



Curses

TFC – Memòria

Autor: Lluís Grau Vallverdú

Consultor: Albert Grau Perisé

7/1/2013

Índex de continguts

1.	Descripció del projecte.....	2
2.	Objectius del projecte.	3
3.	Especificacions.....	4
4.	Planificació.....	6
5.	Disseny centrat en l'usuari	8
5.1	Usuaris i context d'ús.....	8
5.1.1	Perfils d'usuari	8
5.1.2	Context d'ús	10
5.1.3	Requisits funcionals	10
5.1.4	Diagrames casos d'ús.....	12
5.2	Disseny conceptual	13
5.2.1	Escenaris d'ús.....	13
5.2.2	Fluxos d'interacció	14
5.3	Prototipatge.....	15
5.3.1	Sketches	15
5.3.2	Prototip alta fidelitat.....	17
5.3.3	Avaluació.....	19
5.4	Arquitectura del sistema.	20
5.4.1	Client: Android App Curses.	20
5.4.2	Servidor: Google App Engine i Datastore.....	21
5.4.3	Client – Servidor: Connexió.....	22
5.6	Model de dades	24
5.6.1	Diagrama de classes.....	24
5.6.2	Base de dades Datastore	24
5.7	Disseny pantalles	25
6.	Implementació	32
6.1	Part client Android.....	32
6.2	Part servidor Google App Engine.....	37
7.	Conclusions.....	43
8.	Bibliografia.....	45
9.	Annex I.....	46

1. Descripció del projecte.

La idea principal del TFC és realitzar una aplicació per mòbil que ofereixi de manera integrada als seus usuaris el calendari de curses a peu de la zona o ubicació que li interessi, amb la corresponent informació de cada una. Al voltant d'aquesta idea base, hi haurà una sèrie de funcionalitats.

Els perfils usuari podran modificar les dades respectives al seu perfil, visualitzar el calendari de curses, filtrar-ne el llistat de forma senzilla i veure en detall la informació relativa a cada cursa. Per altra banda, podran afegir els resultats de les curses que hagin realitzat i també podran veure el seu històric de curses realitzades per poder analitzar informació: el temps d'una cursa i el ritme.

En un futur es vol complementar la aplicació amb altres funcionalitats tals com permetre publicar informació de curses i resultats a xarxes socials, calendaris d'altres tipus d'esports, una agenda pròpia de curses de l'usuari, cerca de resultats i incorporació automàtica al històric, veure traçats de curses a GoogleMaps, valoració dels usuaris, ràanking de punts per un determinat circuit de curses, noves funcionalitats de la comunitat, missatgeria i notificacions, multi idioma etc...

A la part del servidor, s'haurà de dissenyar i implementar la BBDD que emmagatzemarà les dades necessàries per l'aplicació i que podrà comptar amb un entorn de gestió paral·lel de suport a l'aplicació. Inicialment es vol buscar un col·laborador que vulgui col·laborar en alimentar les dades del sistema, però també es pretén donar la opció a organitzadors de curses per usar aquest entorn i tenir la possibilitat de crear, modificar o eliminar curses al calendari.

Els organitzadors de curses tindran un perfil d'usuari afegint la possibilitat de crear, modificar o eliminar curses al calendari. Òbviament podran gestionar la informació de fitxa de cada cursa respectivament. Els usuaris administradors seran els que tindran capacitat per afegir organitzadors.

És important doncs, separar bé cada capa en el disseny per tal que sigui escalable i es pugui ampliar o estendre de manera poc traumàtica.

El resum dels participants en aquest treball:

- Desenvolupador servidor: Implementa les funcionalitats del servidor i assegura el seu funcionament i escalabilitat.
- Desenvolupador aplicació: Implementa les funcionalitats de l'aplicació així com el disseny de la interfície gràfica per al dispositiu mòbil i assegura la seva robustesa en diferents hardware i versions de S.O.
- Documentalista: gestiona la documentació del projecte recopilant i introduint-la al documentari.
- Beta tester: testeja l'aplicació en l'entorn real d'ús verificant que s'assoleixen els requeriments i alertant dels errors en el funcionament tant a la part del dispositiu com en la de connexió al servidor.

2. Objectius del projecte.

Les motivacions a l'hora de realitzar el Treball Final de Carrera (TFC) són diverses, principalment aprofundir en l'aprenentatge del desenvolupament d'aplicacions específic per a dispositius mòbils i obtenir uns coneixements que puguin fer servei en un futur a nivell professional.

A partir d'aquí va venir la idea de crear una aplicació en un àmbit força concret com són les curses esportives, i ve motivada perquè personalment hi ha dos tipus d'esports que practico de forma habitual: córrer i bicicleta. D'aquí sorgeix la idea inicial de desenvolupar l'aplicació *Curses* que permetrà donar informació de curses a peu a nivell de Catalunya i que en un futur també pot estendre's a altres esports i zones geogràfiques.

Actualment podem trobar alguns portals web oferint serveis d'informació de curses i altres funcionalitats de forma integrada, però no així en aplicacions per mòbils, on la competència és mínima. Per tant, és un món que no està explorat i de la qual no hi ha referències, partint de zero.

Amb la realització d'aquest TFC complirem una sèrie d'objectius:

- Dissenyar i desenvolupar una aplicació específica per dispositius mòbils que sigui escalable en un futur.
- Adquirir coneixements necessaris sobre la plataforma Android i el seu desenvolupament d'aplicacions.
- Implementar un SGBD amb la base de dades específica per suportar la aplicació.
- Adquirir coneixements sobre la plataforma per realitzar la persistència de dades.
- Comunicació i emmagatzematge de dades al servidor.
- Utilitzar llibreries externes mitjançant les API per crear algunes funcionalitats.
- Posar en pràctica els coneixements adquirits durant la carrera.
- Decidir un model de negoci per l'aplicació dins del mercat.
- Posar l'APP a dins el mercat i veure'n la evolució.

3. Especificacions

Després de valorar diferents opcions possibles, per al desenvolupament del projecte, s'ha optat per utilitzar les següents tecnologies:

Entorn de desenvolupament: Eclipse + plugin GPE.

Llenguatge de desenvolupament: Java EE.

Llibreries: JDK 1.7, SDK Android API 16, SDK Google App Eng 1.7.3, JDO, JSON, Datadroid.

Plataforma: Android v2.2 o superior.

Sistema gestor de BBDD: Datastore (HRD).

A continuació detallarem cada tecnologia que hem decidit emprar i les raons de l'elecció entre les opcions possibles. Cal dir que aquest apartat s'ha ampliat durant el desenvolupament del projecte, donat que s'ha cregut oportú fer ús d'algunes eines concretes durant les fases de desenvolupament.

- **Eclipse:** entorn potent de desenvolupament per a la programació del codi de l'aplicació en JAVA i XML. Presenta com avantatges el fet d'incorporar plugins específics per a programació en Android i Google App Engine, que faciliten la vida.
- **Android SDK:** framework per al desenvolupament d'aplicacions per al sistema Android que inclou un **manager** per instal·lar diferents SDK i extres i l'**emulador AVD** amb el que s'han fet tests de l'aplicació sobre un dispositiu virtual.
- **Llibreria GAE:** proporciona l'entorn de programació de l'aplicació web que es depositarà a Google App Engine i que permetra la persistència de dades al Datastore, en la infraestructura High Redundant Datastorage. Els principals avantatges a l'hora d'escollir aquest sistema, són el fet de poder disposar d'una infraestructura potent, escalable, i gratuïta per la nostra part servidor.
- **Llibreria JDO:** és una interfície estàndard de Java per emmagatzemar els objectes del nostre model que contenen dades i és independentment de la base de dades que hi hagi per sota.
- **Llibreria JSON:** és una llibreria per a usar un format de dades independent del llenguatge, de manera que puguem lligar el model de dades del nostre sistema client amb el del servidor amb els respectius *parsers*.
- **Llibreria Datadroid:** codi font de la llibreria que ens permet la connexió i comunicació del nostre client Android amb els WebServices de Google App Engine via HTTP, adaptant les seves capes de connexió i gestor de requests i serveis a la nostra aplicació.
- **Datastore:** infraestructura d'emmagatzematge de Goggle App Engine que proporciona distribució, replicació i balanceig de càrrega per a les dades a través d'una API senzilla amb un potent motor de transaccions i consultes.
- **Samsung Galaxy Ace:** dispositiu mòbil amb la versió 2.3 d'Android per realitzar la instal·lació i les proves en real de l'aplicació.

Per poder realitzar els tests corresponents de les parts que es vagin desenvolupant, tindrem el següents requeriments:

- Dispositiu real i virtual amb sistemes Android versió 2.2 i 4.1.
- Localhost amb entorn d'administració i desenvolupament per Google App Engine.
- Navegador Web per realitzar les crides als WebServices.

Per realitzar els tests en l'entorn real, s'han d'afegir requeriments en particular, donat que s'haurà de disposar de les adreces URL per connectar als serveis externs dels servidor d'aplicacions Google App Engine. Per tant s'haurà de crear l'aplicació i a través del plugin GAE per Eclipse, s'ha de dipositar als servidors Google en un compte creat per a tal fi. A banda, els dispositius hauran de disposar d'una tarifa de dades per tenir connectivitat 3G com a mínim o disposar de connexió Wifi a Internet.

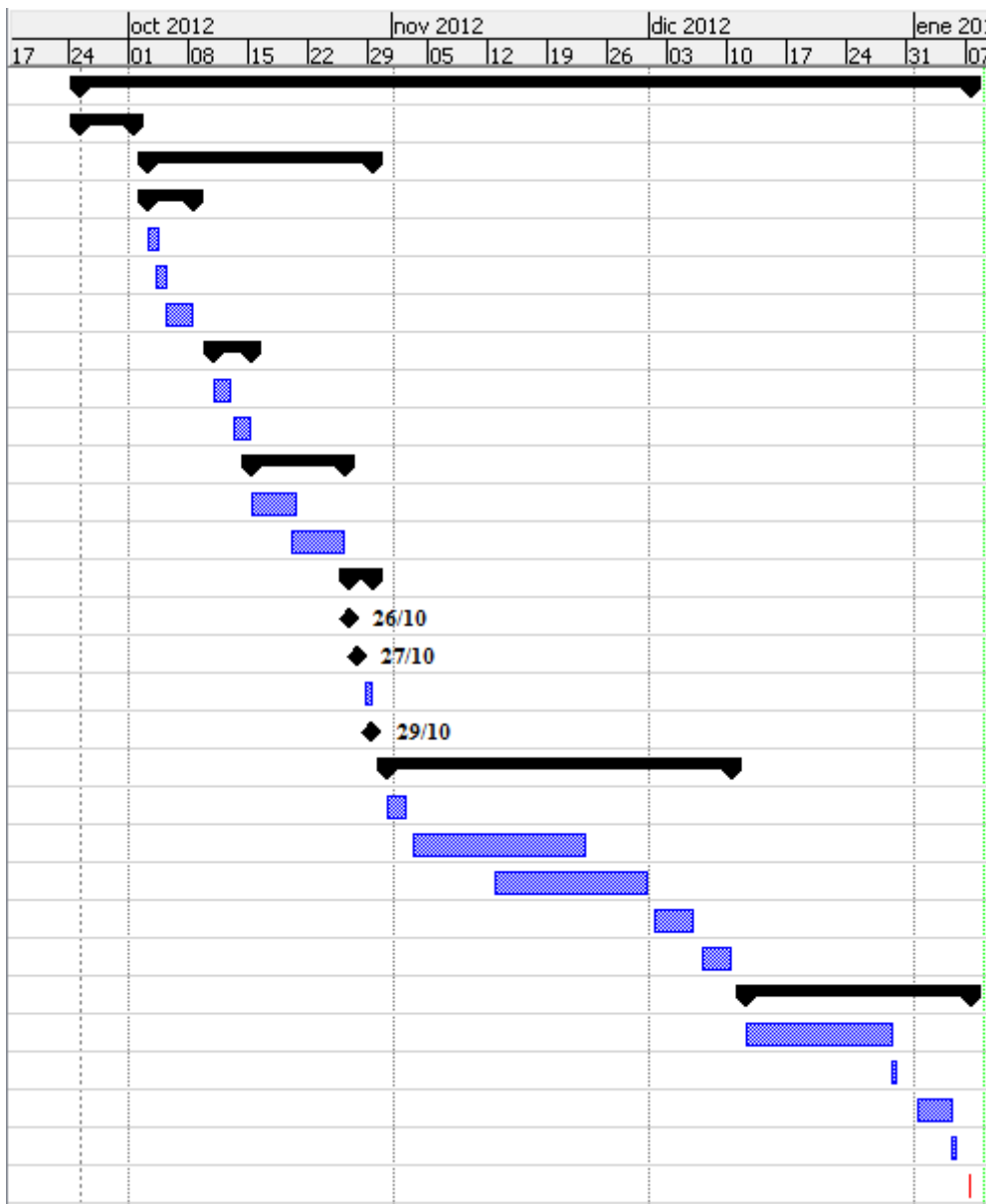
4. Planificació

El projecte va començar el 25/09/2012 i les tasques que s'han realitzat s'han dividit en diferents parts corresponents a cadascuna de les entregues parcials. A continuació es mostra en forma de taula les quatre etapes principals i una taula detallada amb totes les tasques :

