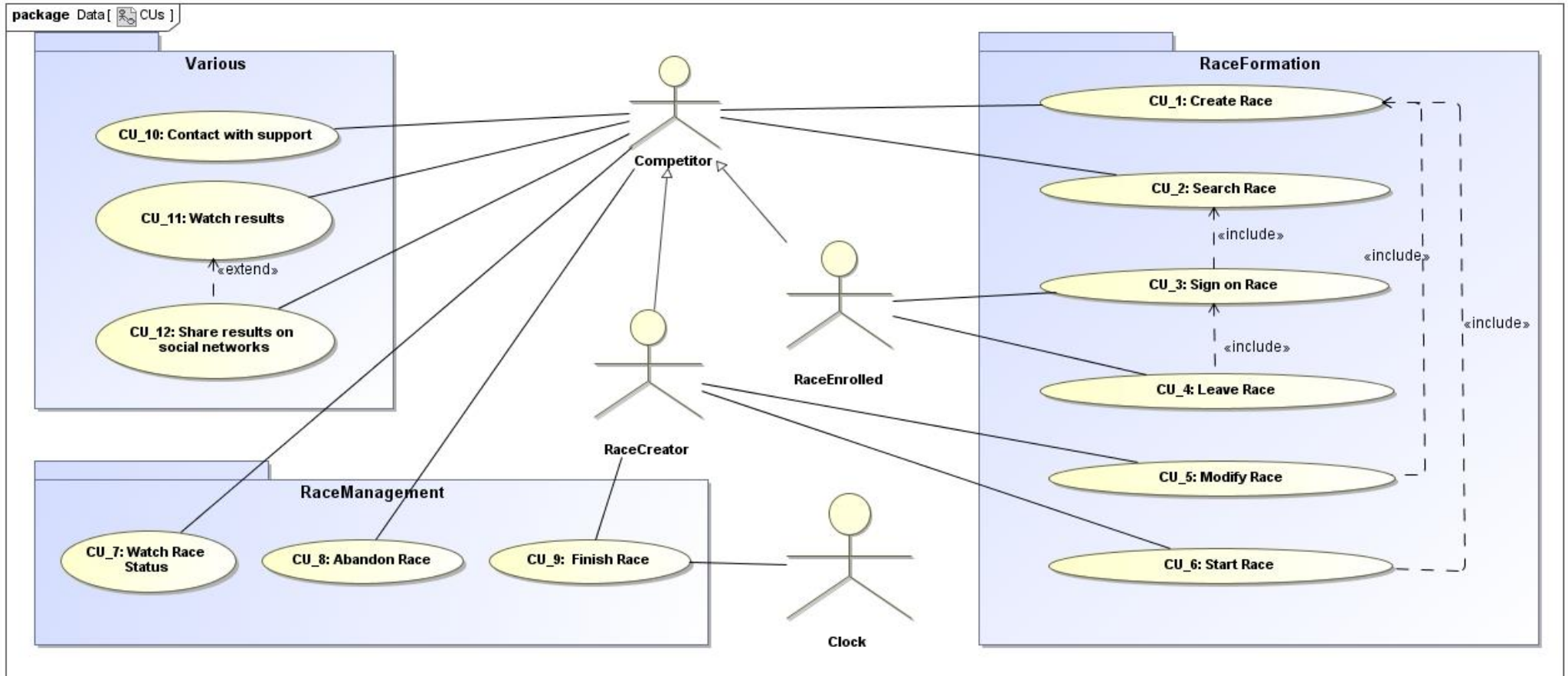
- Quins tipus d'usuaris el faran servir? Competidors: un que crearà la competició i altres que s'hi afegiran.
- Es tracta d'una aplicació a temps real. A més, l'execució està directament relacionada amb els successos relacionats amb el sensor GPS del dispositiu. D'aquesta manera, convindrà establir una rutina que es vagi executant a cada interval X de temps. Així, doncs convindrà tenir en compte l'actor "rellotge" per poder controlar el llançament dels CUs directament relacionats a aquesta rutina.

Definits els actors, també caldrà definir la relació entre ells i els CUs que afecten.

En el cas de l'actor "competidor", caldrà distingir l'actor "creador cursa" del "inscrit a cursa", ja que quan un *stakeholder* crea una cursa pot aplicar una sèrie de casos d'ús que, en cas de ser un competidor inscrit no podria aplicar. En aquesta línia, la mateixa situació es dona en un competidor inscrit respecte un creador de la cursa. Els casos d'ús concrets es poden veure a la il·lustració 6.

L'actor "rellotge" es té en compte per controlar tots els casos d'ús que es poden donar en funció de certes eventualitats en el temps que no són executades directament per un usuari. Aquestes eventualitats tindran a veure tant en el temps d'execució com en la informació subministrada pel sensor GPS del dispositiu. El procediment a seguir serà incloure totes aquestes eventualitats a una rutina que s'anirà executant a cada cert interval de temps. Aquest interval de temps haurà de ser prou breu per tal de treballar amb dades fiables però alhora prou espaiat per evitar col·lapsar el dispositiu. En aquest sentit, cal recordar que s'està treballant per a dispositius mòbils, de manera que el rendiment pot ser molt limitat.

El detall de tots els casos d'ús respecte als actors que els poden aplicar es pot consultar a la figura a continuació:



II-lustració 9: Casos d'ús

Hi ha una relació d'herència entre els rols "Inscrit a cursa" i "Creador cursa" respecte el rol "competidor".

Hi ha una relació d'extensió entre CU_12 i CU_11 ja que el segon pot incorporar eventualment el primer. És a dir, és possible executar el CU_12 a partir de l'execució de CU_11, però també es podria executar a partir de la visualització per alguna altra via dels resultats de la cursa.

Hi ha relació d'inclusió entre els CU_5 i CU_6 amb el CU_1. El motiu és que ambdós CU no tenen sentit sense prèviament haver creat una cursa. Pel mateix motiu, hi ha relació d'inclusió entre CU_3 i CU_2 i entre CU_4 i CU_3.

Tenint definits tots els CU que es contemplaran al sistema, a continuació concretaré l'especificació de cada un:

CU_1: Crear cursa.

Permet a qualsevol competidor donar d'alta una cursa. El flux d'esdeveniments serà el següent:

- 1) L'usuari clica sobre l'opció "crear cursa".
- 2) El sistema condueix a una finestra on l'usuari definirà les característiques de la cursa (nom, disciplina i si es tracta o no de contrarellotge).
- 3) El sistema condueix a una finestra on l'usuari indicarà sobre un mapa de la zona els punts per on passarà la cursa, definint així l'arribada, la sortida i els trams de cursa.
- 4) El sistema dóna d'alta la cursa i condueix a una pàgina de presentació de la cursa, des d'on es podrà modificar, iniciar o eliminar.

CU_2: Buscar cursa.

Permet a qualsevol competidor buscar curses prèviament creades. El flux d'esdeveniments és el següent:

- 1) L'usuari clica sobre el camp de text "buscar cursa", escriu el nom de la cursa a buscar i clica sobre el botó de cerca.
- 2) El sistema mostra la fitxa de la cursa amb el nom coincident. Concretament, mostrarà el nom, disciplina, si es tracta d'un contrarellotge, el mapa i el nombre de competidors apuntats.

En cas que la cursa que s'està buscant no existeixi:

- 2A) El sistema dóna un missatge indicant que la cursa no existeix i es manté al a pàgina principal.

CU_3: Inscriure's a cursa.

Dóna d'alta un corredor a la cursa.

- 1) Executa CU_2.
- 2) L'usuari clica "inscriure's a cursa".
- 3) El sistema condueix a una pantalla d'espera prèvia a iniciar la cursa. D'aquesta manera l'usuari queda inscrit a la cursa, podent crear o inscriure's a d'altres curses mentre aquesta no s'inicia.

CU_4: Sortir de cursa.

En cas que el competidor inscrit s'hi repensi pot abandonar la cursa, en aquest cas, abans que comenci.

- 1) Executa CU_3.
- 2) L'usuari clica "sortir de cursa". Amb això el sistema requereix confirmació.
- 3) En cas que l'usuari confirmi, el sistema condueix a la pantalla inicial.
- 3A) En cas que l'usuari no confirmi, el sistema manté l'usuari inscrit a la cursa i segueix mostrant la pantalla de presentació de la cursa.

CU_5: Modificar cursa.

Aquest CU no té sentit sense prèviament haver executat el CU_1.

Amb la cursa creada, el creador podrà modificar-la encara que s'hagi donat d'alta, sempre i quan no s'hagi iniciat.

- 1) A la pantalla de presentació el competidor clicarà sobre l'opció "modificar cursa".
- 2) El sistema condueix a la pantalla on es dona d'alta una cursa, on permetrà modificar els camps.
- 3) El sistema condueix a la pantalla on hi ha la definició de la cursa sobre mapa, on permetrà modificar tots els punts indicats.
- 4) El sistema notificarà als competidors inscrits que s'ha produït una modificació de la cursa.

CU_6: Iniciar cursa.

Aquest CU no té sentit sense prèviament haver executat el CU_1.

Quan el creador de la cursa consideri que és moment de començar, farà us d'aquest CU per donar el "tret de sortida".

- 1) A la pantalla de presentació de la cursa, el competidor clicarà sobre l'opció "iniciar cursa". Aleshores el sistema mostrarà primer una notificació a tots els inscrits conforme la cursa s'iniciarà en breu, portant-los a la pantalla de presentació de la cursa, on trobaran un compte enrere finalitzat el qual s'iniciarà la cursa.
- 2) Si al finalitzar el compte enrere algun competidor no està col·locat a la zona de sortida, el sistema executarà el CU_4 per aquest competidor.

CU_7: Consultar estat cursa.

Aquest CU només té sentit un cop la cursa ha estat iniciada.

Per definir aquest CU no cal cap flux d'esdeveniments ja que serà la pantalla per defecte que mostrarà el sistema quan la cursa s'hagi iniciat.

A aquest CU intervé activament el rellotge. En aquest punt és quan s'anirà actualitzant la posició de cada competidor. D'aquesta manera, serà necessari consultar la ubicació del propi dispositiu i rebre la ubicació dels dispositius dels altres competidors per mantenir el mapa actualitzat. En el mateix sentit, també caldrà enviar la pròpia ubicació cada vegada que es consulti per tal que els altres dispositius la puguin consultar.

CU_8: Abandonar cursa.

Amb la cursa iniciada, qualsevol competidor, ja sigui inscrit o el propi creador, pot abandonar-la.

- 1) A la pantalla d'estat de la cursa, clicarà sobre "abandonar". Aquesta operació requerirà confirmació.
 - 2) Si la confirmació és positiva, el sistema finalitzarà la cursa per aquest usuari i no en guardarà els temps, deixant-lo com a últim classificat.
- 2A) En cas que la confirmació sigui negativa, es tornarà a la pantalla d'estat de la cursa, mantenint el competidor inscrit.

CU_9: Finalitzar cursa.

Amb la cursa iniciada, el creador pot finalitzar-la quan consideri oportú. Igualment, el rellotge també pot finalitzar-la en cas d'excedir el temps establert o en el cas que la ubicació de tots els corredors actius hagin arribat a la meta.

- 1) S'executa l'ordre de finalitzar cursa, ja sigui per ordre directe del rellotge o per petició del creador.
- 2) El sistema calcula els resultats i els mostra a la pantalla dels competidors.

CU_10: Contacte amb suport.

- 1) A la part superior dreta de la pantalla d'inici, l'usuari clicarà sobre l'interrogant. El sistema conduirà a una pantalla amb un formulari per omplir.
 - 2) L'usuari emplenarà el formulari i clicarà sobre el botó enviar.
- 2A) En cas que no s'hagin emplenat tots els camps obligatoris, es mostrarà un missatge indicant quins camps falten per emplenar. El formulari encara no s'haurà enviat.

CU_11: Consultar resultats.

Qualsevol usuari podrà consultar els resultats de les curses.

En aquest cas hi ha 2 situacions:

- Es vol consultar els resultats de la cursa acabada de finalitzar. En aquesta situació, tal i com s'ha especificat al CU_9, la pròpia aplicació conduirà a la pantalla dels resultats.
 - Es volen consultar resultats de curses anteriors:
 - 1) A la pantalla principal, clicar sobre "Veure resultats". El sistema conduirà a una pantalla amb totes les curses fetes prèviament.
 - 2) Es clica sobre la cursa que es vol consultar. El sistema conduirà a una pantalla amb els resultats de la cursa.
- 2A) En cas que no hi hagi curses amb resultats, el sistema mostrarà la llista buida amb el missatge "no hi ha resultats".

CU_12: Compartir resultats a xarxes socials.

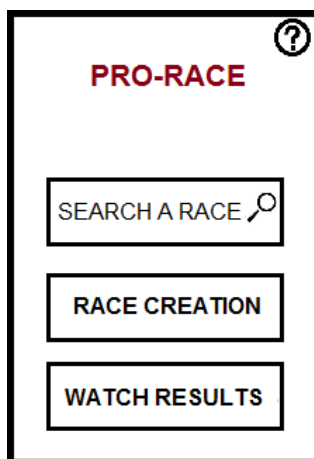
Aquest CU requereix estar a la pantalla amb els resultats d'una cursa. Per això es considera extensió del CU_11.

- 1) Visualitzant els resultats, es clicarà sobre el botó de la xarxa social corresponent.
- 2) L'usuari haurà d'introduir les seves credencials i el sistema publicarà els resultats a la xarxa social.
- 2A) En cas que les credencials siguin errònies, s'informarà missatge d'error i no es farà cap publicació.
- 3) L'usuari visualitzarà la publicació.

2.4 Introducció a la GUI

Amb els requisits clarament identificats i avaluats i els casos d'ús especificats ja hi ha el marc per la interfície gràfica d'usuari. A aquest punt es descriurà com ha de ser la interfície gràfica dels punts més rellevants.

Per començar, als CU 1, 2, 10 i 11 s'ha especificat que s'hi accedeix des de la pàgina inicial. Això, sumat als requisits no funcionals de mantenir una interfície clara i fàcilment intel·ligible (R17 i R19) resulta amb una pantalla d'inici que es limita a les funcionalitats indicades. Per tant, el disseny hauria de ser el màxim de similar possible a la següent:



II-lustració 10: Pantalla d'índex

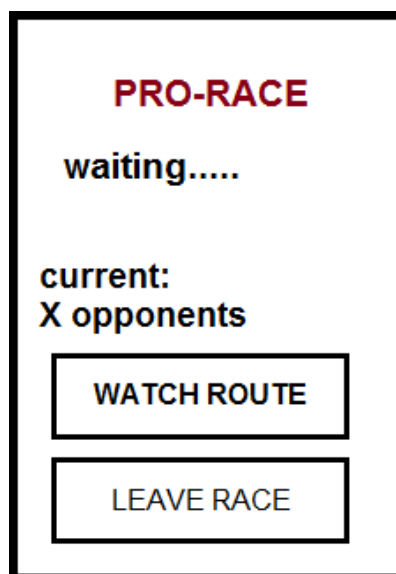
Cal tenir present que la simplicitat d'aquesta pantalla ha de tenir en compte que aquest sistema pot ser utilitzat per gent amb dificultat per entendre aquest tipus de sistema, en el ben entès que es tracta d'una aplicació pionera.

Igualment, la mida dels botons ha de ser suficientment gran per tal de ser polsats amb el dit. En aquest sentit cal no oblidar que en la majoria dels casos no s'utilitzarà un cursor per clicar, sinó el dit de l'usuari, amb una precisió molt inferior.

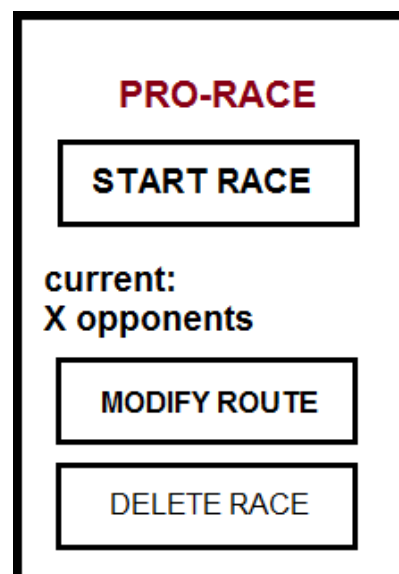
També cal tenir present que aquesta aplicació s'haurà d'executar en una pantalla que oscil·la de 4 a 6 polzades. Així doncs caldrà trobar l'equilibri entre els 2 punts anteriors i aquest, evitant que la pantalla doni una sensació d'estar atapeïda.

A la definició de múltiples CU s'ha pogut observar l'aparició de 2 pantalles que prenen molta importància durant l'execució:

- Pantalla de presentació: com que una cursa és simultània entre tots els competidors, aquesta pantalla és la que presenta la cursa, oferint la ruta i les seves especificacions. A diferència de com s'havia plantejat inicialment (una espera activa). Com que hi constarà la data i hora d'inici de la cursa, l'usuari podrà abandonar aquesta pantalla sabent en tot moment aquest esdeveniment per quan està programat. Així, un cop l'usuari s'ha inscrit, pot seguir utilitzant l'aplicació, inscrivint-se a d'altres curses o bé creant-ne de noves. Amb això, cal tenir en compte que si quan la cursa s'iniciï l'usuari no està utilitzant l'aplicació, seguint el CU_6, l'aplicació llançarà una notificació per informar-li.



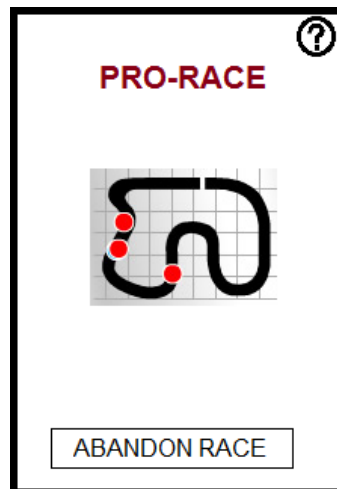
Il·lustració 11: Pantalla de presentació per inscrits



Il·lustració 12: Pantalla de presentació per l'administrador

- Pantalla d'estat de cursa: amb la cursa iniciada, aquesta serà la pantalla visible en tot moment pel competidor. La pretensió d'aquest projecte és aconseguir que a la pantalla hi hagi un mapa del recorregut a pantalla completa. A més, sobre el mapa hi haurà diversos cursors que indicaran la posició a temps real de tots els competidors a la cursa. D'aquesta manera es respon a alguns dels requisits treballats (R1 i R7). Amb la consecució d'aquest objectiu

també es facilita el compliment dels requisits R2, R3, R4, R8, R9 i R11 relatius a la ubicació.



Il·lustració 13: Pantalla d'estat de la cursa

A aquesta pantalla el rellotge pren un paper cabdal ja que ha de mantenir la ubicació dels competidors actius (punts vermells) suficientment actualitzada per tal que s'ajusti el màxim possible a la situació actual.

Sobre les funcionalitats que es deriven d'aquestes pantalles més rellevants, la seva concreció es farà a punts més avançats del projecte, quan el model de computació estigui dissenyat i l'execució d'aquestes funcionalitats s'hagi precisat més, com a conseqüència de l'estudi realitzat.

3. Disseny del sistema

Abans de començar a dissenyar el sistema és ideal haver fet tot l'anàlisi. D'aquesta manera, encara que d'entrada ja tingués una idea de com ha de ser el model final de sistema, hauria començat el disseny amb errors importants.

Concretament, la fase d'anàlisi m'ha fet veure certes limitacions del sistema, com per exemple, la pantalla d'espera activa a la que es troben els competidors abans de començar una cursa.

A la fase de requisits, m'ha permès veure el punt de vista d'altres possibles usuaris, d'aquesta manera he pogut millorar determinats aspectes del meu punt de vista així com ampliar els requisits que inicialment jo hagués anotat.

A la fase de casos d'ús m'ha permès entendre el paper de cada actor i les seves interaccions, així he pogut comprendre les entitats que prenen part i la relació que hi ha d'haver entre elles.

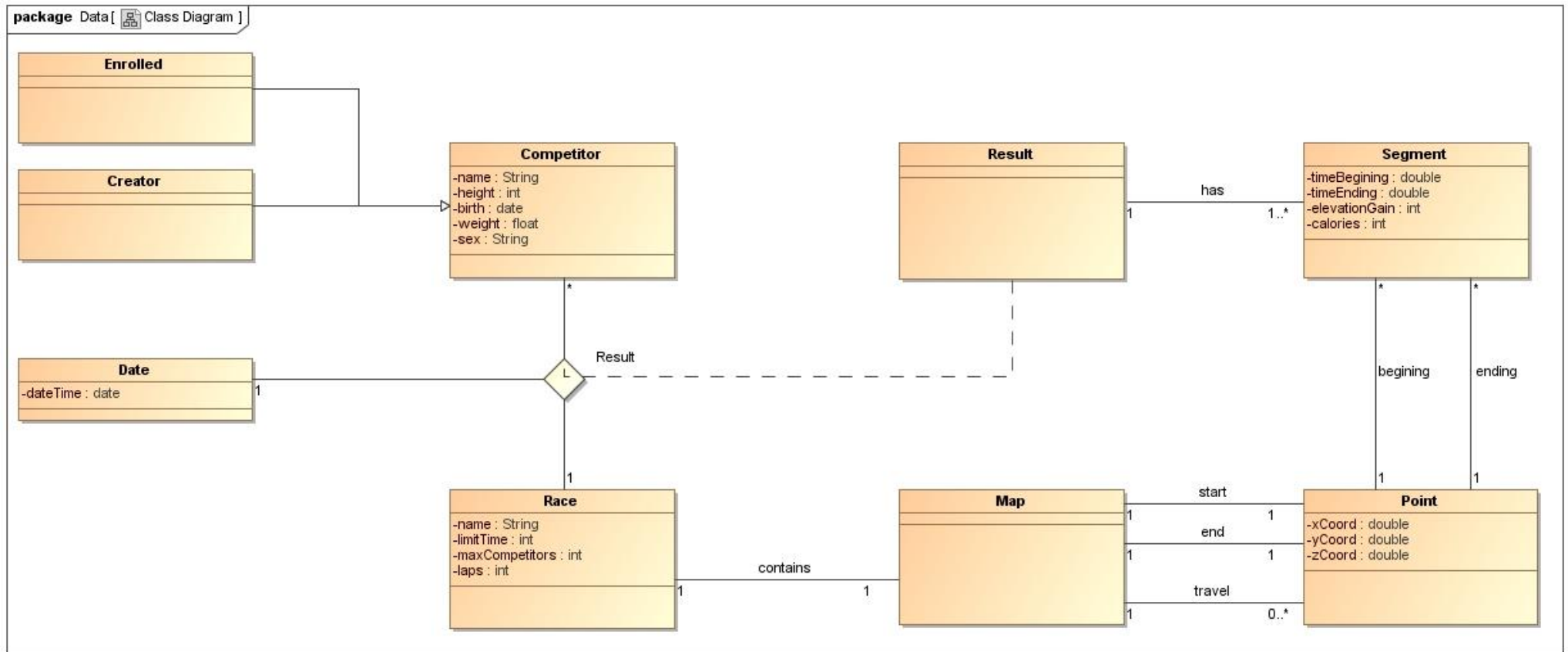
Aquest coneixement del sistema adquirit em permet definir primer, el model conceptual i, després, fer un disseny teòric del sistema.

3.1 Model conceptual

El model conceptual consistirà a definir el sistema. En aquest model agafen forma les entitats i les seves relacions, així com els atributs que les complementen.

Aquesta definició partirà d'un diagrama estàtic, en el que no es tindran en compte ni el comportament ni com haurà de ser la posterior implementació. L'objectiu és deixar totes les entitats existents correctament definides i relacionades entre elles per, a partir d'aquí, procedir amb un model dinàmic que tingui en compte el comportament i les interaccions per després poder començar un disseny amb una bona base teòrica.

Per l'aplicació que s'està estudiant s'ha elaborat el següent diagrama entitat-relació:



II-lustració 14: Diagrama entitat-relació

A mode explicatiu d'aquest diagrama, l'entitat Competidor té herència parcial cap a les entitats Inscrit i Creador. Un cas en el que un competidor no seria inscrit ni creador seria quan entra a l'aplicació i navega sense crear cap cursa i sense inscriure's a cap.

També es pot parlar d'una herència disjunta ja que no es pot donar el cas que un Competidor sigui Inscrit i Creador alhora.

S'aplica el patró d'associació històrica a les entitats Competidor i Cursa. Això consisteix a afegir-hi una relació amb una entitat Data i relacionar-ho amb Resultat com a classe associativa. Així hi ha una relació ternària entre les entitats Competidor, Cursa i Data amb classe associativa Resultat. Aquesta relació pretén definir una instància de Resultat cap a un trio Data – Cursa – Competidor. Amb la ternària també queda definit el següent:

- Per 1 data i 1 cursa hi ha N competidors.
- Per 1 data i 1 competidor hi ha 1 cursa.
- Per 1 competidor i 1 cursa hi ha 1 data.