Fase	Duració	Inici	Fi
Pla de Treball	7 dies	25/09/2012	01/10/2012
Disseny centrat en l'usuari	28 dies	2/10/2012	29/10/2012
Arquitectura, implementació i proves	40 dies	30/10/2012	10/12/2012
Producte, memòria i presentació	28 dies	11/12/2012	7/1/2013
Total	103 dies	25/09/2012	7/1/2013

Ordre Proyecto	Nombre	Inicio	Terminado
☐	Projecte TFC	25/09/12 8:00	7/01/13 17:00
☐	Proposta projecte	25/09/12 8:00	1/10/12 17:00
☐	Anàlisi i DCU	3/10/12 8:00	29/10/12 18:00
☐	Usuaris i context d'ús	3/10/12 8:00	8/10/12 17:00
	Perfis usuari	3/10/12 8:00	4/10/12 17:00
	Context ús	4/10/12 8:00	5/10/12 17:00
	Anàlisi de tasques	5/10/12 8:00	8/10/12 17:00
☐	Disseny conceptual [Disseny]	11/10/12 3:00	15/10/12 10:00
	Escenaris d'ús	11/10/12 3:00	13/10/12 2:00
	Fluxes d'interacció	13/10/12 10:00	15/10/12 10:00
☐	Prototipatge [Disseny]	15/10/12 10:00	26/10/12 9:00
	Sketches	15/10/12 10:00	20/10/12 19:00
	Prototip alta fidelitat	20/10/12 5:00	26/10/12 9:00
☐	Avaluació [Avaluació]	26/10/12 20:00	29/10/12 18:00
	El recull de preguntes d'informació	26/10/12 20:00	26/10/12 20:00
	Les tasques que els usuaris haurien de realitzar	27/10/12 20:00	27/10/12 20:00
	Les preguntes referents a les tasques	28/10/12 21:00	29/10/12 18:00
	Lliurament PAC2	29/10/12 13:00	29/10/12 13:00
☐	Arquitectura , implementació i proves	31/10/12 6:00	10/12/12 17:00
	Document d'arquitectura de l'aplicació	31/10/12 6:00	2/11/12 16:00
	Implementar l'aplicació part mòbil	3/11/12 9:00	23/11/12 14:00
	Implementar la part de servidor	12/11/12 23:00	30/11/12 22:00
	Generar la base de dades	1/12/12 16:00	6/12/12 8:00
	Realitzar un conjunt de proves	7/12/12 7:00	10/12/12 17:00
☐	Entrega Final	12/12/12 8:00	7/01/13 17:00
	Acabar implementar aplicació part client i servidor	12/12/12 8:00	29/12/12 16:00
	Generar versió beta producte.	29/12/12 8:00	30/12/12 0:00
	Elaboració de la memòria.	1/01/13 9:00	5/01/13 17:00
	Elaboració de la vídeo presentació.	5/01/13 9:00	6/01/13 0:00
	Lliurament final	7/01/13 9:00	7/01/13 17:00

Aquí presentem el diagrama de Gantt que resulta de la cronologia de les tasques anteriors:



5. Disseny centrat en l'usuari

5.1 Usuaris i context d'ús.

L'aplicació que es vol desenvolupar està molt dirigit a un tipus d'usuari, principalment el que realitza algun tipus d'esport de forma habitual i que participa en curses. Així que el contingut de la metodologia escollida l'enfocarem molt cap aquest perfil i intentarem veure quin és el model d'usuari i les seves necessitats.

Per realitzar la indagació, s'ha plantejat realitzar una enquesta¹ amb l'objectiu final de recopilar informació per l'aplicació que es vol crear. Dins d'aquesta enquesta, s'inclou un apartat més caracteritzat com un *benchmark*², posant d'exemple un portal Web amb funcionalitats similars a la nostra pròpia *app*³.

La enquesta creada junt amb l'apartat corresponent al *benchmark*, s'ha compartit i distribuït a través de la plataforma *Google Docs*⁴, via correu electrònic i xarxes socials (mitjançant la pertinent URL). Així doncs, es distribueix a qualsevol usuari independentment de l'objectiu que es persegueix.

S'opta per aquesta metodologia conjunta perquè és una manera molt pràctica d'aconseguir recollir informació quantitativa i prou representativa, i la organització no requereix massa temps. Però cal recalcar que prèviament s'han realitzat petites entrevistes personals amb possibles usuaris, que han ajudat a elaborar la enquesta.

5.1.1 Perfils d'usuari

De les més de 50 respostes d'usuaris a l'enquesta, podem centrar-nos en els resultats específics que interessin pel nostre projecte i recordem que responen a aquesta representació de població, no al global:

- El 92% dels enquestats realitza algun tipus d'esport. Per ordre de rellevància: Córrer, Bicicleta i Marxa. Gairebé la meitat realitza més d'un esport a la vegada.
- De tots els enquestats un 30% realitzen curses en algun esport.
- El rang d'edat d'usuaris que realitzen curses comprèn dels 21 als 52 anys i majoritàriament són del gènere masculí (88% aprox.).
- El 99% d'usuaris que realitza curses, busca informació referent a curses i inclús el 14% d'usuaris que no realitza curses també en busca, ja sigui perquè en algun moment té previst fer-ne o bé perquè necessita aquesta informació per altres àmbits.
- Principalment la informació es busca a través de Webs (84%) o del boca a boca (70%). En menor mesura a través de revistes, TV i altres mitjans.
- Gairebé la meitat d'usuaris que busca informació té alguna dificultat a l'hora de fer-ho i fem un recull general de les més comentades:
 - Trobar un calendari de curses global i complet, estan molt disperses.
 - Dificultat en el registre i inscripció a una cursa així com la data límit per fer-ho.

¹ El model d'enquesta es pot veure l'Annex d'aquest document.

² Metodologia d'anàlisi comparativa.

³ Aplicació per a mòbil.

⁴ Plataforma Google per crear i compartir diferents tipus de documents.

- Poca claredat i rigor de la informació trobada.
- Trobar de manera fàcil curses d'una determinada zona, ubicació...
- Dificultat per trobar informació relativa al circuit, participació, resultats, dificultat de la mateixa o a qui va adreçada.
- Els interessos o motivacions generals de l'usuari es poden agrupar en els següents:
 - Calendari de curses.
 - Zona i ubicació.
 - Dia i hora de la cursa.
 - Circuit, km, desnivell, perfil, dificultat de la cursa.
 - Informació de la inscripció i preu de la cursa.
 - Organitzador de la cursa.
- Altres interessos més específics però a tenir en consideració:
 - Indicacions per arribar a la cursa.
 - Participació i classificacions de la cursa.
 - Material necessari per la cursa.
 - Valoració dels corredors.
 - Obsequis de la cursa.
 - Historia de la cursa.
 - Estadístiques de la cursa.
 - Climatologia prevista.

Ens trobem amb un primer perfil d'**usuari esporàdic** que és una persona de qualsevol edat que no consumeix productes similars a l'app o relacionats amb l'esport, però té un interès directe o indirecte amb el món. El seu objectiu és trobar informació puntual o inclús iniciar-se en aquest món. Per tant necessitarà usar algunes de les funcionalitats.

Tenim un altre perfil d'**usuari esportista**, persones que avarca la **jove i mitjana edat** que realitza esport de forma habitual de manera **amateur o professional**. Té interès per aplicacions relacionades amb el món dels esports i en bona part per les curses que es realitzen. Disposa de dispositiu tipus Smartphone majoritàriament amb Android, i estaria interessat en disposar d'una aplicació que li permetés tenir informació de curses a través d'un calendari.

Hi ha un altre perfil d'usuari important però que no s'estudiarà en aquest projecte, el de **l'organitzador** i per tant creador de les curses. Aquest usuari és un organitzador d'esdeveniments com són les curses i té l'objectiu de difondre la informació relativa a aquests, al major nombre d'usuaris que puguin estar interessats en obtenir informació i participar-hi. Inicialment només es busca un sol administrador i un sol *partner*⁵ organitzador que hi vulgui col·laborar. Es mantindrà contacte amb diferents associacions, federacions i clubs per intentar trobar-ne algun.

⁵ Col·laborador i distribuïdor de la nostra app.

5.1.2 Context d'ús

L'objectiu de l'app que es vol desenvolupar és que sigui una font d'informació permanent i que pugui usar-se en qualsevol context i espai de temps, ja que precisament aquest és l'esperit dels dispositius mòbils i creiem que la filosofia dels seus usuaris.

Si ens fixem en els resultats de la segona part de l'enquesta, podem veure els indicadors que van en la línia del que comentàvem. Un 90% dels enquestats disposa d'un dispositiu de tipus Smartphone que permet, entre moltes coses, instal·lar aplicacions. La majoria de dispositius són amb plataforma Android 72%, seguit de iOS 21% i la resta 7% repartits entre altres plataformes.

Per altra banda hi ha una pregunta exclusivament relacionada amb on s'utilitza normalment el mòbil i és evident que els usuaris el fan anar en qualsevol lloc. Tan sols la minoria ho fa a casa o la feina per exemple. També una minoria l'usa de manera esporàdica o poc freqüent.

5.1.3 Requisits funcionals

Podem enumerar les diverses tasques dels dos perfils d'usuari principals:

Tasques usuari

CrearUsuari

Crear un compte propi d'usuari associat a un correu electrònic, per poder accedir després a l'aplicació. L'usuari passarà a estar registrat.

AccedirApp

L'usuari registrat pot accedir a l'aplicació a través de descriure el Mail usat en el registre i la contrasenya proporcionada pel sistema.

BuscarCursa

L'usuari registrat ha de poder buscar i seleccionar una cursa dins del calendari. Disposa de filtres per acotar i facilitar la búsqueda.

DetallarCursa

L'usuari ha de poder seleccionar una cursa per visualitzar-ne la informació detallada.

BuscarResultat

L'usuari vol cercar un resultat personal d'alguna cursa realitzada. Disposa de filtres per acotar i facilitar la búsqueda.

CrearResultat

L'usuari ha de poder incloure un resultat personal d'una cursa que ha realitzat. Ho fa a partir de les curses existents al calendari que prèviament pot buscar.

EliminarResultat

L'usuari elimina un resultat que havia afegit.

EditarUsuari

L'usuari ha de poder veure i modificar les dades del seu perfil i compte dins de l'aplicació.

ConfigurarOpcions

L usuari ha de poder modificar les opcions de configuració generals de l'app.

Tasques organitzador

Aquestes tasques es realitzaran a la part del sistema servidor, i per tant no apareixeran inicialment com opcions funcionals dins de l'app. L'objectiu és establir un vincle amb la base de dades de cada organitzador de curses, però es proporcionarà funcionalitats addicionals per permetre les següents tasques:

CrearCursa

L'usuari organitzador crea de forma manual una cursa per afegir-la al calendari de curses.

EditarCursa

L'usuari organitzador modifica les dades referents a una cursa del calendari.

EliminarCursa

L'usuari organitzador elimina una cursa del calendari.

5.1.4 Diagrames casos d'ús

Diagrama casos d'ús Usuari

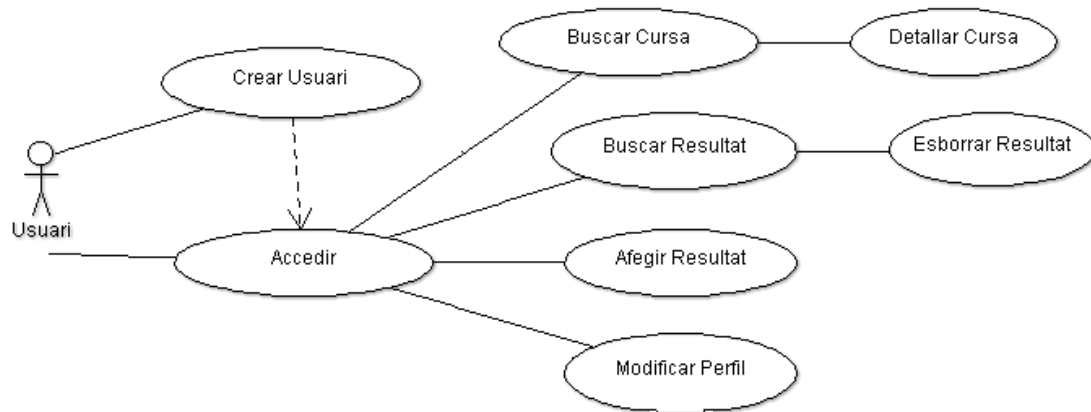
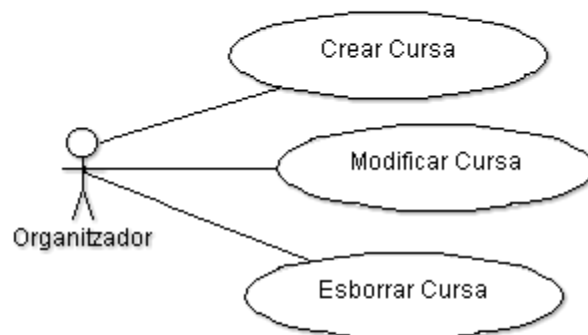


Diagrama casos d'ús Organitzador



5.2 Disseny conceptual

5.2.1 Escenaris d'ús

Usuari esporàdic

Les necessitats poden sorgir en l'entorn laboral o en algun esdeveniment específic i de manera puntual. El seu objectiu es recopilar informació d'un esdeveniment concret encara que no hi ha de participar directament, per tant la necessitat exclusiva serà buscar informació d'una cursa. Això ho farà accedint al calendari de curses i seleccionant la informació d'una cursa en concret.

Usuari amateur i professional

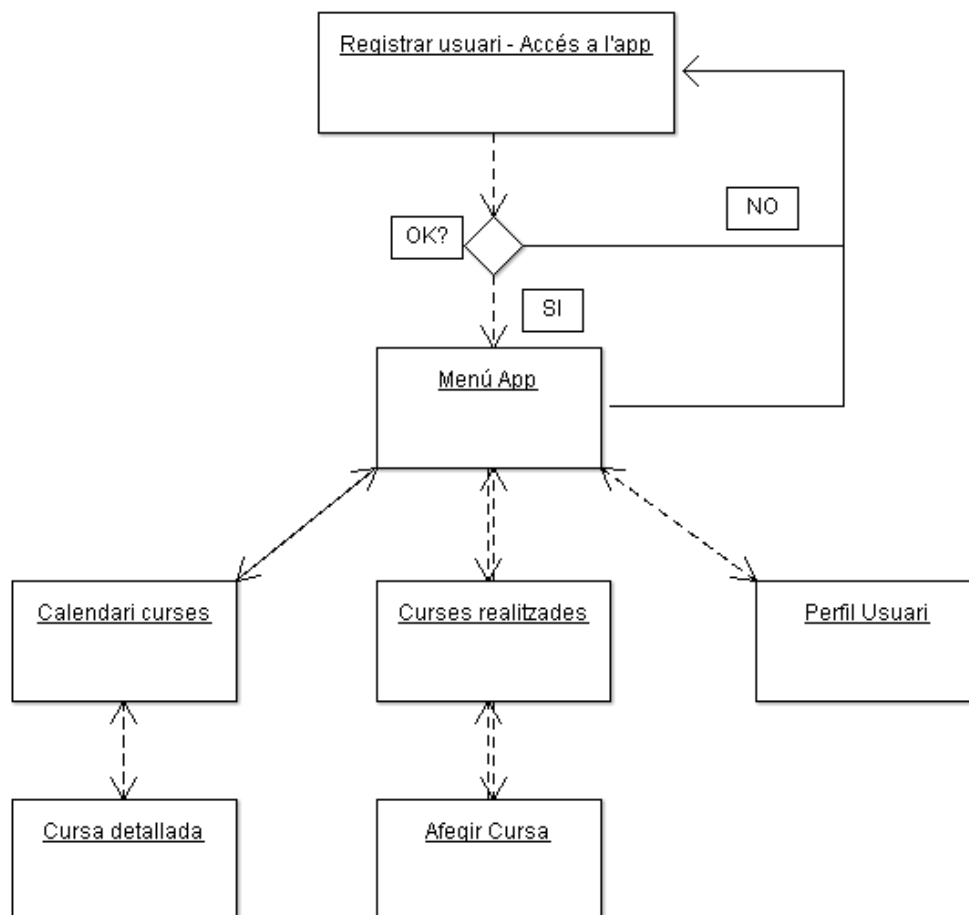
Es mou en entorns on habitualment es practica esport, com ara un club. L'usuari necessita saber amb antelació quines seran les properes curses que es realitzen a prop seu, perquè vol planificar en quines participarà, per tal de preparar-les convenientment. El seu objectiu és conèixer i posseir la informació necessària relativa a curses d'una manera integrada i suficientment detallada, no tan sols per ell sinó per poder transmetre-la a altres companys. Per aconseguir-ho, realitzarà tasques de cerca al calendari i detall d'informació de cada cursa que li interessi. També necessita tenir un historial de curses realitzades perquè vol fer un seguiment de les fites que ha assolit de cara a les motivacions personals o resultats competitius, i per fer-ho haurà d'afegir els resultats de les curses realitzades. Per altra banda, li interessa tenir un perfil de corredor amb la informació personal i de servei que voldrà guardar i recuperar en qualsevol moment.

Organitzador

Usuari que crea i organitza esdeveniments durant l'any i per tant en el seu entorn laboral o de club necessita donar a conèixer i fer córrer informació idònia de les seves curses. L'objectiu és arribar a tots els perfils d'usuari interessats mínimament i per tant la seva tasca és alimentar el sistema de l'app amb aquesta informació, i poder gestionar-la amb les funcionalitats mínimes: crear noves curses al calendari, modificar-les o eliminar-les. Això pot requerir en un moment donat poder fer-ho inclús des de la pròpia app, però inicialment ho realitzarà des del sistema servidor.

5.2.2 Fluxos d'interacció

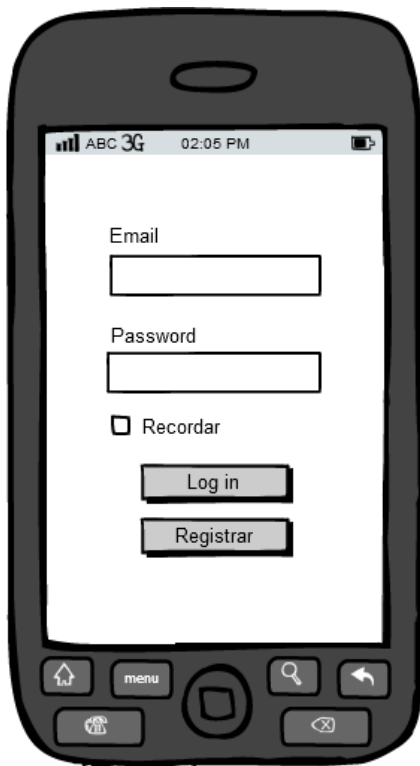
A continuació es mostra un diagrama de flux d'interacció entre l'usuari i les diverses pantalles, que representa la navegació per l'aplicació per realitzar les tasques necessàries.



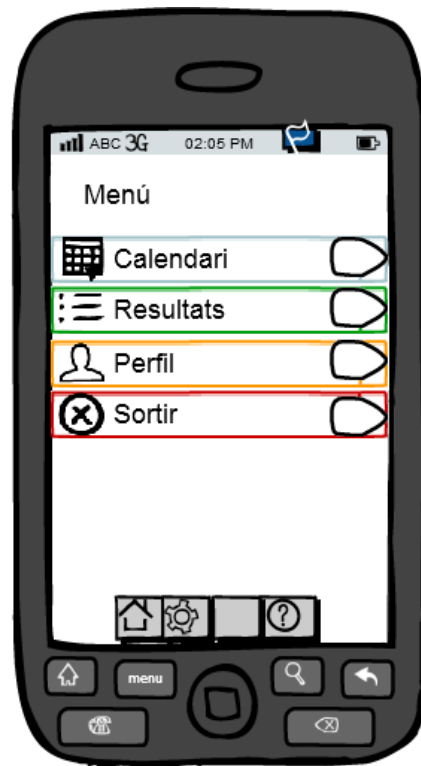
5.3 Prototipatge

5.3.1 Sketches

Aquests són els esbossos de les pantalles que s'han ideat en primera instància i que s'ha presentat a un cert nombre d'usuaris que han realitzat l'enquesta i han mostrat interès en donar la seva opinió respectiva:



1. Pantalla d'accés a l'aplicació



2. Pantalla de menú inicial



3. Pantalla per visualitzar calendari de curses a peu



4. Pantalla per visualitzar i compartir resultats



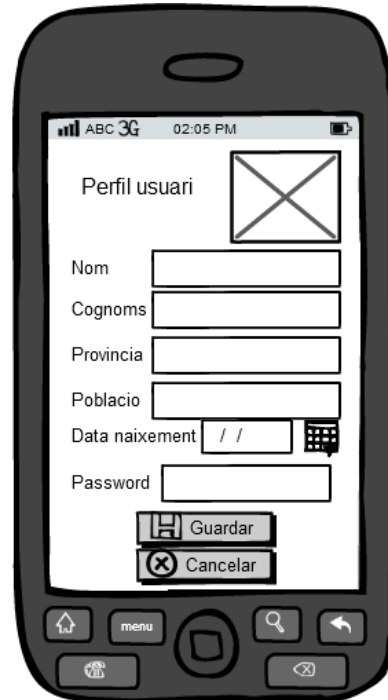
5. Pantalla per visualitzar i compartir cursa



6. Pantalla per afegir una cursa



7. Pantalla per afegir un resultat de cursa



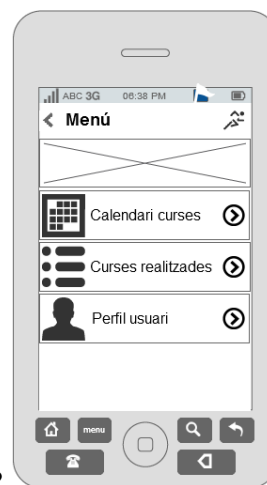
8. Pantalla de modificació del perfil usuari

5.3.2 Prototip alta fidelitat

A continuació es presenta el prototipus horitzontal d'alta fidelitat estàtic que s'ha dissenyat a partir del feedback dels usuaris que han participat i han vist els esbossos de les pantalles, la numeració correspon a cada pantalla sketch:



1.



2.



3. 
4. 
5. 
6. 
7. 
8. 

5.3.3 Avaluació

Per realitzar l'avaluació de l'app a través del test d'usuaris, haurem de buscar els candidats més adients que siguin potencials usuaris i sobretot tinguin força interès en col·laborar i usar-la. Per aquest motiu, farem un recull d'informació de l'usuari més específic que respon afirmativament:

- Realitza esport i curses?
- Li interessa trobar informació sobre curses?
- Li interessa gestionar informació relacionada amb les curses?
- Disposa d'un dispositiu amb plataforma Android?
- Acostuma a descarregar i provar aplicacions per mòbil?
- Li interessa una app amb les característiques descrites?
- Li motiva poder testear l'app descrita?
- Disposa de temps per col·laborar amb el test de l'app descrita?