Amb aquesta ternària també va implícita la possibilitat que un competidor no tingui resultats: es donaria en el cas que no hagués participat a cap cursa.

El Resultat estarà compost per, com a mínim, 1 Segment.

1 Segment tindrà 2 Punts: el d'inici i el de fi de segment. Tot i que 1 mateix Punt pot coincidir a diversos segments.

1 Cursa tindrà associat 1 Mapa.

1 Mapa tindrà associats els Punts de la següent manera:

- 1 que marcarà la sortida.
- 1 que marcarà la meta.
- Una sèrie de punts de control. Cal tenir en compte que poden no existir.

Amb el diagrama d'entitat-relació definit, ara només queda definir les restriccions (*constraints*) d'integritat del sistema, per evitar que es produeixin inconsistències:

- C1: Una cursa només pot ser creada per 1 competidor.
- C2: Si el punt de sortida i d'arribada d'un mapa no coincideixen, el nombre de voltes de la Cursa ha de ser 1.
- C3: No caldrà autenticació per iniciar. Podrà haver-hi usuaris amb el mateix nom durant la cursa. Al calcular els resultats des del propi dispositiu no hi haurà confusions.
- C4: Totes les variables numèriques tindran valors positius.

- C5: La variable *dateTime* de l'entitat Cursa contindrà el dia i hora definits per la cursa, però únicament a nivell informatiu. La cursa s'iniciarà quan ho indiqui el Creador.

Totes aquestes restriccions s'hauran de tenir en compte al disseny ja que el diagrama entitat-relació no les contempla. Tot i això, és important deixar-les definides a aquest punt ja que conceptualment estan lligades al diagrama d'entitat-relació. D'aquesta manera, si no s'estudiessin fins la implementació molt probablement no quedarien totes definides ja que s'hauria fet l'estudi fora de context.

Amb el diagrama d'entitat-relació modelat i les restriccions d'integritat definides ja hi ha la base per dissenyar un sistema persistent.

3.2 Disseny teòric del sistema

Per tal de dissenyar l'aplicació interessa estructurar-la de manera clara i entenedora però alhora que cada mòdul sigui modulable i independent dels altres. Així l'aplicació podrà ser sotmesa a modificacions fàcilment i únicament havent d'aplicar els canvis als mòduls afectats, sense haver d'ajustar la resta.

Per ajustar-se a aquesta necessitat, el disseny necessari encaixa amb el que ens ofereix el patró d'arquitectura *Layers* (arquitectura en capes). Seguint la solució oferta pel patró, hi haurà 3 capes:

- Presentació: treballarà al nivell d'abstracció de la interfície gràfica, essent responsable de la interacció entre l'usuari i els elements de la interfície.

En el cas del sistema estudiat, cal tenir en compte que el disseny serà en Java i la interfície podrà ser gestionada directament amb APIs Java. Aleshores aquesta capa contindrà tot el disseny de la interfície amb la que l'usuari interactuarà i sol·licitarà les dades que necessiti a la capa de negoci. Per exemple, quan un usuari vol buscar curses, aquesta capa traslladarà la petició a la capa de negoci i, un cop li arribi el resultat, el mostrarà per pantalla.

Aquesta capa és l'executora de tots els casos d'ús de l'actor "competidor".

- Negoci: treballarà al nivell d'abstracció de les regles de negoci, restriccions i gestionarà les ordres rebudes de la interfície que necessiten ser processades i rebre dades de la capa de més baix nivell. A més, aquesta capa veu les entitats com a tal, sense especificar al tipus de dades que les componen.

En el cas del sistema estudiat, la capa de negoci gestionarà les interaccions entre entitats, recollint dades de les capes d'integració i

negoci i gestionant la informació per oferir el resultat demanat. Un exemple seria la comunicació entre els dispositius a l'hora d'inscriure's a una cursa, o bé el càlcul dels resultats.

Aquesta capa és l'executora de tots els casos d'ús de l'actor "rellotge".

- Integració: el seu nivell d'abstracció és el de la gestió de les entitats del sistema i dels atributs que les formen. Aquesta gestió ha de ser completa per evitar que la capa de negoci pugui rebre objectes defectuosos.

En el cas del sistema estudiat, la capa d'integració estarà formada per una sèrie de classes amb els noms de les entitats definides a la [il·lustració 14](#), garantint la cardinalitat que les relaciona les unes amb les altres i definint tots els seus atributs.

Amb aquest model d'arquitectura queda tot el que fa referència als diferents nivells d'enteniment del sistema encapsulats en diferents nivells d'abstracció i si durant el disseny es respecte aquest enteniment que s'ha definit, les 3 capes seran independents.

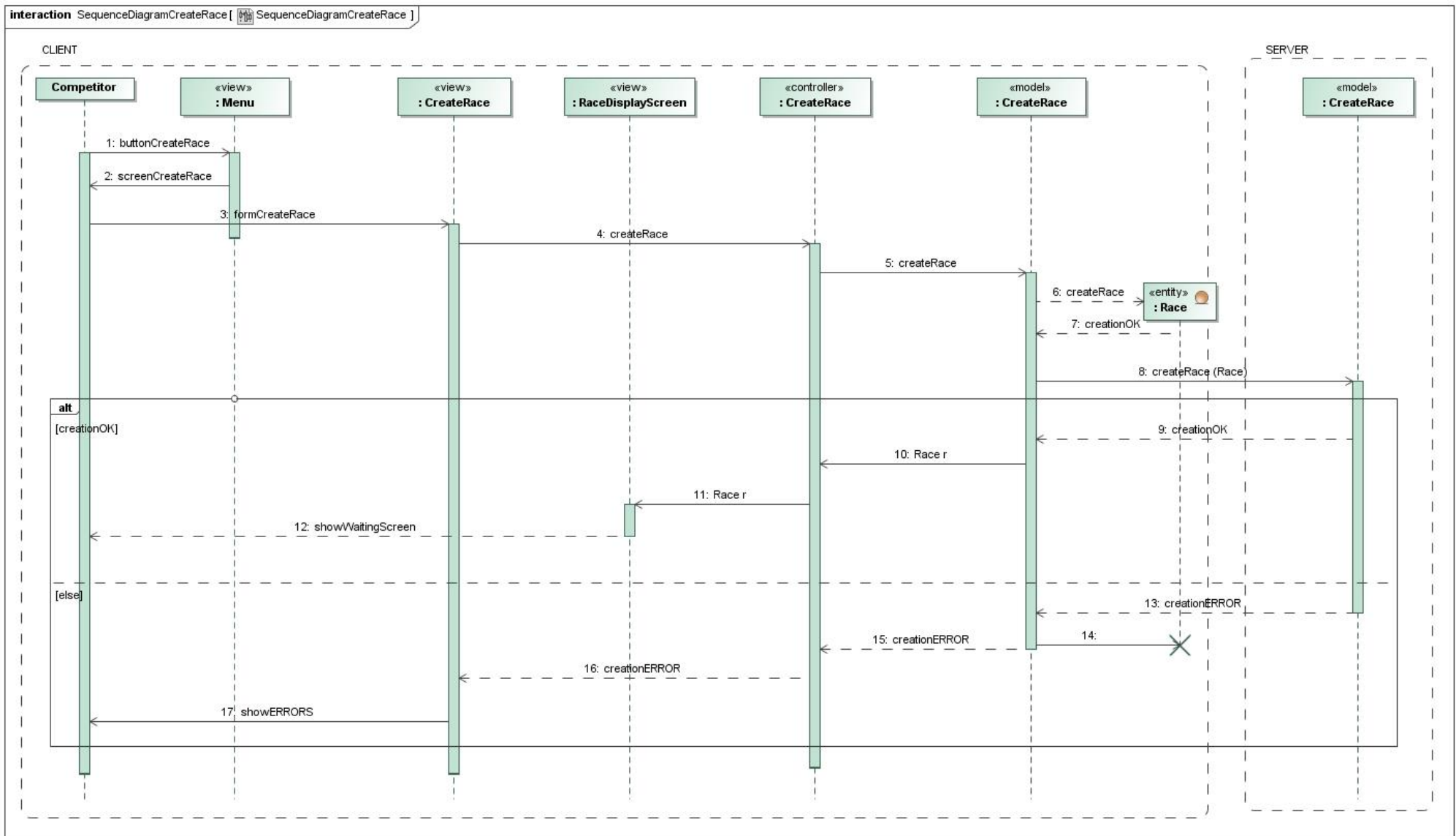
Però amb això encara no és suficient. Cal tenir present que a la capa de presentació hi queden acoblades la interfície gràfica amb les accions que proposa aquesta. A aquest fet cal afegir-hi que els canvis a la interfície gràfica poden ser freqüents i a vegades també dràstics. D'aquesta manera, si no es desacoblen aquests dos aspectes un canvi a la interfície gràfica implica una revisió de les funcionalitats implicades.

Per desacoblar més la capa de presentació el patró *Model-view-controller* ofereix una solució:

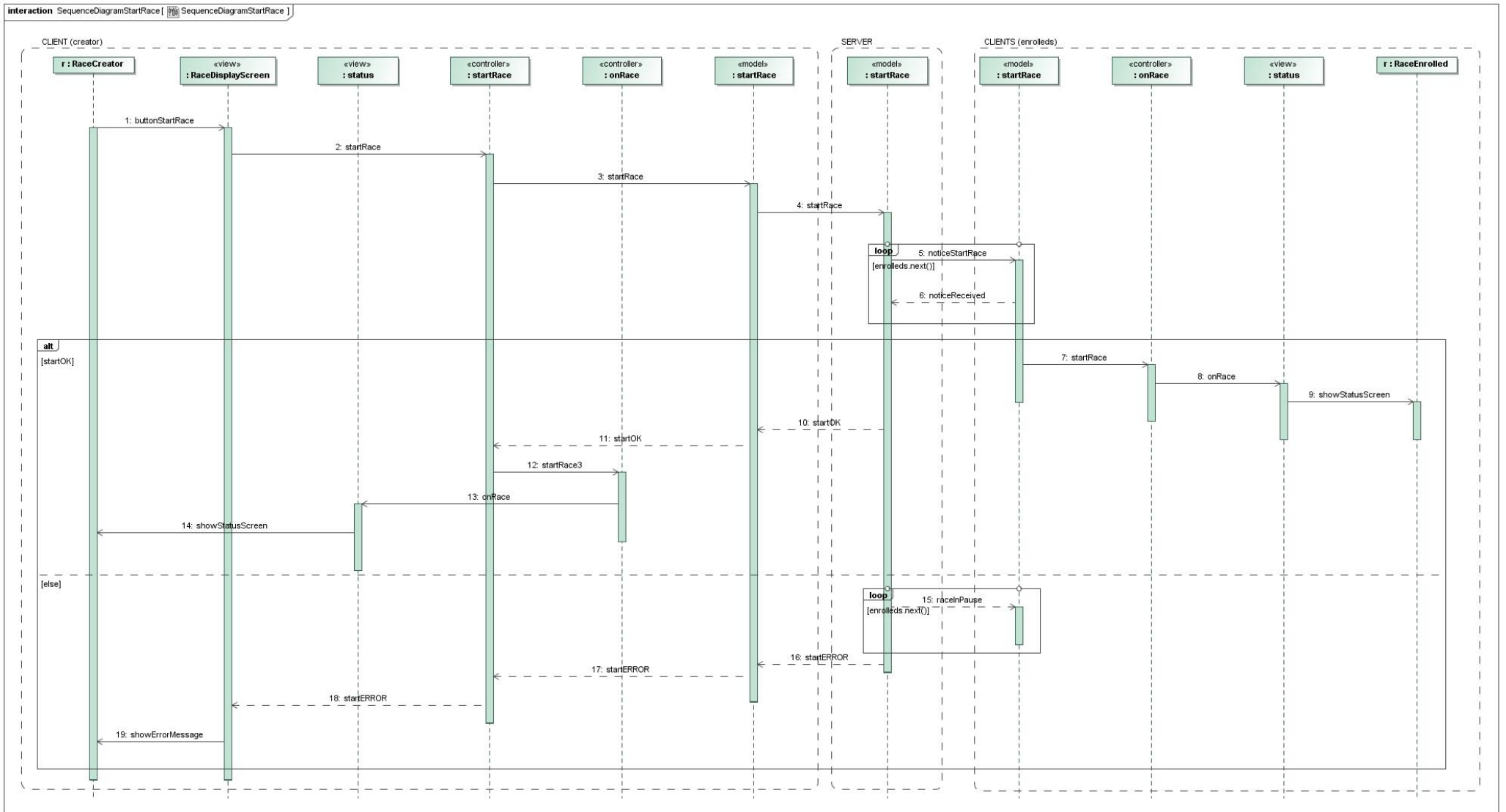
- Vista: inclourà la GUI.
- Model: estarà a la capa de negoci i, rebent l'ordre del controlador, gestionarà la seva resolució.
- Controlador: rep les accions que l'usuari sol·licita a la GUI, les sol·licita al model i gestiona la nova vista a mostrar. Degut a l'alta varietat d'accions (CUs) que es poden produir, el controlador seguirà el model de "controlador de casos d'ús". D'aquesta manera, a més de desacoblar les diferents funcionalitats, facilitarà la tasca del disseny.