Per portar a terme l'avaluació, s'escolliran 10 usuaris potencials que responguin a les anteriors preguntes i que mostren total disposició a participar en el test de l'app. Es demanarà el model del dispositiu i la versió de sistema operatiu amb totes les característiques necessàries per fer l'estudi i estadística dels errors que apareguin.

Per facilitar la feina, es crearà de la manera més senzilla un fòrum on s'obriran diferents fils per poder contemplar i debatre conjuntament cada part del test.

La dinàmica del test serà sempre la mateixa i es realitzarà de manera iterativa amb un temps limitat per mantenir el feedback refrescat i que no es converteixi en una feina feixuga. Per una banda el desenvolupador realitzarà els següents passos:

- a) Es penja una versió beta de l'app a algun sistema per compartir-la.
- b) Es donen les indicacions concretes als usuaris de la versió a descarregar, les funcionalitats que s'han de testear i el plaç de temps que tenen.
- c) Corregir errors i implementar les millores necessàries a l'app.

Les tasques de l'usuari dins el temps establert seran les següents:

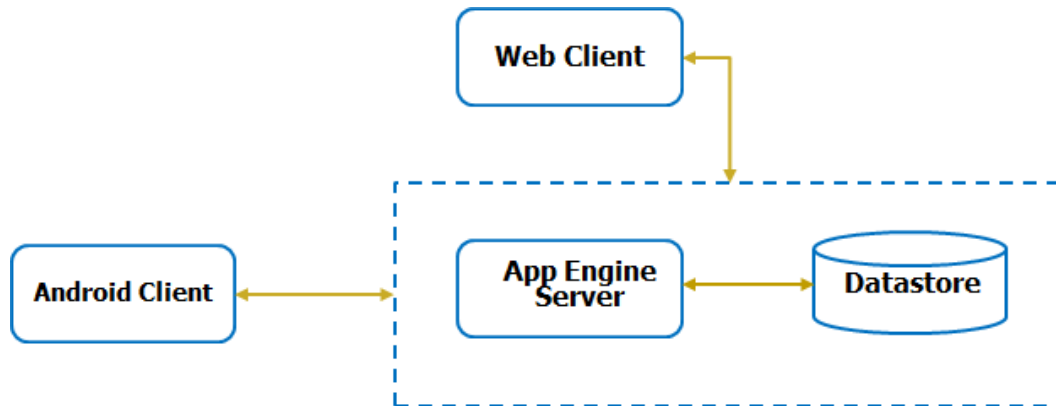
- a) Descarregar la versió, instal·lar-la o actualitzar-la.
- b) Provar exhaustivament les funcionalitats concretes que se li demana.
- c) Escriure al fil corresponent del fòrum els errors detectats, suggeriments, etc.

Quan l'usuari realitza el test pròpiament d'alguna funcionalitat, podem orientar i ajudar-lo per escriure de manera clara i concisa quins són els problemes que ha pogut detectar. Per exemple es pot formular que s'escriuin respostes d'acord a certes preguntes tals com:

- S'ha executat la funcionalitat?
- Ha realitzat la tasca esperada?
- Ha obtingut el resultat esperat?
- Ha estat àgil el procés?

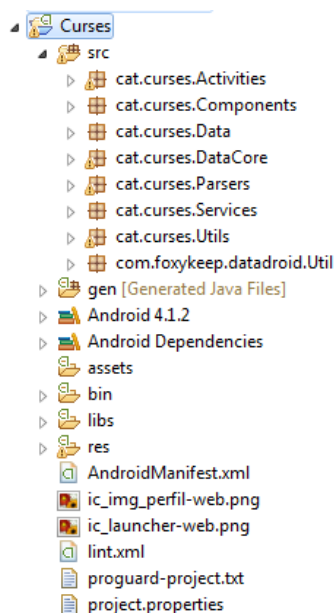
5.4 Arquitectura del sistema.

L'arquitectura del nostre sistema és basa en un model client-servidor que està format per l'aplicació Android Curses per la banda client i la plataforma Google App Engine amb el sistema d'emmagatzematge DataStore a la part servidor. A continuació podem observar un esquema que representa la nostra arquitectura:



5.4.1 Client: Android App Curses.

La part client es compon d'una aplicació desenvolupada en Java. L'entorn de programació Eclipse disposa dels complements necessaris per tal de poder crear la estructura bàsica de l'aplicació i disposar d'altres recursos com per exemple els AVD (dispositius Android Virtuals) que ens permetran poder realitzar proves i tests apart de fer-ho en un dispositiu real. A més podem instal·lar els SDK Android que creguem necessaris per usar les API de les noves funcions i tecnologies més recents del mercat, i altres complements extra a través del SDK Manager.



5.4.2 Servidor: Google App Engine i Datastore.

La plataforma de Google és una té la avantatja de comptar amb una infraestructura d'alt rendiment amb servidors distribuïts que ofereixen una sèrie de serveis en línia i ens allibera de molta feina en aquest sentit. Es una llançadora per poder crear i albergar aplicacions Web de manera fàcil, ràpida i escalable, amb un cost zero inicial.

Per guardar les dades persistents de la nostra aplicació Android client, hem optat per usar la infraestructura Datastore, que no és exactament una base de dades relacional. El Datastore guarda objectes coneguts com *entities*⁶ que tenen una o varies propietats amb un o varis valors. És un model semblant al relacional però no té totes les característiques, i per tant més restrictiu, però suficient. La peculiaritat és que el Datastore està creat sobre una infraestructura d'emmagatzematge que és il·limitat i que disposa de balanceig de càrrega i alta redundància (HRD) per les aplicacions que s'hi alberguen.

A través del nostre entorn de programació Eclipse i el *plugin*⁷ per a Google (GPE), disposem de les API de Google, en concret farem servir el Persistence Manager, que és una llibreria que usa l'API del Datastore, a més alt nivell i la connexió directa entre els dos entorns mitjançant un compte Google nou creat: *uoccurses@gmail.com*.

Així doncs des d'aquest propi entorn a Eclipse, podem crear una aplicació Web client de suport, que ens permetrà crear el model de dades persistents i algunes funcions específiques per la part servidor i ho farem usant els estàndards de programació Java. Bàsicament ens proporciona la JVM, interfícies Servlet i interfícies estàndard com JDO enfocades a desenvolupar la nostra aplicació Google App Engine sobre el Datastore.

L'entorn de desenvolupament també ens permet executar una replica del servei a la nostra maquina local per tal de realitzar les proves pertinents a la nostra aplicació web abans de desplegar-la de manera còmoda i directa a través del plugin als servidors de Google.

Altres punts a favor d'usar aquesta plataforma, és que Android està íntimament lligada amb Google i té facilitats per obtenir altres serveis integrats com l'autenticació d'usuaris amb Google Accounts, missatgeria Gmail, portal d'administració o accés al mercat per aplicacions Android PlayStore.

⁶ Classes que representen els objectes persistents al Datastore.

⁷ Component afegit a un software per realitzar funcions especials de tercers.

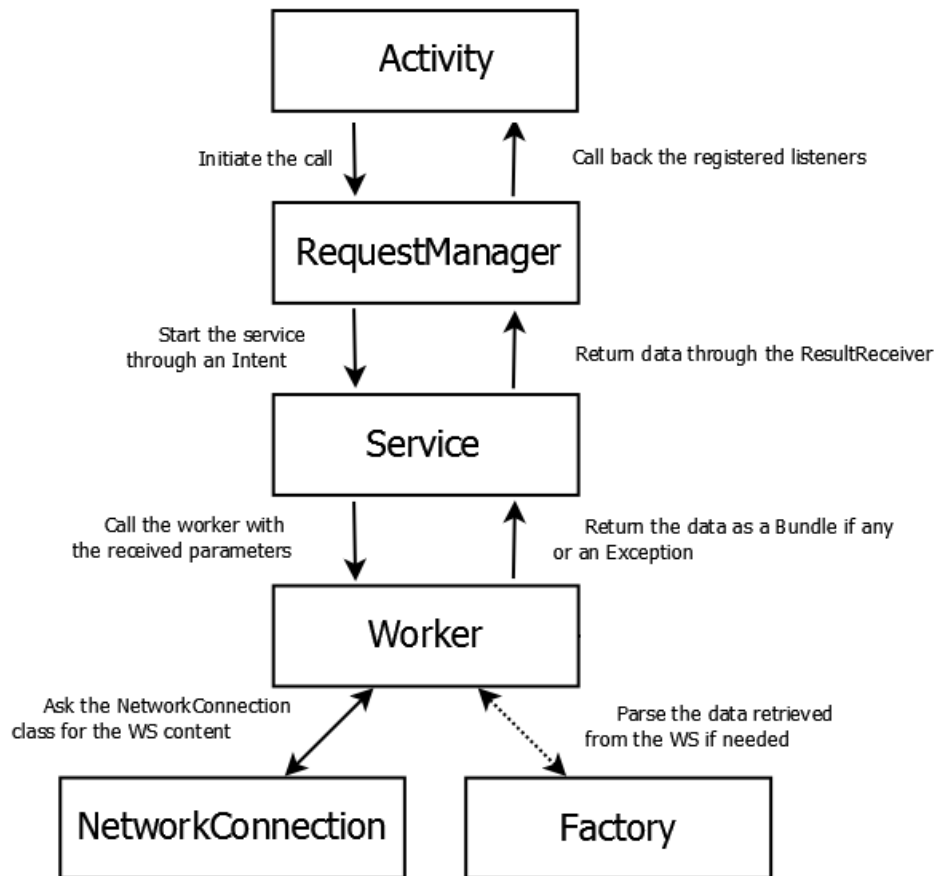
5.4.3 Client – Servidor: Connexió.

El sistema Google App Engine disposa de diverses formes per accedir a la informació emmagatzemada al Datastore (BigTable) i una d'elles ja l'hem comentat, és a través d'una aplicació client web desplegada a App Eng, que serveix les peticions invocant els *servlets*⁸ que proporciona un descriptor del directori Web.

Però per al cas principal, el nostre client particular d' Android, no hi ha possibilitat de realitzar operacions amb el Datastore directament. Entre les diverses opcions de connexió que existeixen, s'ha decidit fer-ho a través de crides als Web Services que es proporcionaran des de l'aplicació App Engine. Aquestes peticions es realitzaran sobre el protocol HTTP que és una manera senzilla de comunicar amb l'aplicació web.

S'usarà una llibreria lliure anomenada DataDroid com a base per implementar les diferents capes de comunicació a la part del client Android i de la qual s'adaptarà el codi per a que funcioni amb la resta de la estructura de l'aplicació.

L'esquema gràfic que representa les diferents capes segons el model seguit a la llibreria de la part client Android és la següent:



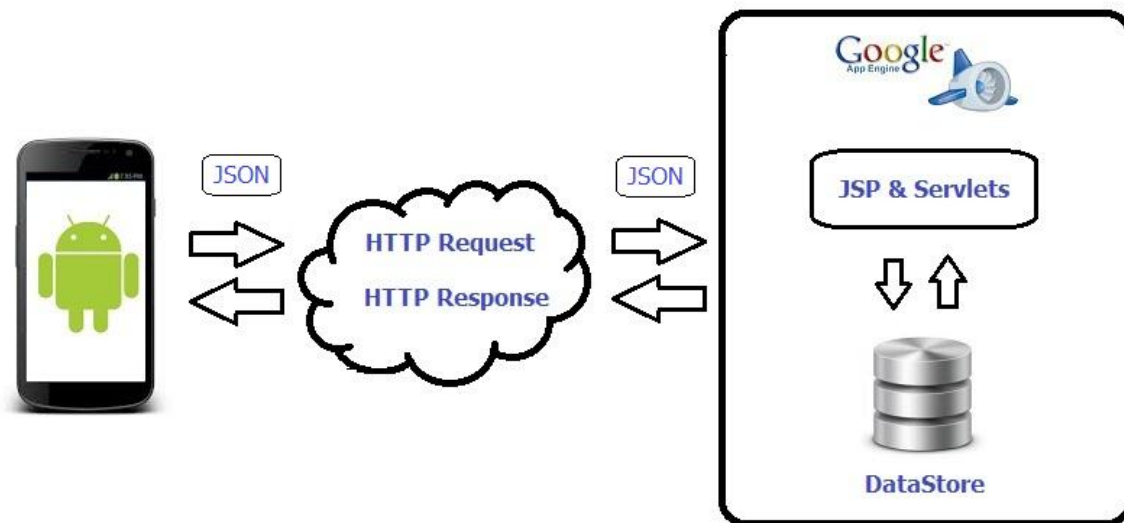
⁸ Petit programa en java que s'executa en un servidor web

Així l'aplicació Android obtindrà les dades fent peticions asíncrones i amb possibilitat de fer-les concurrentment (multithread) als diferents Web Services que s'oferirà des de la part servidor, en funció de l'ús que en faci l'usuari i les necessitats que tingui.

El llistat de crides genèriques possibles a la part servidor són els següents:

- **Autenticació:** Permetrà validar l'usuari de l'aplicació Android i assegurar que és ell amb aquesta aplicació qui intenta accedir als recursos, i no tercers.
- **Llistar:** Quan l'usuari vulgui veure el llistat de curses del calendari o resultats que ha afegit prèviament.
- **Llegir:** Poder obtenir les dades d'un objecte que s'ha seleccionat d'una llista o del menú del perfil d'usuari.
- **Afegir:** Crear un resultat i emmagatzemar-lo al Datastore.
- **Eliminar:** Eliminar un resultat del Datastore.
- **Modificar:** Guardar les dades del perfil de l'usuari a través de l'identificador de l'objecte que el representa.

Cada acció de l'usuari provoca a través de les diferents capes una crida a un Web Service que s'encapsula en una petició HTTP d'escriptura o lectura i que retorna una resposta que rebran els diferents *callback*⁹ de les capes de connexió. Finalment s'haurà de processar la cadena retornada en el format d'objecte JSON per poder fer un mapeig de cada valor/nom del resultat amb els diferents camp de l'objecte del nostre model de dades.



⁹ Fragments de codi executats com a resposta d'una crida inicial

5.6 Model de dades

5.6.1 Diagrama de classes

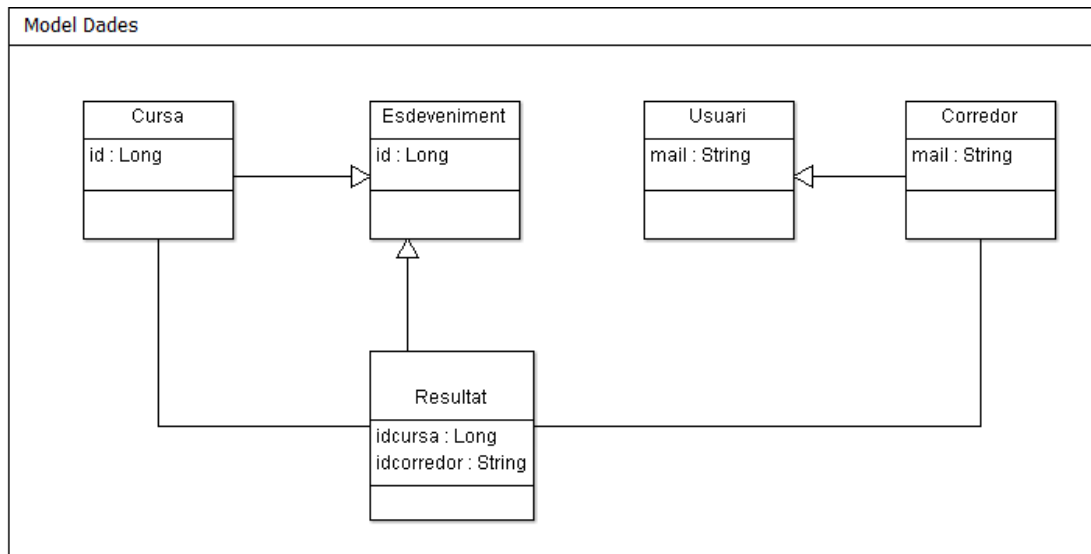
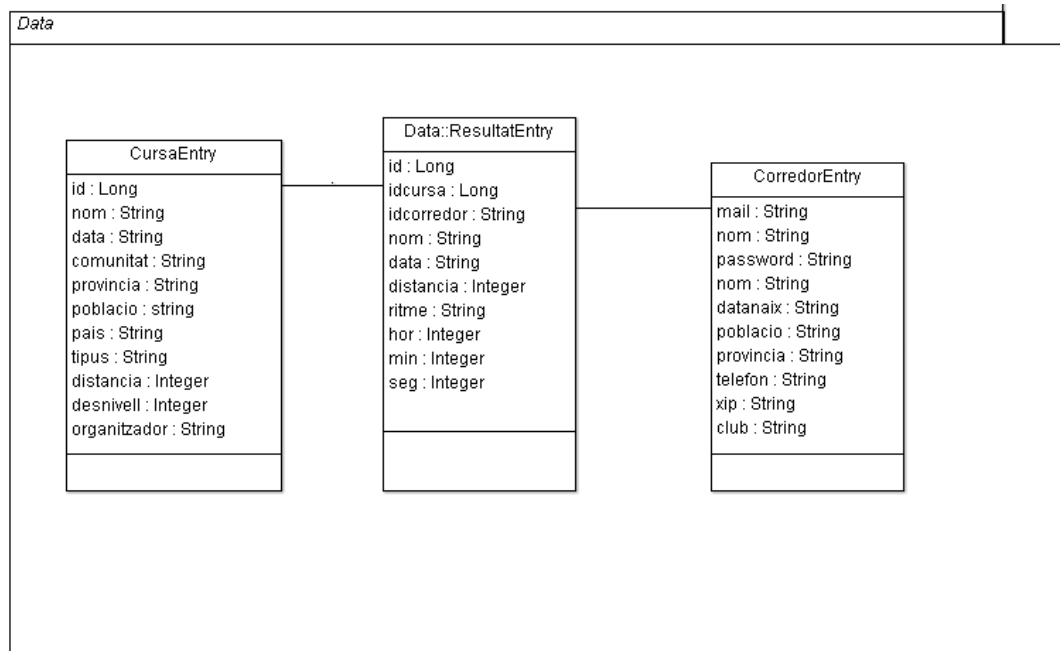


Diagrama de classes per al model de dades de l'APP. Es pretén crear un *framework*¹⁰ a nivell de dades (DataCore) per poder reutilitzar el model de dades i classes en altres lògiques de negoci. Al diagrama només s'especifica els atributs clau.

5.6.2 Base de dades Datastore



Disseny de les entitats creades per al nostre sistema de gestió de dades del Datastore.

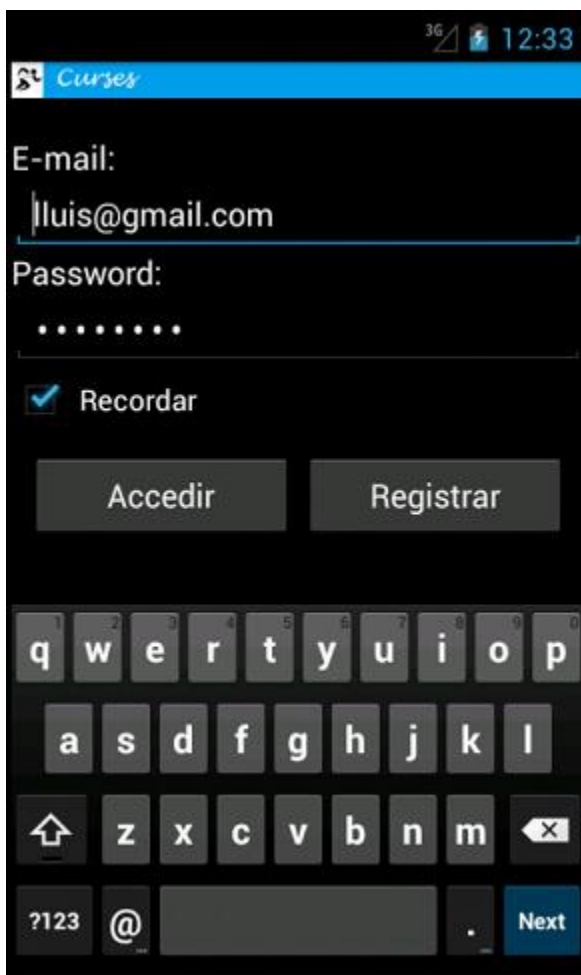
¹⁰ Plataforma universal de software reusable

5.7 Disseny pantalles

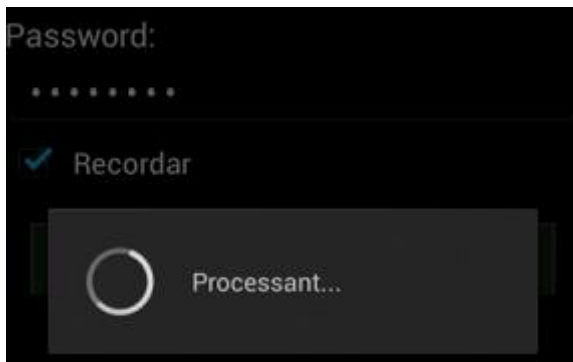
A continuació es presenta el disseny de les pantalles que s'ha realitzat a partir del prototipus horitzontal d'alta fidelitat estàtic. S'ha dut a terme de manera iterativa per cada una de les pantalles creades, canviant, afegint o prescindint d'alguns elements.

Pantalla de registre i accés a l'aplicació.

Pantalla per registrar l'usuari inicialment, i un cop es rep el correu amb l'usuari i el password, ja es pot accedir a l'APP. S'opta per permetre que es recordi la contrasenya de l'usuari per poder accedir més ràpidament, a pesar del risc de seguretat que comporta. En cas que no existeix l'usuari o no es posi el password correcte, el sistema ens avisarà i sinó accedim:

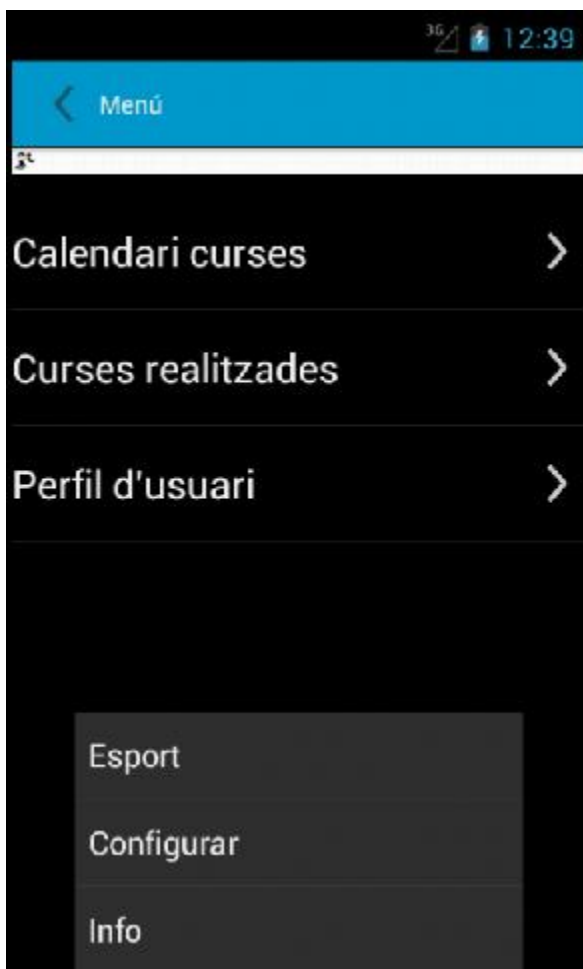


The screenshot shows a mobile application interface for 'Curses'. At the top, there's a status bar with '3G', a signal icon, and the time '12:33'. Below it is a blue header bar with the app name 'Curses' and a small icon. The main form has two input fields: 'E-mail:' with the text 'lluis@gmail.com' and 'Password:' with masked characters '.....'. Below the password field is a checkbox labeled 'Recordar' which is checked. At the bottom of the form are two buttons: 'Accedir' and 'Registrar'. A virtual keyboard is visible at the bottom of the screen, showing letters, numbers, and a 'Next' button.