A partir d'aquests patrons ja es pot començar a dissenyar el diagrama de seqüència, que és amb el que detallaré les interaccions del sistema per tal de dur a terme cada cas d'ús.

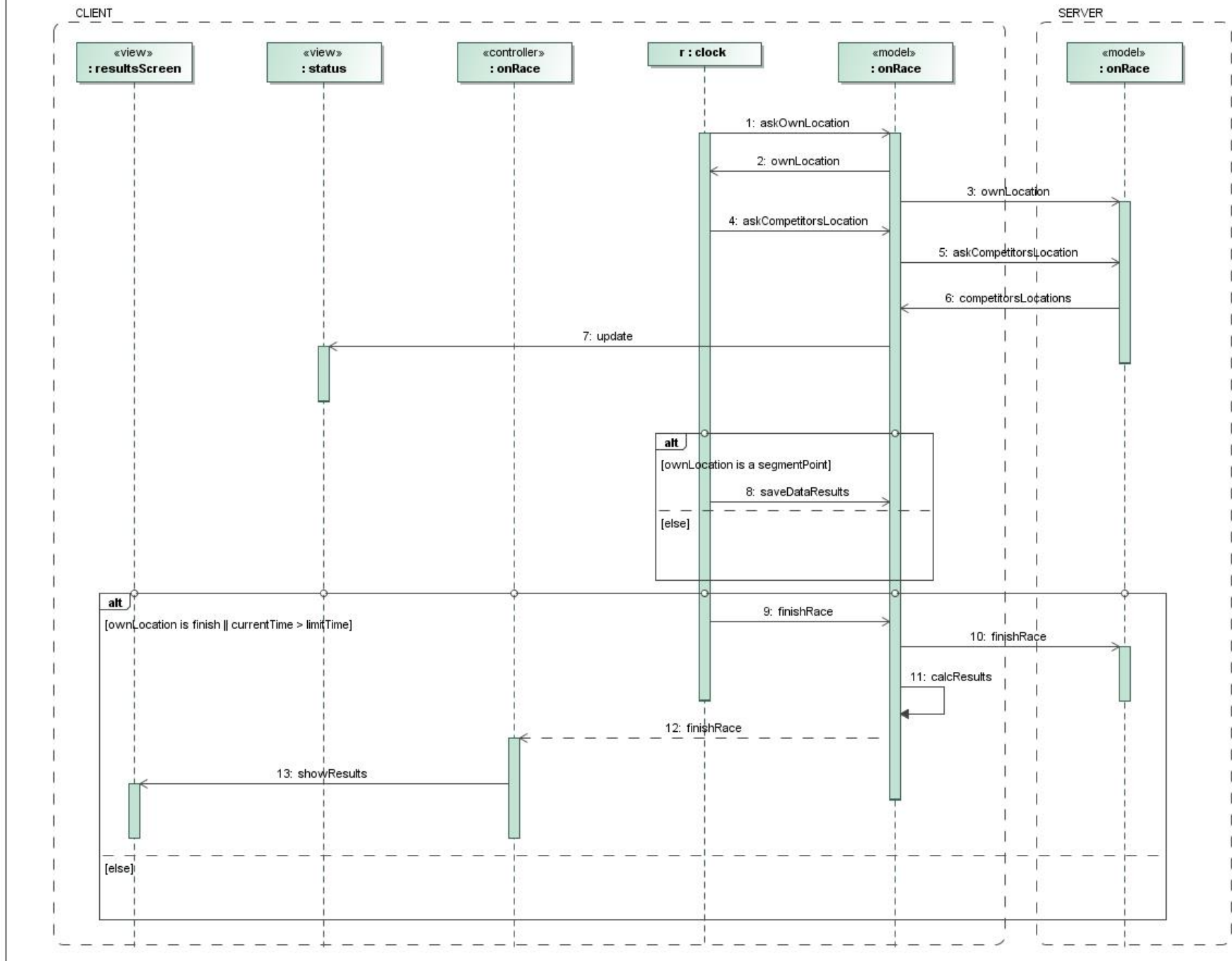
Degut a que la funcionalitat de molts casos d'ús és trivial, es dissenyaran únicament els diagrames de seqüència dels CUs crear cursa, iniciar cursa i finalitzar cursa (clicar sobre el diagrama per veure'n l'explicació):



II-lustració 15: Diagrama de Seqüència pel CU Crear Cursa



II-lustració 16: Diagrama de Seqüència pel CU Iniciar Cursa



És interessant observar que a tots 3 diagrames es distingeix entre client i servidor. Pel correcte funcionament d'aquest sistema, hi haurà un servidor encarregat de dur a terme certes tasques centralitzades. Concretament, aquestes tasques seran emmagatzemament de dades comunes (curses i classificacions) i la comunicació tipus *broadcast*. D'aquesta manera es minimitza el pes de l'aplicació als dispositius. Això és molt important ja que, com es veurà a l'apartat d'implementació, en cas que el dispositiu necessiti més memòria de la disponible, el propi dispositiu pot aturar una activitat en pausa, fet que podria provocar errors i inconsistència a l'aplicació.

El diagrama de seqüència de crear cursa l'inicia un competidor qualsevol. A aquest punt, aquest competidor encara no ha d'estar definit com a creador o inscrit. Un cop l'usuari fa la interacció corresponent per tal d'iniciar aquest CU, el sistema interactua amb l'usuari a través de la capa de presentació, amb els components vista i controlador. D'aquesta manera la capa de presentació recull les dades necessàries, els aplica un control bàsic i les transmet al component model, a la capa de negoci. És aquesta capa la encarregada de verificar la consistència de les dades rebudes i, en cas que no hi hagi incoherències, primer encarregarà la creació de la cursa a la capa d'integració i després transmetrà aquesta cursa al servidor per tal que la creï. Només en el cas que el servidor confirmi la creació, aquesta confirmació es traslladarà a l'usuari transferint-lo a la pantalla de presentació. Però en el cas que el servidor no confirmi la creació de la cursa o en doni un error, el sistema comunicarà aquest error a l'usuari.

[\(Veure diagrama de seqüència per crear cursa ampliat\)](#)

El cas d'iniciar una cursa reflecteix la necessitat d'organitzar aquest sistema amb un servidor responent a certes tasques centralitzades, ja que iniciar una cursa serà una tasca centralitzada.

Per dur a terme aquest CU, l'acció inicial la sol·licitarà un competidor creador a la pantalla de presentació. Aquesta petició viatjarà de la capa de presentació a la de negoci, i d'aquí el model la transmetrà al servidor, qui transmetrà a tots els usuaris inscrits a la cursa un missatge a fi de comprovar la disponibilitat dels usuaris. El sistema de cada usuari confirmarà recepció del missatge al servidor. Aleshores el servidor jutjarà si s'han rebut suficients confirmacions i, en cas afirmatiu, informarà a tots els competidors de l'inici immediat (s'afegirà un *delay* de X segons) de la cursa, mostrant la pantalla d'estat de la cursa. En cas negatiu, el servidor enviarà a tots els inscrits un missatge de "*race in pause*", a fi que cada sistema actualitzi l'estat de la cursa.

[\(Veure diagrama de seqüència per iniciar cursa ampliat\)](#)

El CU finalitzar cursa és especialment interessant perquè el crida l'actor rellotge. D'aquesta manera es concreta què farà el rellotge.

Primer, cal entendre que el rellotge només s'executarà durant la cursa, fet que coincidirà amb la visualització de la pantalla de l'estat de la cursa. Això ho provoca la motivació que el rellotge mantingui actualitzada la pantalla d'estat de la cursa.

El primer que fa el rellotge és actualitzar les ubicacions de la pantalla d'estat, començant per la pròpia i seguint per preguntar al servidor sobre la resta d'ubicacions. Amb la pantalla d'estat actualitzada, el rellotge gestiona la pròpia ubicació per conèixer de quin tipus d'ubicació es tracta. En cas que es tracti d'un punt que coincideixi amb algun dels punts de segment gravats al moment de l'alta de la cursa, es gravaran el temps actual i el punt actual a la variable destinada a emmagatzemar les dades per calcular els resultats.

En el cas que la ubicació actual coincideixi amb el punt de fi de la cursa, el rellotge iniciarà el CU finalitzar cursa, comunicant-ho al model, qui ho comunicarà al servidor, a continuació calcularà els resultats obtinguts i ho comunicarà al controlador. Aquest conduirà a l'usuari a la pantalla de resultats.

[\(Veure diagrama de seqüència per finalitzar cursa ampliat\)](#)

4. Implementació del prototip

Un cop acabada la fase d'anàlisi i disseny de l'aplicació s'entra a la fase d'implementació. A partir de l'estudi fet fins aquí s'ha definit prou específicament l'aplicació com per començar a implementar-la amb una guia clara.

De totes maneres, amb aquest projecte no s'implementarà aquesta aplicació per motius de coneixement i de temps. Tal i com s'ha plantejat, aquest projecte pretén fer una aproximació a la programació per dispositius Android, de manera que serveixi com a aprenentatge pel projectista amb l'objectiu de deixar-ho tot a punt per fer desenvolupar l'aplicació *a posteriori* de la finalització d'aquest projecte.

Seguint la planificació plantejada a l'inici del projecte, les següents fases de treball són aprenentatge i disseny d'aplicacions de prova. Per motius d'eficiència en l'aprenentatge, i a fi de fer un aprenentatge esglaonat, s'ha alterat la planificació i s'han unit la fase d'aprenentatge amb la fase de disseny d'aplicacions de prova. Però a fi de plasmar-ho esquemàticament, a aquesta memòria es presentaran les fases per separat.

4.1. Aprenentatge món Android

Per tal de dur a terme un aprenentatge el màxim de complet i alhora gradual, s'han estudiat paral·lelament diversos documents destinats a servir de manual pel desenvolupament per a Android:

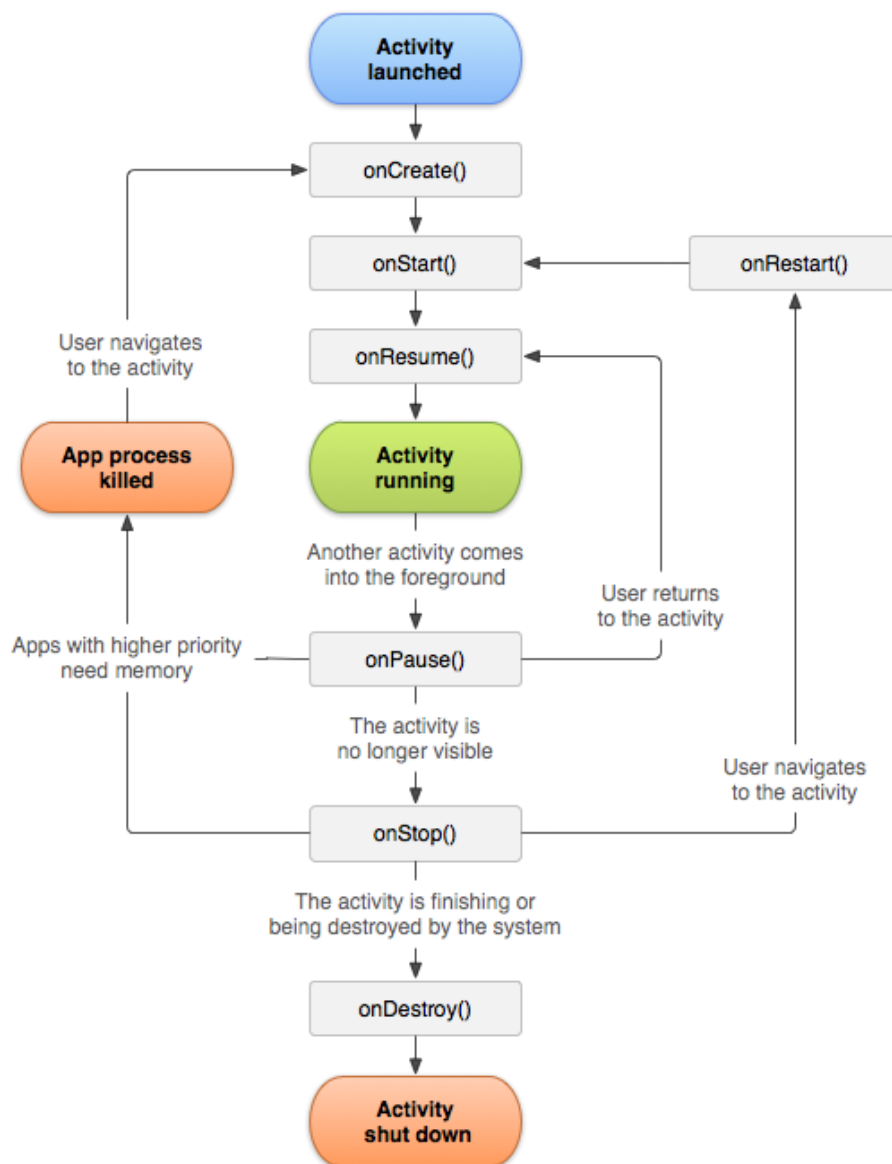
- Curso de programación en Android para principiantes.
- Introducción a Android.
- Desarrollo de aplicaciones móviles.
- ArcGIS runtime for Android.
- I nombrosos recursos web indicats a la bibliografia amb dates entre 28/11/2014 i 04/12/2014.

Gràcies a tots aquests recursos he pogut tenir un aprenentatge similar al que hauria pogut tenir cursant una assignatura, evitant fer un aprenentatge lliure, obviant conceptes bàsics que podrien dificultar la posterior ampliació d'aquests coneixements.

El més bàsic sobre la programació per Android són les activitats. Cal entendre que una activitat és el que col·loquialment es coneix com a "pantalla", tot i que més concretament és la tasca que fa una aplicació a partir d'una interacció amb l'usuari. L'activitat crea una finestra mitjançant la qual s'estableix una comunicació amb l'usuari per tal que aquest interaccioni amb l'aplicació.

L'objectiu que té una activitat és dur a terme una determinada tasca. Es diu que una activitat compleix un cicle de vida complet quan es crea, duu a terme la tasca encomanada i finalment es destrueix. De totes maneres aquesta és una manera molt lineal de veure el cicle de vida de les activitats. Realment, una activitat pot pausar-se ja que requereix del resultat d'una altra activitat per continuar amb la seva tasca. Per entendre aquest procés cal saber que les activitats ofereixen un comportament de tipus pila, de manera que quan una activitat executa una altra activitat, aquesta última es col·loca al capdamunt de la pila i, quan es finalitza es destrueix, deixant l'activitat prèvia al capdamunt de la pila.

Per tal d'entendre millor el cicle de vida global d'una activitat i les alternatives que hi ha, developers.Android ens ofereix, a més de la definició de totes les classes, inclosa Activity, una imatge explicativa del cicle de vida de les activitats:



Il·lustració 18: Cicle d'estats i vida d'una Activity

Amb aquesta imatge es pot observar que, quan una activitat està en funcionament, es pot:

- Pausar: quan una altra activitat passa a primer pla, però aquesta encara no ha finalitzat la seva tasca.
- Aturar: quan l'activitat ja no ha de ser visible.
- Destruïr: quan l'activitat ja ha finalitzat la seva tasca.

També cal tenir en compte que aquest esquema també detalla un altre motiu pel qual es pot destruir una activitat, i és en cas que el sistema necessiti la memòria que aquesta utilitza.

Amb el que s'ha explicat prèviament sobre el funcionament de les activitats en mode de pila, cal tenir en compte que, encara que una activitat estigui pausada, si no es troba al capdamunt de la pila, pot ser que el sistema necessiti la memòria que aquesta activitat està utilitzant i aleshores el sistema la destrueixi. És per això que:

- No s'ha d'abusar de la creació d'activitats, ja que les activitats creades i no destruïdes ocupen un espai de memòria que pot ser utilitzat posteriorment. Sempre és convenient gestionar una destrucció controlada de les activitats.
- És convenient mantenir un ordre a la creació de noves activitats, ja que una incoherència a la seva creació pot provocar que, en cas de destrucció de l'activitat activa, la següent activitat activa no sigui la que l'usuari coherentment esperava.

Com a última notació sobre les activitats, també cal destacar la possibilitat de fer córrer diverses activitats de la mateixa aplicació en paral·lel. Això es fa tractant les activitats com a fils (*threads*) diferents. Aquests fils es declaren mitjançant la classe *thread*, una classe per Android que permet que una activitat no es pausi en cas que s'iniciï una altra activitat, sinó que s'executin paral·lelament. De totes maneres, únicament el fil principal podrà visualitzar-se per pantalla.

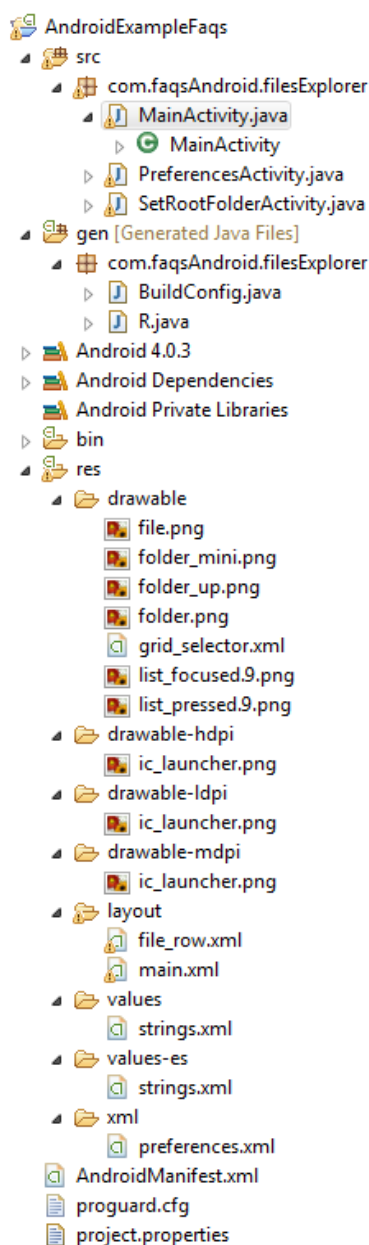
Amb aquesta explicació queda clar què fa una activitat, però no queda clar com aquesta activitat mostra la informació amb la que interactuarà amb l'usuari. Això es fa mitjançant capes (*layouts*). Una o diverses capes van associades a una activitat concreta, mostrant el seu contingut quan aquesta activitat és la principal. Alhora, cada capa està formada per elements. Aquests elements poden ser cadenes de text, imatges, botons... És mitjançant aquests elements com via la classe Java vinculada a l'activitat associada s'aconsegueix la interacció de l'activitat amb l'usuari.

Cal destacar que, a l'hora d'implementar, s'ha de tenir en compte que cada activitat queda plasmada per un costat, com a classe Java i, per l'altre costat, com a recurs en format XML, dins el directori *layout*. A més, no n'hi ha prou amb això per tal de fer funcionar una activitat, ja que d'aquesta manera queda creada però el sistema no en reconeix la seva

existència. Per tal de poder-hi navegar és necessari declarar-la dins l'arxiu del manifest: manifest.xml.

Sobre el manifest, a més de la declaració imprescindible de totes les activitats, també és imprescindible tenir en compte que hi constarà la versió mínima i màxima d'Android per la qual es destina l'aplicació. És important fer el manteniment adequat d'aquestes versions ja que l'aplicació pot quedar inutilitzable en cas que apareguin noves versions d'Android i l'aplicació no s'hi adapti, principalment, modificant el *tag* adequat del manifest.

Com a última part de la fase d'aprenentatge, considero important fer una explicació del sistema d'arxius, ja que la no comprensió del seu funcionament dificulta àmpliament el desenvolupament d'aplicacions. Utilitzaré a mode d'exemple un sistema d'arxius extret d'un curs online de la web www.faqsandroid.com



1. *src*. Directori on hi haurà les classes Java. En aquest cas, només les activitats.
2. *gen*. Directori on Eclipse hi guarda les classes encarregades de gestionar la connexió entre els documents java, els xml i tots els recursos utilitzats. Els documents d'aquest directori són generats automàticament, així que no és recomanable modificar-los.
3. *res*. Directori de recursos. S'hi guarden tots els recursos de l'aplicació separats per directoris. A més, cada tipus de recurs té un tipus de directori i cada tipus de directori té una part del nom en comú i una part diferent:
 - a. *drawable*. S'hi guarden les imatges. Cada tipus de directori *drawable* es diferencia dels altres amb la resolució. Android reconeix la resolució del dispositiu i executa el tipus d'imatge que més convé.
 - b. *layout*. S'hi guarden les capes de les activitats.
 - c. *values*. S'hi guarda principalment les cadenes de caràcters a mostrar. A més, també s'hi guarden valors com dimensions i estils (*dimens.xml* i *styles.xml*). Principalment per les cadenes de caràcters també hi ha directoris de *values* diferenciats per idiomes.
 - d. *xml*. S'hi guarden altres documents xml no contemplats a la resta de directoris. En aquest cas, s'hi guarda el document de preferències de visualització de pantalla.

Il·lustració 19 : Esquema d'arxius

4.2 Entorn utilitzat

A la fase d'aprenentatge també s'ha configurat un entorn adequat per tal de facilitar i alhora agilitzar la programació.

Aquest entorn s'ha obtingut de <http://developer.android.com>. Des d'aquest lloc web s'ha obtingut:

- Un administrador del SDK. Aquest administrador s'actualitza amb totes les versions dels SDK corresponents a cada versió d'Android, a més de tenir una sèrie d'utilitats com un emulador d'Android, de gran utilitat per provar les aplicacions al mateix lloc on es programen.
- Un Eclipse correctament configurat per compilar aplicacions Android. Tot i això, veient que aquest Eclipse –almenys per Windows– presenta mancances bàsiques de funcionament (accessos directes pràcticament inexistent i menús amb mancança de certes funcionalitats), he optat per traspasar a la carpeta del SDK l'Eclipse existent al meu PC i configurar-lo adequadament. Val a dir que l'emulador presentat al SDK manager no connecta amb l'Eclipse si aquest no està a la carpeta del SDK manager.

La conveniència de l'ús d'aquestes eines és:

- L'agilitat amb la que s'obté un entorn correctament configurat.
- La integració entre Eclipse i l'emulador d'Android, que és tal que fins i tot permet *debuggar*, visualitzen a Eclipse la línia que s'està executant i a l'emulador tal i com queda l'aplicació a aquest moment d'execució. És important afegir que també admet la inclusió de *breakpoints* al codi.
- Al crear una nova activitat, quan via Eclipse s'afegeix al manifest, aquest obre automàticament un quadre de diàleg amb l'assistent de generació de la classe Java d'aquesta nova activitat, deixant-les correctament vinculades.
- L'autogeneració de codi d'Eclipse. Tal i com s'ha dit a l'apartat anterior, dins el sistema d'arxius hi ha el directori *gen*, que conté dues classes Java. Una d'aquestes, concretament la classe *R*, conté tots els recursos de l'aplicació. Aquesta classe és la que vincula els documents *xml* amb les classes Java del directori *src*. Amb això, la facilitat que ofereix Eclipse és que auto completa aquestes classes, estalviant la feina de fer-ho manualment.

Un aspecte important que val la pena comentar d'Eclipse és que, per aprofitar els avantatges que ofereix, és important no fer modificacions manualment. Si es fan modificacions manuals als arxius XML en nombroses ocasions Eclipse deixa de reconèixer-lo. També en el cas que es creïn noves activitats o capes sense fer-ho mitjançant el menú *new->other->Android->Android XML Layout File / Android Activity* es corre el risc que Eclipse no ho reconegui i no actualitzi la classe gestora de recursos *R*, deixant sense valor els nous documents creats.