Pantalla del menú inicial de l'aplicació.

Menú inicial de l'APP on apareixen les diferents opcions que podem escollir: calendari, resultats o perfil del corredor. Veiem desplegat el menú contextual on tenim el botó Multiesport que es desitja incorporar en el futur i permetrà canviar a l'esport corresponent, menú Configurar que pel moment no està habilitada (no cal guardar cap paràmetre) i un menú d'informació de l'aplicació. La barra d'acció ens permet sortir a la pantalla Login.



Pantalla calendari de curses.

Apareix un llistat amb les properes curses que es realitzaran. Tenim la possibilitat de filtrar a través d'un buscador que ho farà pel nom, els km i el tipus de cursa. També podem refrescar el llistat en qualsevol moment, i com a opcions futures podem acotar per un període. Al clicar a un element de la llista, obrirem la informació relativa a la cursa:



Pantalla de detall d'informació de cursa.

Pantalla on apareix la informació bàsica de la cursa i un camp Més Info hi poden aparèixer altres dades com per exemple links referents a pàgina web de l'esdeveniment.

En un futur si està disponible per part de l'organitzador, es podrà obrir la ruta a Google Maps i es podrà compartir a xarxes socials.



Pantalla de resultats de curses realitzades.

En aquesta pantalla es presenta el llistat de resultats que hem afegit de curses realitzades. Inicialment està buida i per tant amb el botó d'afegir ens obrirà el calendari de curses per poder seleccionar la que volem introduir i al clicar la llista mostrarà un *DialogBox*¹¹ per posar el temps en que s'ha realitzat la cursa. Automàticament calcula el ritme i afegeix el resultat. Podem fer selecció múltiple per poder eliminar els resultats que desitgem, amb el botó eliminar.



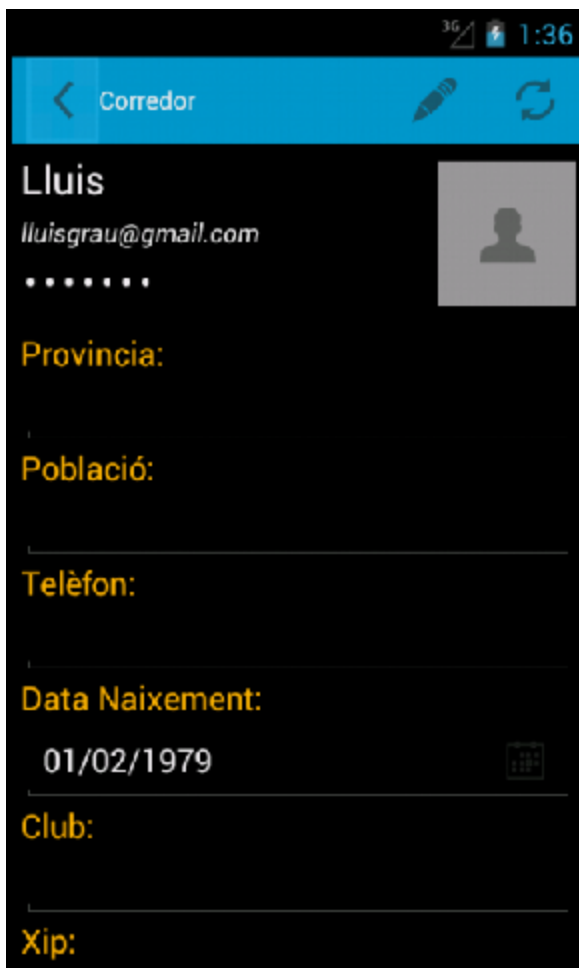
¹¹ Finestres de diàleg

Pantalla amb les dades del perfil de l'usuari.

Es separa les dades referents al perfil de l'usuari i les dades en referència al compte d'usuari que es poden modificar apart a través de la icona d'edició de la part superior. També hi trobem un botó per refrescar les dades en qualsevol moment.

Hi ha diversos camps que es poden afegir o modificar, i en concret tenim la data de naixement que ens obrirà un selector de data per facilitar-nos la feina.

En un futur es desitja incorporar la foto del usuari corredor.



The screenshot shows a mobile application interface for a runner's profile. At the top, there is a blue header bar with a back arrow, the text "Corredor", an edit icon (pencil), and a refresh icon (circular arrow). Below the header, the profile information is displayed: the name "Lluís", the email "lluisgrau@gmail.com", and a series of six dots. To the right of this information is a placeholder for a profile picture, represented by a grey silhouette icon. Below the profile information, there are several form fields with labels in yellow text: "Provincia:", "Població:", "Telèfon:", "Data Naixement:", "Club:", and "Xip:". The "Data Naixement:" field is filled with the date "01/02/1979" and has a small calendar icon to its right. The background of the form is dark grey.

Wed, Feb 1, 2012

Data de naixement:

Jan	29	2011
Feb	01	2012
Mar	02	2013

Done

Al clicar al botó d'editar de la barra d'acció, apareix un Dialogbox on podem introduir el nostre nom i el password si volem canviar-lo. Seguidament OK

< Corredor

LLUIS

lluiscrau@gmail.com

Escriu el teu nom i

Nom:

Password:

Repeteix Password:

OK Cancelar

Un cop haguem fet les modificacions necessàries, trobarem al final del llistat un botó per guardar les dades, i si el procés és correcte, mostrarà un text conforme s'han guardat.

Xip:

Guardar

6. Implementació

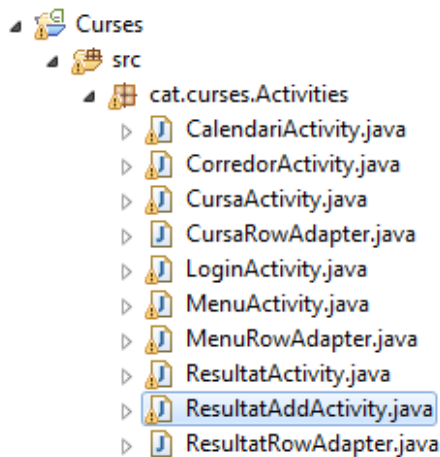
Durant el procés d'implementació s'han hagut de prendre decisions per tal de dur a terme el projecte en el termini previst, i s'han descartat algunes funcionalitats inicials o s'han resolt diferent.

6.1 Part client Android

En aquest apartat s'expliquen els punts més rellevants de la implementació de l'aplicació i que permeten tenir una visió global d'aquesta i de les diferents parts realitzades. Farem una descripció dels diferents paquets que conformen la part client amb les seves classes i seguidament veurem la part dels fitxers XML de l'aplicació.

Cal dir que el codi font de l'aplicació està degudament comentat i per tant allí es pot veure detalladament els aspectes més tècnics.

Paquet cat.curses.Activities:

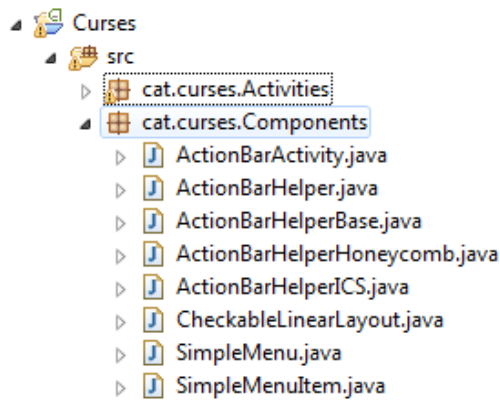


Aquest és el paquet de classes que gestionen les pantalles (Activities) i per tant que controlen les accions de l'usuari al interactuar amb elles. Hi ha dos tipus generals de pantalles, com hem vist anteriorment, la de llistats com CalendariActivity i les personalitzades com el LoginActivity.

Les pantalles de llistats incorporen un layout personalitzat per dibuixar cada fila i per tal de carregar les dades, s'ha creat un adaptador específic com per exemple ResultatRowAdapter.

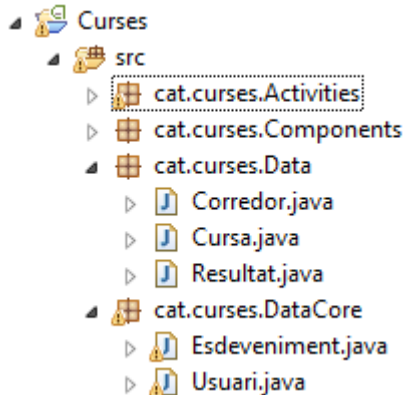
Val a dir que totes les pantalles a excepció de la d'accés a l'app tenen implementada una barra d'acció per fer la navegació i les accions que pren l'usuari de manera més senzilla. També tenen un control del cicle de vida de l'aplicació captant els events i l'estat en tot moment.

Paquet cat.curses.Components:



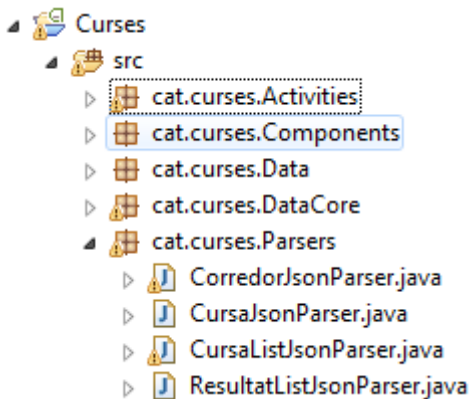
Són components bàsicament per a fer compatible la ActionBar (barra d'acció) amb altres versions anteriors d'Android, ja que en les versions noves, tan sols cal aplicar un tema que la suporti i es genera automàticament (després cal inflar-hi els menús contextuels).

Paquet cat.curses.Data:



Aquestes són les tres classes del nostre model de dades que s'estenen de les superclasses DataCore per poder reutilitzar en altres aplicacions. Consten d'un constructor, atributs i els respectius *getters* i *setters*.¹²

Paquet cat.curses.Parsers



¹² Funcions per llegir i escriure els atributs d'un objecte d'una classe

Les classes del paquet Parsers són exclusivament fetes per lligar les dades que ens arriben del servidor via les peticions HTTP i que estan en format JSON. Podem veure la classe que mapeja la cadena que arriba en una resposta a un llistat de Curses:

```
public class CursaListJsonParser {

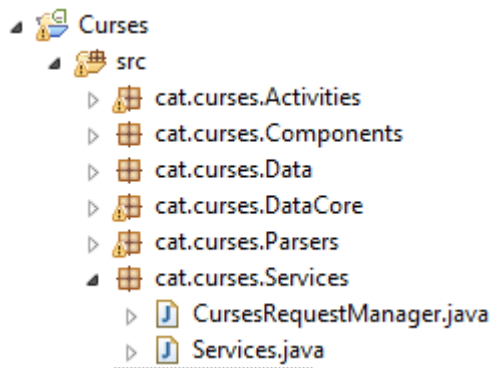
    private static String TAG = CursaListJsonParser.class.getSimpleName();

    public static ArrayList<Cursa> parseResult(final String wsResponse) throws JSONException {
        final ArrayList<Cursa> cursesList = new ArrayList<Cursa>();
        final JSONObject parser = new JSONObject(wsResponse);
        final JSONArray jsonCursaArray = parser.getJSONArray("curses");
        final int size = jsonCursaArray.length();
        for (int i = 0; i < size; i++) {
            final JSONObject jsonCursa = jsonCursaArray.getJSONObject(i);
            final Cursa cursa = new Cursa(jsonCursa.getLong("id"));
            //si existeixen parametres no obligatoris
            cursa.setNom(jsonCursa.optString("nom"));
            cursa.setData(jsonCursa.optString("data"));
            cursa.setHora(jsonCursa.optString("hora"));
            cursa.setPoblacio(jsonCursa.optString("poblacio"));
            cursa.setProvincia(jsonCursa.optString("provincia"));
            cursa.setComunitat(jsonCursa.optString("comunitat"));
            cursa.setDistancia(jsonCursa.optInt("distancia"));
            cursa.setDesnivell(jsonCursa.optInt("desnivell"));
            cursa.setOrganitzador(jsonCursa.optString("organitzador"));
            cursa.setTipus(jsonCursa.optString("tipus"));

            cursesList.add(cursa);
            Log.i(TAG, "Cursa: " + cursa.toString());
        }

        return cursesList;
    }
}
```

Paquet *cat.curses.Services*:



Aquestes dues classes són força importants ja que com hem vist en la il·lustració de la arquitectura del client, són les que ens permeten crear les funcions que realitzen les crides via HTTP i que posteriorment reben la informació a través dels diferents callback que implementen i que es disparen quan succeeix un algun event. S'ha de destacar que s'han adaptat de la llibreria Datadroid per a que es puguin fer crides de l'estil CRUD (create, read, update, delete) d'una manera integrada en una mateixa funció passant per paràmetre l'acció que interessa en cada moment, i per altra banda es fan les crides de recuperació de llistats.