Val a dir que no s'ha fet una comparativa amb altres entorns de desenvolupament, s'ha utilitzat Eclipse a mode d'efecte arrossegament. Tot i haver vist un IDE de Google, Android Studio, he considerat que Eclipse satisfaria completament les meves necessitats pels següents motius:

- Tinc coneixement i experiència amb aquest IDE, estalviant-me el cost d'aprendre a fer funcionar un nou IDE com és Android Studio.
- La majoria de programadors utilitzen Eclipse. Així ho he pogut constatar a base de consultar dubtes genèrics. D'aquesta manera tinc més seguretat de poder trobar resolució a dubtes concrets d'Eclipse.
- Hi ha un ampli ventall de solucions a problemes de configuració d'Eclipse resultants d'una experiència de més de 3 anys en treballant amb Android, mentre que Android Studio només té 1 any i mig de vida.

4.3 Aplicació de l'aprenentatge amb app de prova

Amb la teoria apresada, he considerat convenient el desenvolupament d'una aplicació de prova per posar en pràctica l'aprenentatge de la teoria i també per poder aplicar determinats aspectes tècnics que requerirà el desenvolupament de l'aplicació estudiada, com ara l'API de Google Maps i l'obtenció de les coordenades de geolocalització.

A partir d'aquesta pràctica, puc obtenir un grau d'experiència que em permetrà desenvolupar el prototip entenent el comportament que pot tenir l'aplicació i coneixent les mancances i errades comeses a aquesta aplicació per millorar-ho a la fase de desenvolupament del prototip.

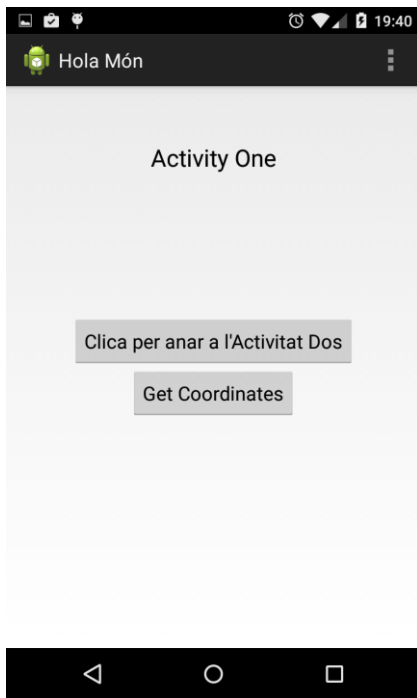
El desenvolupament de l'aplicació de prova s'ha fet separant a cada activitat les funcionalitats que es volien desenvolupar:



ActivityMain: A aquesta activitat he combinat text amb botons, ordenant els botons amb una capa lineal. Cada botó dirigeix l'usuari a una activitat diferent.

Un altre aspecte amb el que també he treballat és amb l'idioma. Per defecte, l'aplicació està escrita en anglès, però sense escriure cadenes de text estàtiques enlloc que no sigui el document XML destinat a això. Així ha resultat fàcil traduir l'aplicació al català. Simplement ha calgut replicar el document XML anglès i traduir-lo al català. En aquest sentit, es podrà veure a algunes pantalles que el text surt en anglès. Això es deu a que l'arxiu en català no ha estat actualitzat al 100%.

Il·lustració 20: ActivityMain

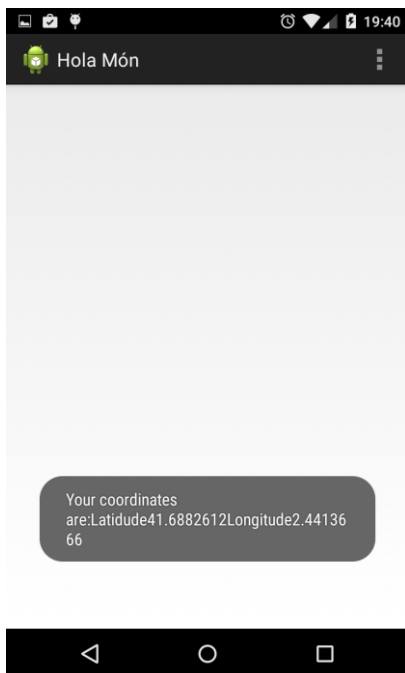


II-lustració 21: ActivityOne



II-lustració 22: ActivityTwo

ActivityOne i **ActivityTwo**: aquestes dues activitats s'han desenvolupat paral·lelament a fi de treballar amb la navegació. Cada una té un botó que condueix a l'altra, així com un botó per una altra funcionalitat. Per tal de millorar la navegació, s'ha modificat el funcionament de la tecla *back* del dispositiu. En aquest cas, en comptes de dirigir l'aplicació a l'activitat directament anterior, dirigeix a ActivityMain.



II-lustració 23: ActivityCoordinates

ActivityCoordinates: l'objectiu d'aquesta activitat ha estat aprendre a interactuar amb el GPS. Fent un paral·lelisme amb l'aplicació estudiada, tinc clar que necessitaré obtenir de manera ràpida i iterativa la ubicació del dispositiu. Doncs aquesta activitat mostra periòdicament la ubicació actual del dispositiu.



Il·lustració 24: ActivityMap

sense els quals l'aplicació no funciona sense donar cap missatge d'error per manca de permisos. En el meu cas, la dificultat ha estat que 1 permís l'he inserit dins el *tag* de l'activitat *AndroidMap*, quan havia d'estar fora d'aquest *tag*, directament amb el *tag* de l'aplicació.

Un aspecte que no contemplava introduir fins a la fase de prototip però que he acabat avançant ha estat la prova de l'aplicació amb un dispositiu real. D'aquesta manera he pogut constatar la fiabilitat de les dades obtingudes, recordant que la precisió és un dels requisits del sistema estudiat.

El resultat d'aquesta aplicació ha estat positiu ja que considero haver assolit els objectius plantejats: aprenentatge i experiència.

Pel que fa al factor experiència, puc valorar que hi ha una sèrie d'errors comesos a aquesta aplicació que convé tenir en compte per no cometre'ls al prototip. Així desenvolupar una aplicació de més qualitat.

- A *ActivityMain* s'ha mantingut el menú d'opcions, però aquestes opcions no contenen res. Cal tenir en compte que, per tenir-lo buit és preferible deshabilitar-lo.
- La navegació entre les activitats *ActivityMain*, *ActivityOne* i *ActivityTwo* és deficient en el sentit que s'ha millorat la navegabilitat entre les *One* i *Two* però no s'ha millorat la del *Main*, resultant amb un empitjorament de la navegabilitat. L'error és que quan l'usuari està al *Main* i clica el botó *back*, l'aplicació el retorna a l'activitat anterior, essent molt probablement *One* o *Two*, les quals sempre el retornaran al *Main*. D'aquesta manera, l'usuari mai podrà sortir de l'aplicació sense tancar-la forçosament o bé clicant el botó *home*. Per solucionar aquest error, cal modificar el comportament del botó *back* del *Main* i

ActivityMap: també fem un paral·lelisme amb l'aplicació definitiva, caldrà fer un ús continu de l'API de Google Maps. D'aquesta manera, a aquesta activitat he treballat amb una ubicació fixa d'un mapa. De totes maneres, això no obvia la ubicació física del dispositiu, que es pot centrar clicant el botó de la part superior dreta de la pantalla.

Dins d'aquesta API, he pogut treballar més concretament amb el zoom i amb el tipus de mapa (en aquest cas satèl·lit).

La gran utilitat de treballar amb aquesta API en una aplicació de prova ha estat conèixer la dificultat de plasmar un mapa de Google per pantalla. Més enllà de la clau d'autorització de Google, també cal declarar al manifest una sèrie de permisos

indicar-li que destrueixi l'activitat. En aquest sentit, també caldrà ampliar la funcionalitat de *back* a One i Two per tal de destruir també aquestes activitats.

- L'activitat `ActivityCoordinates` executa l'ordre d'actualitzar la ubicació i mostrar el valor de cada actualització per pantalla. El problema és similar al cas anterior, com que al sortir de l'activitat aquesta no es destrueix, segueix mostrant per pantalla el valor de la ubicació, a més que manté la ubicació GPS activada constantment. Igual que al cas anterior, això se soluciona donant l'ordre al botó *back* que destrueixi l'activitat en sortir-ne.
- També a l'activitat `ActivityCoordinates` hi ha hagut un altre contratemps. No he pogut executar el *listener* de la localització amb el GPS. En aquesta aplicació, la ubicació es detecta a partir del proveïdor de xarxa. En aquest sentit, fent referència als requisits plantejats per l'aplicació estudiada a l'inici d'aquest treball, com a requisit no funcional s'hi presentava la necessitat de prescindir de la connexió de xarxa. A aquest punt aquest requisit encara no es compleix. Doncs, de cara al prototip, caldrà rectificar la recepció de dades per fer-ho via GPS.

Per tal de poder visualitzar l'execució d'aquesta aplicació, així com per poder comprovar els problemes de navegació que presenta, he elaborat un vídeo amb una execució completa. S'hi pot accedir des del sumari de productes obtinguts o [clicant aquí](#).

També annexo el [codi font](#) d'aquesta aplicació de prova.

4.4 Contextualització del prototip

Tot el que s'ha après i practicat sobre les aplicacions Android cal afegir-ho a l'estudi fet prèviament.

L'estudi contempla una arquitectura en 3 capes, tenint la capa de presentació separada en vista i controlador, deixant la capa de negoci com a model. D'aquesta manera puc contextualitzar l'aplicació de la següent manera:

- Els documents *xml* situats dins el directori *res/layout* formaran part de la vista ja que és el document que conté els elements dins les seves capes, tot associat a l'activitat corresponent.
- Al directori *src*, on hi ha les classes Java, hi haurà 3 paquets que distingiran clarament les 3 capes.
- Paquet amb capa de presentació: contindrà el controlador de les vistes.
- Paquet amb capa de negoci: contindrà les classes que gestionaran la informació dels controladors i comunicaran aquesta capa amb la capa d'integració, per tal de generar les entitats necessàries.
- Paquet amb capa d'integració: contindrà les entitats declarades a la [il·lustració 14](#), garantint la seva consistència i la coherència amb l'estudi fet.

4.5 Disseny del prototip

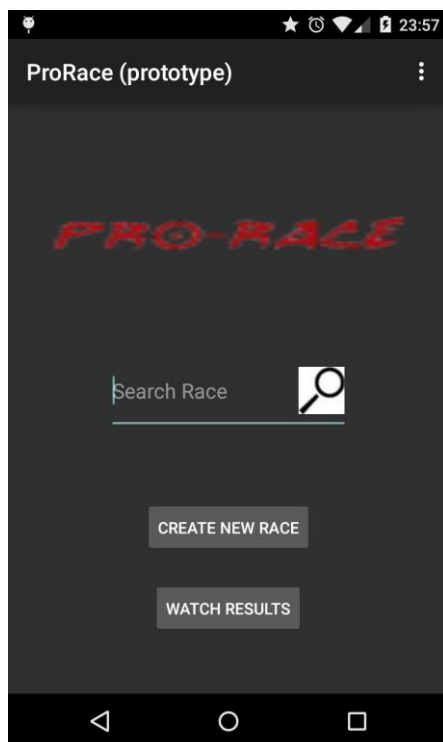
Tal i com s'ha comentat prèviament, l'aplicació estudiada no es desenvoluparà completament per motius principalment de temps. Aquesta fase consisteix a dissenyar un prototip que implementarà les funcionalitats principals de l'aplicació estudiada, deixant la base de l'aplicació preparada per una futura ampliació.

Els objectius pels quals es desenvolupa aquest prototip són els següents:

- Aplicació de les funcionalitats apreses i requerides per comprovar el funcionament en un entorn real i d'aquesta manera avaluar definitivament el cost d'una implementació total.
- Fer una implementació que realment serveixi com a base de l'aplicació completa, evitant desenvolupar un sistema que, en el cas d'una futura implementació total, s'hagi de reestructurar completament. Així es pretén garantir des de l'inici l'escalabilitat del sistema i la independència entre els diferents mòduls.

En relació a la implementació, tractant-se d'una fase d'aprenentatge, s'ha prioritzat un disseny complet i de qualitat abans que aconseguir un ampli abast de funcionalitats. Així que totes les activitats dissenyades han estat pensades en el fet que tinguin una implementació completa, és a dir, aptes per ser integrades a una versió completa de l'aplicació sense haver-n'hi de fer cap modificació substancial.

Les activitats del prototip han quedat de la manera següent:

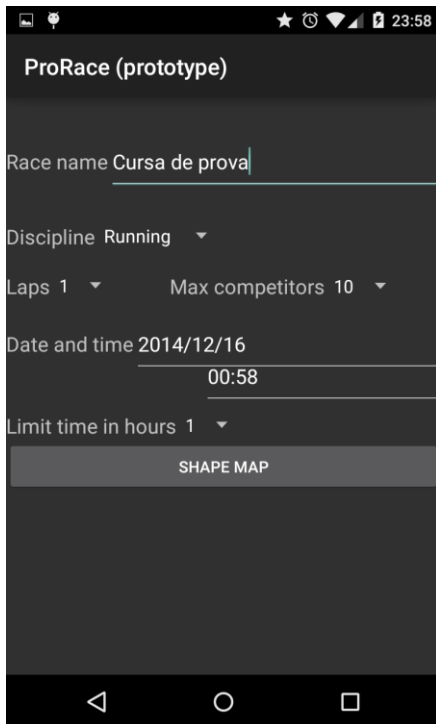


II-Il·lustració 25: ActivityMain

ActivityMain: s'ha intentat dissenyar una pàgina inicial completa.

- A les opcions s'hi ha afegit una pestanya de perfil de l'usuari.
- S'ha afegit un camp de text amb un *drawable* inserit per ajudar a la navegabilitat. Aquest camp de text reacciona a l'enter del teclat i al clic al *drawable*.
- Dos botons de navegació. El codi relacionat a la seva acció s'ha separat del principal per reduir el treball del mètode principal *onCreate*. En aquest sentit, el mateix s'ha fet amb el camp de text de cerca.
- Una modificació de la tecla *back* per afavorir la navegabilitat.

Com a curiositat, aquesta activitat només es pot visualitzar amb la pantalla vertical.



Il·lustració 26: CreateActivity

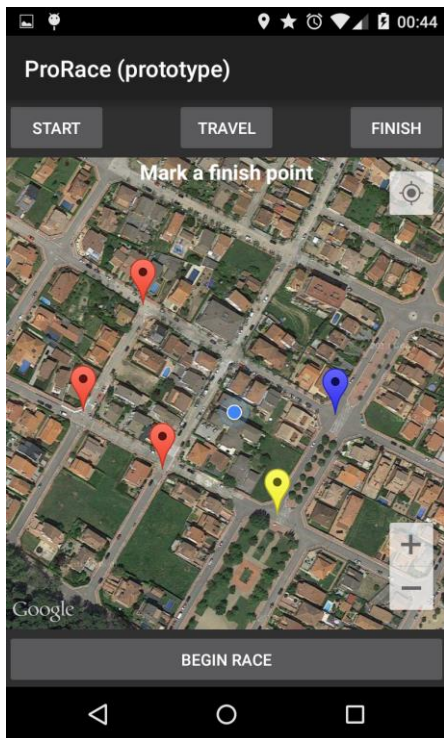
CreateActivity: al tenir més elements que el cas anterior, s'ha procurat donar la mínima complexitat possible a l'activitat, principalment descarregant el mètode *onCreate*.

El desenvolupament d'aquesta activitat ha consistit en treballar amb els elements següents:

- *Spinners*: són les llistes desplegable. Agafen els valors indicats a *strings.xml* i permeten l'accés a l'opció triada per l'usuari.
- *EditText*: són camps de text. Al cas del nom de la cursa s'hi ha fet una modificació bàsica indicant que no pot superar 1 línia i 30 caràcters. En el cas de la data i l'hora s'ha privat a l'usuari que pugui modificar-ho manualment.

- *DatePicker* i *TimePicker*: tot i que no estan implementats a la vista (xml), sí que ho estan al controlador. S'executen quan l'usuari vol modificar la data o l'hora. Aleshores l'aplicació obre el *picker* corresponent per tal que l'usuari editi el camp de text a través del *picker*.
- Botó amb funcionalitat *submit*. Aquest botó s'ha formatat a similitud d'una barra. Executa les següent funcionalitats:
 - o Control d'errors. Informa l'usuari en cas que es detectin dades incoherents o incompletes. El cas més habitual és el nom de cursa buit. Tot i que també verifica aspectes de la data i hora com ara que sigui posterior a la data actual.
 - o Pas de paràmetres. A aquest punt no es crea l'objecte *Race*. Aquest objecte es crearà al final del procés. És per això que es requereix que aquesta activitat transmeti les dades captades i verificades a la següent activitat.
 - o Link a la següent activitat.

A diferència de l'activitat anterior, i degut a l'elevat nombre d'elements que té aquesta activitat, s'ha hagut de contemplar un disseny compatible amb la pantalla en vertical i en horitzontal. Per fer-ho s'ha introduït la funcionalitat de *scroll*, alhora que això ha obligat a fer una combinació de *RelativeLayout* per poder mostrar el contingut complet sense que cap element solapi a cap altre. En aquest sentit, com que la *scrollView* només pot tenir 1 capa filla, a aquesta capa filla se li han hagut d'afegir 2 capes filles més, una per els camps del formulari i l'altra pel botó, ja que d'altra manera el botó solapava el temps límit.



II-Il·lustració 27: CreateMapActivity

demana per defecte que s'introdueixi el punt d'inici i, quan aquest s'ha introduït, automàticament demana que s'introdueixin els punts de mig camí.

- Botó d'iniciar cursa. Aquest botó s'encarrega de crear la l'objecte *Race* i, en l'aplicació completa, comunicaria amb la capa de model per introduir la cursa a la base de dades del servidor. En aquest prototip únicament condueix a la pantalla d'espera de la cursa, des d'on es pot visualitzar i modificar.

Sobre aquesta activitat, val la pena mencionar el procediment de marcar el traçat de la cursa. Els objectes emprats són els *Markers*, facilitats a l'API de Google. Per tal d'afavorir la usabilitat del sistema, s'ha fet el següent:

- Els marcadors es creen amb un clic llarg sobre la pantalla, per evitar l'acumulació de marcadors creats involuntàriament.
- Distingir la sortida, de l'arribada i del trajecte, posant marcadors de diferents colors.
- Anomenar tots els marcadors, incloent una numeració ordenada dels marcadors de trajecte.
- Possibilitat d'arrossegar-los per tal que l'usuari pugui corregir possible errors sense molèstia.
- Possibilitat d'eliminar-los individualment fent-hi un clic curt al damunt.

També és important mencionar que aquesta activitat és la que completa la creació de la cursa. Al clicar el botó "begin race" l'activitat crida el mètode de crear cursa situat a la capa de negoci, el qual revisarà que tot sigui correcte i, en una versió definitiva, indicaria al servidor la creació d'aquesta cursa.

CreateMapActivity: el primer que fa aquesta activitat és rebre els paràmetres de l'activitat anterior.

També a mode de minimitzar la càrrega del mètode *onCreate*, la configuració del mapa es crida des del *listener* creat per aquesta finalitat.

Sobre el disseny d'aquesta activitat, també cal comentar que s'han hagut de diferenciar les capes dels botons (una pels 3 de dalt i una pel de baix) a fi que no tapessin botons rellevants del mapa (ubicació i zoom).

En referència als elements creats:

- Els 3 botons de la capa superior són per gestionar quin tipus de punt es vol crear. De totes maneres, l'aplicació s'ha pensat per facilitar la navegació. Quan s'entra a l'activitat, aquesta



Il·lustració 28: DisplayRaceActivity

DisplayRaceActivity: aquesta activitat rep com a paràmetre la instància de l'objecte cursa creada a l'activitat anterior. El seu funcionament únicament consisteix a mostrar el resultat de la cursa creada.

Per finalitzar amb l'explicació d'aquest prototip, es pot consultar [el vídeo que adjunto](#), on es pot veure el seu funcionament així com es fa una navegació que no es queda només en l'aplicació del cas d'ús, sinó que navega entre activitats explicant les mesures preses per evitar els *bugs*.

Igualment el codi del prototip s'adjunta al següent [enllaç](#).

4.6 Mancances del prototip i visió de futur

Per esquematitzar millor aquest apartat, i degut a l'envergadura que pot tenir aquesta aplicació, es detallaran les mancances del prototip segons una fita en el futur:

- Curt termini: les mancances corresponen als aspectes que estaven contemplats dins del prototip i que no s'han desenvolupat. També s'hi considerarà l'ampliació de l'aplicació per poder utilitzar les funcionalitats de GPS que requeriria l'aplicació completa.
- Mitjà termini: afegir un treball amb base de dades, una mancança imprescindible de solventar per poder assolir el 100% de l'aplicació.
- Llarg termini: establir una arquitectura client – servidor per poder convertir l'aplicació individual que és en aquest moment a una aplicació col·lectiva.

A nivell tecnològic, l'objectiu del disseny del prototip era explotar al màxim l'ús de mapes i del sensor GPS. En aquest sentit, i tenint en compte que la planificació consistia a arribar a la pantalla d'espera, únicament quedaria pendent aconseguir que el fragment de mapa anés seguint el moviment del competidor.

Degut a falta de temps, també ha quedat pendent la inclusió d'una finestra lliscant a l'activitat on es mostra el traçat de la ruta. Aquesta funcionalitat hauria pogut demostrar que el pas de dades entre activitats s'ha fet correctament. Val a dir que les coordenades obtingudes a l'última activitat ja s'han passat com a paràmetre entre activitats, fent ús de la classe *Parcelable* per poder passar el mapa com a objecte.

En visió d'un futur a curt termini, seria interessant treballar aquests aspectes pendents, dissenyant una aplicació de cursa individual que incorpori les funcionalitats de GPS descrites anteriorment i que ara manquen. Amb això s'avançaria en els següents aspectes:

- Poder visualitzar les característiques de la cursa juntament amb el mapa de la cursa, oferint-ne la modificació i l'inici immediat de la cursa.
- Al poder seguir al competidor, es podria finalitzar el recorregut oferint una activitat amb els resultats. Així s'incorporarien requisits com el càlcul del desnivell acumulat i les calories cremades.

A mitjà termini, convindria incorporar una altra mancança d'aquest prototip: una base de dades. Amb això s'aconseguiria el següent:

- Es podria treballar amb un perfil d'usuari i millorar la precisió de dades més personals com les calories cremades.
- Es podrien guardar resultats de curses anteriors per tal que l'usuari pogués comparar resultats.
- Amb aquestes dues últimes funcionalitats afegides, es podria utilitzar el perfil de l'usuari per compartir resultats a xarxes socials, fet que també és un requisit a més d'un reclam per l'ús de l'aplicació.
- Caldria tenir en compte que, amb el perfil en funcionament, caldria afegir a la cursa el nom del creador, per tal de poder-lo distingir.

A més llarg termini i amb les anteriors fites realitzades seria correspondria dur a terme una arquitectura client – servidor. Per tal de poder assolir aquesta fita caldria tenir en compte el següent:

- Les comunicacions dels clients cap al servidor es faran des de la capa de model. A més, han de ser àgils i fiables ja que la continuïtat de l'aplicació en molts casos requereix una resposta del servidor.
- Les comunicacions del servidor als clients podran ser o bé individualitzades o bé de tipus *broadcast*. Les individualitzades correspondran a peticions individuals dels clients, com ara la cerca d'una cursa, la inscripció a aquesta i tasques com la creació o eliminació de curses. Sobre el *broadcast*, cal tenir-lo en compte en situacions com la distribució d'ubicacions dels competidors durant una cursa, el "tret de sortida" per l'inici d'una cursa, l'eliminació d'una cursa a la que ja hi ha competidors inscrits i la distribució de resultats centralitzats.
- El servidor només caldrà que incorpori una base de dades de curses. Segons l'estudi en aquest projecte, les dades corresponents al perfil no seran necessàries pel servidor.
- La capa de model dels clients haurà de desenvolupar el cas d'ús "rellotge".

Amb totes aquestes fites complertes s'assoliria el 100% del desenvolupament, permetent un inici de la distribució a través de la Google Play Store.

Tractant-se d'un projecte informàtic, cal tenir present que mai s'arriba al 100% de l'objectiu. En el cas d'un hipotètic èxit de l'aplicació, el projecte podria seguir amb les següent fites:

- Una de les fites més importants a assolir per l'aplicació seria ampliar-ne els seus usuaris potencials elaborant una aplicació per iOS.
- Una altra fita important hauria de ser establir comunicació amb aplicacions d'èxit com és el cas de dues de les estudiades a l'inici del projecte: Strava i Endomondo.
- A conseqüència d'aquestes incorporacions, convindria donar-li la capacitat de mostrar publicitat i, a conseqüència, poder vetar la publicitat si s'accedeix a una versió de pagament.