CursesRequestManager s'encarrega d'agafar les peticions que fan les pantalles i en funció del tipus de dada i l'acció, revisa que no s'estigui ja fent la mateixa crida i passa tots els paràmetres necessaris a la classe Services:

```
public int crudCorredor(final String reqTipus, final String mail, final String password, final String nom,
    final String datanaix, final String provincia, final String poblacio, final String telefon,
    final String club, final String xip) {

    // Comprovar que el tipus Request que es demana no s'esta duplicant pel mateix corredor
    final int requestSparseArrayLength = mRequestSparseArray.size();
    for (int i = 0; i < requestSparseArrayLength; i++) {
        final Intent savedIntent = mRequestSparseArray.valueAt(i);

        if (savedIntent.getIntExtra(Services.INTENT_EXTRA_WORKER_TYPE, -1) != Services.WORKER_TYPE_CORREDOR) {
            continue;
        }
        if (!savedIntent.getStringExtra(Services.INTENT_EXTRA_REQUEST_TYPE).equals(reqTipus)) {
            continue;
        }

        if (!savedIntent.getStringExtra(Services.INTENT_EXTRA_CORREDOR_MAIL).equals(mail)) {
            continue;
        }

        return mRequestSparseArray.keyAt(i);
    }

    final int requestId = sRandom.nextInt(MAX_RANDOM_REQUEST_ID);

    // Creem l'Intent amb els parametres del Request i engegarem el service
    final Intent intent = new Intent(mContext, Services.class);
    intent.putExtra(Services.INTENT_EXTRA_WORKER_TYPE, Services.WORKER_TYPE_CORREDOR);
    intent.putExtra(Services.INTENT_EXTRA_RECEIVER, mEvalReceiver);
    intent.putExtra(Services.INTENT_EXTRA_REQUEST_ID, requestId);
    intent.putExtra(Services.INTENT_EXTRA_REQUEST_TYPE, reqTipus);
    intent.putExtra(Services.INTENT_EXTRA_CORREDOR_MAIL, mail);
}
```

La capa de serveis s'encarrega de realitzar les crides en tasques asíncrones, per evitar congelar l'aplicació i invoca una altra capa que és la de connexió passant-li novament els paràmetres però ja en un format que sigui acceptat pels HTTP Requests dins de la URL de connexió:

```
try {
    switch (workerType) {

        case WORKER_TYPE_CORREDOR:
            //Preparem parametres
            params.put("accio", intent.getStringExtra(INTENT_EXTRA_REQUEST_TYPE));
            params.put("mail", intent.getStringExtra(INTENT_EXTRA_CORREDOR_MAIL));
            params.put("nom", intent.getStringExtra(INTENT_EXTRA_CORREDOR_NOM));
            params.put("datanaix", intent.getStringExtra(INTENT_EXTRA_CORREDOR_DATANAIX));
            params.put("password", intent.getStringExtra(INTENT_EXTRA_CORREDOR_PASSWORD));
            params.put("poblacio", intent.getStringExtra(INTENT_EXTRA_CORREDOR_POBLACIO));
            params.put("provincia", intent.getStringExtra(INTENT_EXTRA_CORREDOR_PROVINCIA));
            params.put("telefon", intent.getStringExtra(INTENT_EXTRA_CORREDOR_TELEFON));
            params.put("club", intent.getStringExtra(INTENT_EXTRA_CORREDOR_CLUB));
            params.put("xip", intent.getStringExtra(INTENT_EXTRA_CORREDOR_XIP));
            //Realitzem la connexio i recuperem la resposta del webservice en un string dins el bundle data
            wsResult = NetworkConnection.retrieveResponseFromService(CORREDOR_URL,
                NetworkConnection.METHOD_GET, params);
            response.putString(CursesRequestManager.RECEIVER_EXTRA_CORREDOR, wsResult.wsResponse);
            Log.i(TAG, "wsResponse: " + wsResult.wsResponse);
            break;
    }
}
```

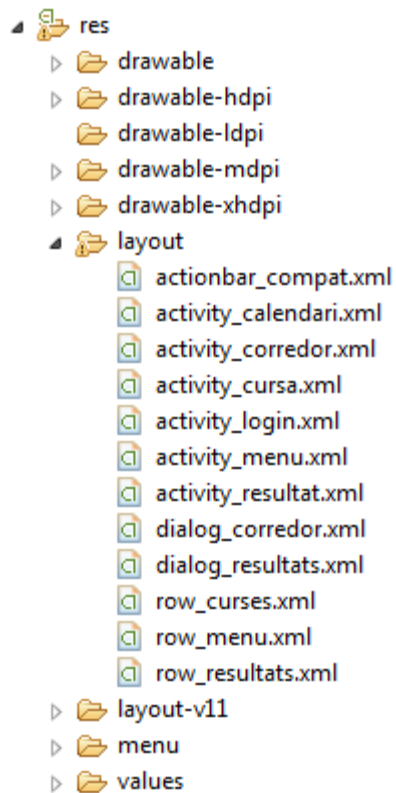
La resta de classes de la llibreria es troba al paquet datadroid.Util i no s'ha modificat.

També hi ha un paquet *cat.curses.Utils* que conté una classe amb mètodes per realitzar accions sobre objectes llista, i poder llegir, ordenar, consultar un array de la classe cursa.

Al directori de recursos (Res) mostrem com a principal el directori layout que conté les vistes corresponents a cada pantalla Activity de l'aplicació, és a dir , tots els widgets que incorpora cada pantalla amb els que l'usuari interactuarà. Veiem en particular els layout row_XXX.xml que fan referència als adaptadors de files dels llistats que hem comentat.

També tenim un directori de menú que conté cada botó dels menús contextuais que apareixen a les pantalles, ja sigui a la barra d'acció o desplegats a l'inferior de la pantalla.

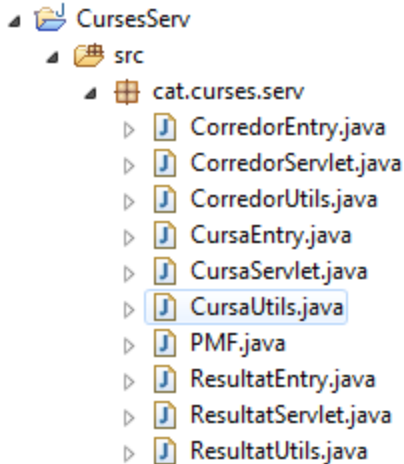
I dins de values, hi tenim els fitxers xml que defineixen colors, identificadors i cadenes de text, que ens permetrà en un futur aplicar multi idioma sense haver de tocar el codi font de l'app.



6.2 Part servidor Google App Engine

En l'apartat del servidor explicarem primer el paquet on definim les classes de les nostres entitats al Datastore i seguidament veurem la estructura de l'aplicació web i els fitxers més rellevants i útils. Per altra banda, mostrarem com hem creat l'aplicació a la plataforma de Google i com l'hem desplegat a través del plugin per a Eclipse.

Paquet *cat.curses.serv*:



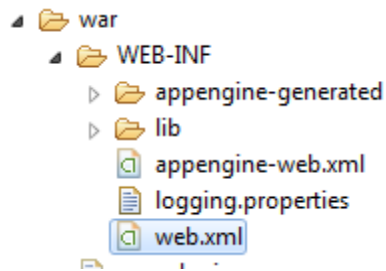
En aquest paquet tenim tres tipus de classes, que són les que modelen les nostres entitats *CursaEntry*, *CorredorEntry* i *ResultatEntry*. Per tal de rebre les crides que han de realitzar la persistència de les dades, la modificació o la eliminació, s'han creat *servlets* específics per cada entitat que cridaran les funcions implementades al respectius fitxers *xxxUtils* i aquests usaran la llibreria d'alt nivell *Persistence Manager Factory* per sobre de la API del Datastore:

```
public class CursaUtils {

    // Buscar cursa per id al Datastore
    public static CursaEntry getById(Long id) {
        PersistenceManager pm = PMF.get().getPersistenceManager();
        CursaEntry cursa = null;

        try {
            cursa = pm.getObjectById(CursaEntry.class, id);
        } catch (JDOObjectNotFoundException e) {
            return null;
        }
        return cursa;
    }
}
```

Aquests servlets són els punts d'entrada de les peticions que es fan via HTTP. L'enllaç entre les crides i les funcions que responen, es defineixen en el fitxer *web.xml* que s'alberga en el directori on resideix l'aplicació web:

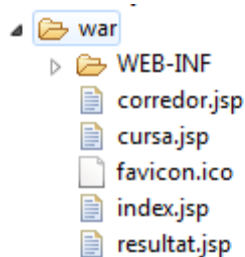


Veiem com definim el servlet i el seu mapeig tant a la classe com en el patró url que haurà de cridar un client que fa la petició.

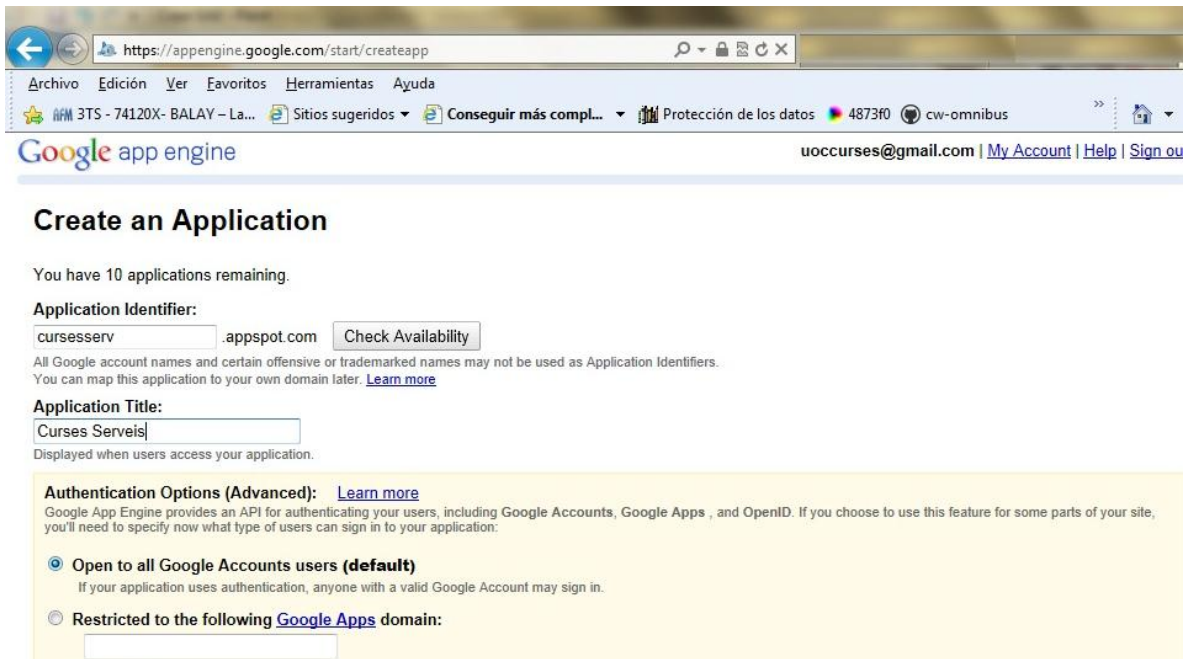
```
<servlet>
  <servlet-name>CorredorServlet</servlet-name>
  <servlet-class>cat.curses.serv.CorredorServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>CorredorServlet</servlet-name>
  <url-pattern>/corredor</url-pattern>
</servlet-mapping>
```

Els mateixos servlets proporcionen la resposta al client que fa la crida, en el nostre cas principalment el dispositiu Android i ho fa escrivint al canal de comunicació que existeix, i si s'escau la resposta es torna en format JSON per transmetre les dades demanades o sinó amb missatges de text de confirmació.