5. Conclusions

L'elaboració d'aquest Treball s'ha fet amb la claredat de saber fins on es volia arribar. En aquest sentit, a l'inici es van detallar una sèrie d'objectius per ajudar a organitzar correctament el Treball i així assolir-los. En aquest sentit, i segons els objectius plantejats:

- Mitjançant el coneixement del projectista dins el camp de les curses, i afegint-hi una enquesta realitzada a una sèrie de persones que participen a curses a nivell d'aficionats, s'ha conegut què pot esperar un usuari final de l'aplicació a desenvolupar.
- A partir de saber què pot esperar un usuari final, i afegint-hi els coneixements en informàtica i programació del projectista, s'han detallat i esquematitzat una sèrie de requeriments del sistema.
- Amb els requisits del sistema i la prèvia definició clara del comportament que ha de tenir l'aplicació a desenvolupar, s'han identificat i definit tots els rols que intervindran en la interacció amb l'aplicació, així com també s'ha definit totes les entitats que en prendran part. La consecució d'aquest objectiu ha significat tenir el disseny teòric del sistema completat.
- S'ha dissenyat un prototip essent conscient de les mancances que aquest ha de presentar i així poder tenir en compte com hauria de ser realment (en el cas d'una aplicació completa). Així s'ha dissenyat el prototip com a aplicació escalable.
- Tot i l'escalabilitat del prototip, s'ha dissenyat amb completesa, i així es demostra amb l'execució que s'ha adjuntat amb vídeo, on es busquen –i no es troben– possibles *bugs*.
- Sobre la intel·ligibilitat del projecte, tot i tractar-se d'una característica subjectiva, també s'ha pretès que així fos, elaborant un codi molt esquematitzat i on els mètodes prenen el nom de la funció que fan.

Per tal d'assolir tots els objectius detallats, s'ha hagut d'organitzar la consecució de cada un d'aquests objectius, permetent un assoliment unitari i eficient. Així doncs, a nivell organitzatiu, s'ha elaborat una planificació per poder separar el projecte en fases, on cada fase té una fita, i així poder dedicar el temps necessari a cada fase (i sub-fase). Havent finalitzat el projecte, val a dir que la planificació ha estat completament satisfactòria amb els següents aspectes a remarcar:

- A la fase de l'elaboració del model conceptual, degut a una sèrie de replantejaments que he hagut de fer sobre les bases del Treball, s'ha arribat a acumular un endarreriment d'una setmana. Això s'ha pogut compensar ajustant més el temps reservat per la revisió d'aquesta fase. En aquest sentit, val a dir que al final de cada fase s'ha reservat 1 setmana de revisió de la fase amb l'objectiu de poder corregir errors així com un lleuger endarreriment en la planificació.

- La planificació també s'ha vist alterada a la fase d'aprenentatge de desenvolupament d'aplicacions Android. Aquesta alteració es deu a la manca de coneixements del món Android per part del projectista. En aquest sentit, l'alteració s'ha degut a unir les sub-fases d'aprenentatge i de desenvolupament d'una aplicació de prova, d'aquesta manera s'han anat plasmant a l'aplicació de prova els conceptes que s'anaven assolint. La necessitat d'aquest canvi es fonamenta en que els conceptes a aprendre eren molt variats, anant més enllà de la programació en Java, tot i que també afecta directament a la implementació en Java (per exemple per l'API de Google Maps).

El resultat d'aquesta alteració ha estat excel·lent, ja que ha permès identificar els errors comesos a l'aplicació de prova per encarar un desenvolupament del prototip de més qualitat. A més, l'alteració s'ha fet amb premeditació i marcant unes fites més concretes, permetent ajustar aquesta nova planificació al termini adequat, evitant improvisacions i endarreriments.

Per tant, a nivell de sistema, la conclusió final que n'extrec és la viabilitat del projecte. Degut a la nul·la oferta en aplicacions similars, he hagut d'analitzar bé l'aplicació per entendre si no hi ha oferta per tractar-se d'un projecte inviable o per què de moment és un camp on les aplicacions mòbils encara deixen un buit. Així doncs, havent fet un estudi complet dels requisits, havent-los plasmat en un disseny teòric del sistema i finalment haver desenvolupat un prototip d'aplicació, puc afirmar que la finalització del projecte és viable.

5.1 Comentari com a projectista

Com a projectista, el primer que cal destacar és l'enriquiment que proporciona el desenvolupament d'un Treball com aquest, que abasta des de l'anàlisi complet d'una aplicació nova dins del seu camp fins a un autoaprenentatge partint de zero en el camp de les aplicacions per Android, passant per l'exigència que presenta l'elaboració de la memòria final, tant a nivell de competències comunicatives com a nivell de coneixement, havent de resultar amb un document professional amb mínimes errades.

Igual que pel desenvolupament d'aquest Treball vaig marcar una sèrie d'objectius, també me'ls vaig marcar com a projectista. Així doncs, segons els objectius plantejats:

- He comprès el significat d'elaborar una aplicació mitjana i completa, entenent que un treball de qualitat requereix un cost temporal més elevat que la implementació de les funcionalitats. Cal tenir en compte que els usuaris finals mai fan anar l'aplicació tal i com els desenvolupadors voldríem, provocant que l'aplicació hagi d'estar preparada per no cometre errors deguts a una interacció inesperada.

- He pogut sintetitzar i plasmar una gran varietat de conceptes apresos al llarg del Grau en Informàtica, tant pel que fa a aspectes del disseny teòric, tècniques de desenvolupament, patrons i implementació en Java.
- He assolit un nivell adequat per programar aplicacions per dispositius Android, quedant preparat per aplicacions que van més enllà de la plantejada gràcies a la familiarització que he adquirit amb el SDK de developer.android. En aquest sentit, he assolit uns coneixements que considero necessaris per un estudiant que finalitza el Grau d'Enginyeria Informàtica.

Tot i això, tot i no marcar-ho a la planificació del projecte per no veure-ho viable, personalment m'havia marcat com a objectiu aconseguir que el prototip funcionés per fer una cursa a nivell individual, igual que fa l'aplicació RaceChrono estudiada a la introducció. El motiu pel qual no he assolit aquest objectiu personal ha estat la inversió de temps feta en aconseguir una aplicació de qualitat, fent primer una aplicació de prova i després un prototip que, tot i només tenir implementada la part de la creació de curses, funcionés i estigués estructurat com hauria d'estar-ho l'aplicació completa.

Fent referència a aquest objectiu personal no assolit, aquest serà el primer aspecte a abordar en una línia futura de treball, seguint per organitzar tots els aspectes comentats a la secció de mancances del prototip per tal d'elaborar una nova planificació i així aconseguir completar el prototip per convertir-lo en una aplicació distribuïble, fet que cal recordar que és l'objectiu pel qual s'ha escollit l'elaboració d'aquest Treball.

6. Glossari

CU: Cas d'ús.

GUI: Graphic User Interface. Interfície gràfica d'usuari.

MVC: *Model-view-controller*.

Stakeholder: usuari potencial del sistema.

7. Bibliografia

Pàgines web:

- http://ca.wikipedia.org/wiki/Diagrama_de_Gantt: 29/09/2014
- <http://www.gantt.com/>: 29/09/2014
- <http://www.monografias.com/>: 02/10/2014
- <http://www.ef.com.es/epi/>: 06/10/2014
- www.ironman.com/: 06/10/2014
- <http://developers.facebook.com/>: 16/10/2014
- <http://es.calcuworld.com/calculadora-nutricional/calculadora-de-calorias-harris-benedict/>: 16/10/2014
- ohmyphone.orange.es/: 22/10/2014
- www.tuexperto.com/: 22/10/2014
- www.techulator.com/: 22/10/2014
- <http://maratomontseny.com/>: 30/10/2014
- <http://www.cursademuntanyadegirona.com/>: 30/10/2014
- <http://www.matagallsmontserrat.cat/Mm/>: 30/10/2014
- <http://www.forociclista.com/>: 30/10/2014:
- <http://montenbaik.com/2012/05/como-sacarle-el-maximo-provecho-a-strava/>: 30/10/2014
- <http://www.velocidadabsurda.com/enganchado-al-strava/>: 30/10/2014
- <http://www.gadgetsparacorrer.com/2013/01/29/strava/>: 30/10/2014
- <http://baymard.com/blog/kano-model/>: 03/11/2014
- <http://www.dsdm.org/>: 03/11/2014
- <http://www.nomagic.com/files/manuals/MagicDraw%20UserManual.pdf>: 07/11/2014
- <http://video.online-convert.com/>: 08/12/2014

Pàgines web utilitzades com a suport durant la fase d'implementació:

- <http://developer.android.com/>: 28/11/2014
- <https://software.intel.com/>: 30/11/2014
- <http://developer.android.com/reference/android/app/Activity.html>: 01/12/2014
- <http://developer.android.com/guide/topics/location/strategies.html>: 01/12/2014
- <https://www.youtube.com/watch?v=xClmhYXkV08>: 01/12/2014
- <https://developer.android.com/sdk/installing/studio.html>: 03/12/2014
- <http://www.elandroidelibre.com/2010/08/aprendiendo-android-v-inicializacion-a-la-api-del-gps.html>: 04/12/2014
- <https://www.youtube.com/watch?v=xn9KYnwloBE>: 04/12/2014
- <https://www.youtube.com/watch?v=XKBJKS84UYI>: 04/12/2014
- <https://developers.google.com/maps/documentation/android/>: 05/12/2014
- <https://developers.google.com/maps/documentation/android/start>: 05/12/2014

- <http://developer.android.com/training/basics/firstapp/running-app.html>: 06/12/2014
- <http://developer.android.com/tools/extras/oem-usb.html>: 06/12/2014
- <http://mrbool.com/how-data-is-shared-between-multiple-activities-in-android/28094>: 13/12/2014
- <http://javahowto.blogspot.com.es/>: diverses consultes als seus posts des del 24/11/2014 al final del projecte.
- <http://developer.android.com/guide/index.html>: nombrosos accessos a fi de consultar diverses APIs de classes Android durant la fase de desenvolupament, del 24/11/2014 al final del projecte.
- <http://stackoverflow.com/>: nombrosos accessos a una gran varietat dels seus fils durant la fase de desenvolupament, del 24/11/2014 al final del projecte.

Llibres:

- Jordi Pradel Miquel i Jose Raya Martos, Enginyeria de Requisits, Primera Edició, Eureka Media S.L., Barcelona, any 2012.
- Jordi Fernández González, Jordi Pradel Miquel i José Antonio Raya Martos, Enginyeria del programari orientat a l'objecte, Primera Edició, Eureka Media S.L., Barcelona, setembre de 2005.
- Manuel Báez, Álvaro Borrego, Jorge Cordero, Luis Cruz, Miguel González, Francisco Hernández, David Palomero, José Rodríguez de Llera, Daniel Sanz, Mariam Saucedo, Pilar Torralbo, Álvaro Zapata; Introducción a Android, E.M.E. Editorial.

Papers:

- Gonzalo Génova, Juan Llorens, Paloma Martínez; Semantics of the Minimum Multiplicity in Ternary Associations i UML; Computer Science Department, Carlos III University of Madrid.
- Robert P.; Curso de programación en Android para principiantes.
- Manuel Báez, Álvaro Borrego, Jorge Cordero, Luis Cruz, Miguel González, Francisco Hernández, David Palomero, José Rodríguez de Llera, Daniel Sanz, Mariam Saucedo, Pilar Torralbo, Álvaro Zapata; Introducción a Android, E.M.E. Editorial.
- Antonio Remírez Remírez, ArcGIS Runtime for Android.
- Adrián Catalán per www.maestrosdelweb.com, Curso Android – Desarrollo de aplicaciones móviles.
- Salvador Gómez Oliver, Curso Programación Android.

8. Annexes

Àudios:

- [enquesta_josep_parc_motor.mp3](#)
- [enquesta_toni_pages.mp3](#)
- [enquesta_josep_vilaro.mp3](#)
- [enquesta_maite_garcia.mp3](#)

Fulls de càlcul:

- [avaluacio_requisits.xlsx](#)

Imatges:

- [SequenceDiagramCreateRace.jpg](#)
- [SequenceDiagramStartRace.jpg](#)
- [SequenceDiagramFinishRace.jpg](#)

Vídeos:

- [app_helloWorld.mp4](#)
- [app_proRace_prototype.mp4](#)

Arxius de codi:

- [HelloWorld.zip](#)
- [ProRace_prototype.zip](#)