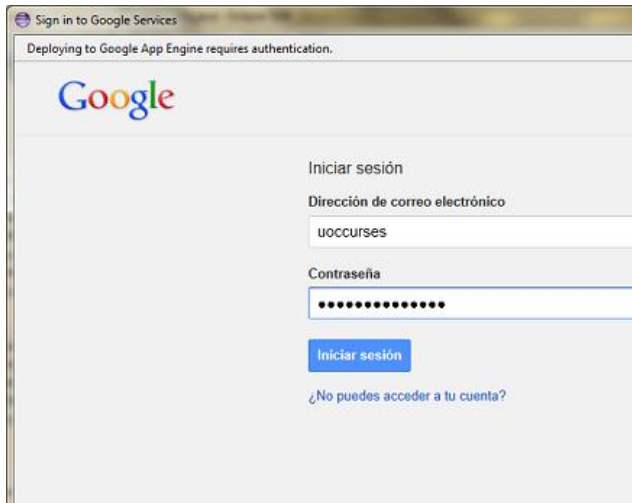
Hem dit que principalment els clients del Curses.Serv són els dispositius mòbils, però podem donar accés als organitzadors per exemple, per a introduir les curses i modificar-les des de la pròpia estructura web, mitjançant pàgines JSP per a tal efecte i que es poden executar des de qualsevol navegador web:



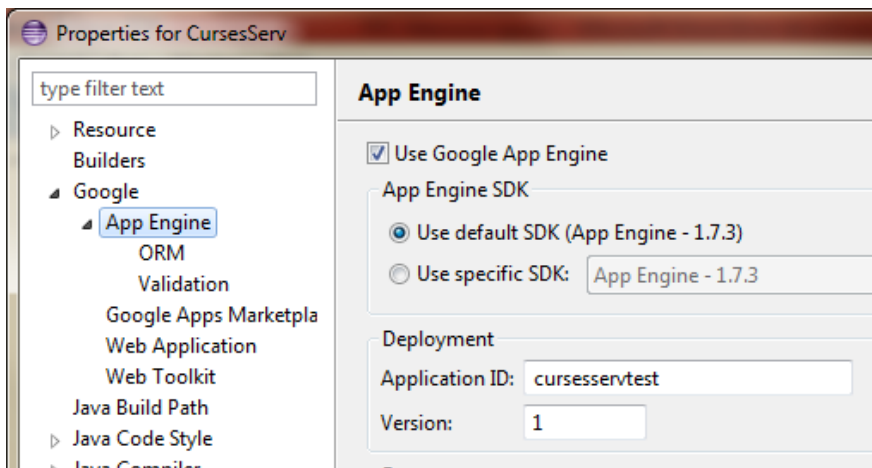
Com dèiem, un cop implementada la part servidor en el nostre entorn, cal crear l'aplicació App Engine a la plataforma de Google mitjançant un compte especial, uoccurses@gmail.com, i es realitza en dos passos. Hem creat una aplicació de test **cursesservtest** per realitzar les proves abans no es posi en producció:



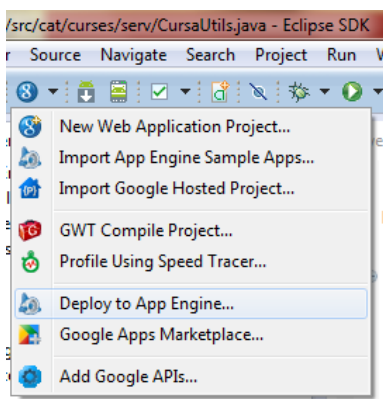
Quan l'aplicació ja està creada, només ens queda configurar l'entorn Eclipse per a que es vinculi primer amb el compte de Google:

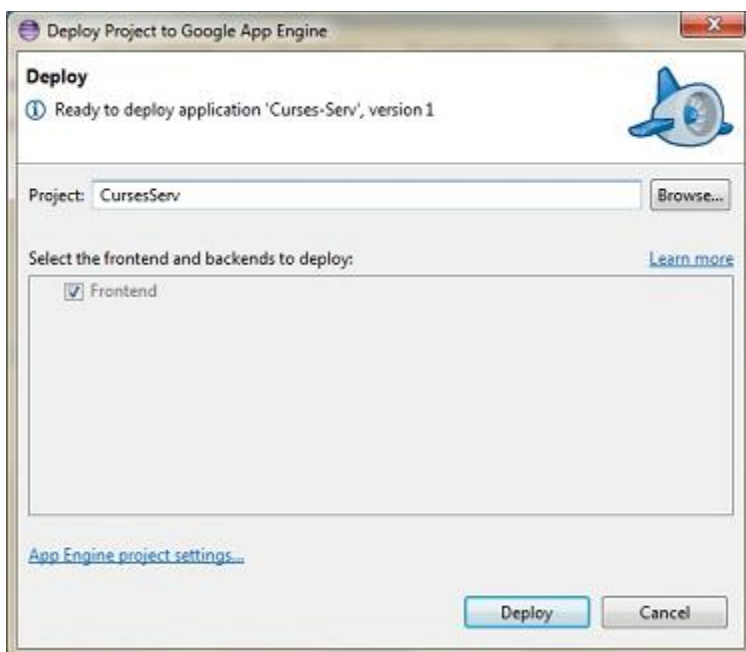


Es configura l'aplicació CursesServ per tal que es referencii l'aplicació Google App Engine que acabem de crear:



I es puguin realitzar finalment el desplegament:





```

Problems @ Javadoc Declaration Console Development Mode LogCat
CursesServ - Deploy to App Engine

----- Deploying frontend -----

Preparing to deploy:
  Created staging directory at: 'C:\Users\Usuario\AppData\Local\Temp\appcf8
  Scanning for jsp files.
  Compiling jsp files.
  Scanning files on local disk.
  Initiating update.
  Cloning 1 static files.
  Cloning 51 application files.

Deploying:
  Uploading 0 files.
  Initializing precompilation...
  Deploying new version.

```

El desplegament es farà efectiu si el procés és correcte, i podrem comprovar-ho accedint a la URL que ens proporciona Google per la nostra aplicació: <http://cursesservtest.appspot.com/cursa.jsp>

http://cursesservtest.appspot.com/cursa.jsp

Archivo Edición Ver Favoritos Herramientas Ayuda

Sitios sugeridos Conseguir más compl... Protección de los datos 48

Insertar Cursa

Nom:

Data:

Hora:

Comunitat:

Provincia:

Poblacio:

Distancia:

Desnivell:

Tipus:

Organitzador:

Accio:

7. Conclusions

A la finalització del projecte podem afirmar que s'han complert tots els objectius bàsics de l'aplicació que ens havíem marcat, si bé s'han hagut de descartar algunes funcionalitats pensades inicialment per la previsió de manca de temps.

El desenvolupament de tot el sistema ha estat una tasca força complexa principalment perquè s'ha realitzat des del desconeixement de la plataforma Android i s'ha hagut d'invertir molt temps en investigar i conèixer el sistema en general, i aprendre les peculiaritats del disseny de les interfícies d'usuari i dels recursos XML que probablement és l'aspecte més important en una APP per a mòbil vistosa, pràctica i usable. Per això s'ha decidit seguir les pràctiques d'última tendència en pantalles amb barra d'accions com a tret característic.

Un altre punt fort que s'ha de preveure, és la constant evolució de les versions Android. Curiosament en el transcurs del desenvolupament ja podíem descarregar una nova API per a programar amb nous complements i funcionalitats, que per una part motiva per poder innovar i ser més creatiu, però per una altra banda, hi ha un seguit de complicacions afegides per poder donar suport a dispositius amb versions antigues. En el nostre cas, ens hem trobat en un parell de casos on s'ha hagut de resoldre els conflictes de versions, tant per la barra d'acció com pel buscador.

També cal comentar que s'opta per la gestió de les dades de l'aplicació amb la plataforma Google App Engine, teòricament una plataforma "germana" d'Android però que presenta carències en el tema de la comunicació entre ambdues. Curiosament no hi ha un clar manual de bones pràctiques o una preferència clara per a connectar entre si, i no hi ha manera de fer-ho directament sobre les API del Datastore, per tant i inevitablement, sempre s'acaba anant a parar a una comunicació via HTTP de crides a serveis web, amb l'inconvenient dels forats de seguretat que presenta aquest canal i havent de generar per duplicat el model de dades i crear els *parsers* per vincular les dades al objecte corresponent, una feina força rudimentària al meu entendre. La avantatge principal és poder disposar d'un entorn servidor potent i fiable amb una sèrie de serveis d'una manera ràpida i econòmica.

A nivell personal considero que he posat en pràctica el conjunt de coneixements adquirits en assignatures relacionades amb les metodologies de programació orientada a objectes i el disseny de bases de dades principalment i a més ha estat molt satisfactori el descobriment i assimilació de nous coneixements com son les metodologies del disseny centrat en l'usuari, que al cap i a la fi és el destinatari final de l'aplicació i el gran beneficiat, així he intentat enfocar-ho.

Crec que ha estat una vivència molt engrescadora, i s'han anat superant tots els reptes tant tecnològics, com humans i creatius. Em quedo, per tant, amb un molt bon gust de boca i amb moltes ganes d'aprofundir en alguns aspectes interessants que no ha estat possible. Tot plegat, molt probablement farà d'aquesta experiència una bona base per a la futura construcció d'aplicacions a nivell professional.

Per tant puc concloure que aquest projecte té unes perspectives de millora i ampliació molt grans, i mostra d'això, és la llista de funcionalitats que es poden incorporar en futures versions:

- Crear calendari de curses per altres esports.
- Crear multi idioma.
- Estats d'una cursa.

- Publicar informació a xarxes socials.
- Crear diferents vistes del calendari.
- Notificacions de noves curses
- Usar la localització per filtrar curses d'una zona.
- Agenda de curses pròpia per usuari.
- Compartir informació amb comunitat d'amics.
- Ruta mostrada a GoogleMaps.
- Valoració de curses.
- Perfil desnivell d'una cursa.
- Comparativa de resultats.
- Inscripcions a curses.

8. Bibliografia

Play Store. Market oficial d'android-google.

Búsqueda d'aplicacions relacionades amb informació de curses a peu.
<<https://market.android.com/>> [Setembre 2012]

Android Developers. Pàgina oficial desenvolupadors d'android-google.

Manual de bones practiques. Tutorials. Exemples i API per Android
< <http://developer.android.com/index.html/> > [Setembre 2012]

Runedia. Portal web de curses estatals.

Referència d'aplicació web que proporciona informació de curses a peu, resultats i altres serveis.
< <http://www.runedia.com/> > [Setembre 2012]

Ajpdsoft. Portal de projectes TIC per a usuaris Internet open source i gratuït.

Revisió de treballs amb les tecnologies que usarem al projecte.
< <http://www.ajpdsoft.com/> > [Setembre 2012]

Enginyeria del programari. Documentació de l'assignatura per la UOC.
Benet Campderrich P00/05007/00299 [Setembre 2012]

Informe APEI sobre usabilidad. Document sobre usabilitat.
Yusef Hassan Montero, Sergio Ortega Santamaría
<<http://www.nosolousabilidad.com/manual/index.htm>> [Octubre 2012]

Disseny centrat en l'usuari. Documentació de l'assignatura per la UOC.
Muriel Garreta Domingo, Enric Mor Pera PID_00176048 [Octubre 2012]

Google App Developers. Portal del programador per Google App Engine.
Manuals, guies i exemples per programar a l'entorn GAE en Java. API Datastore
< <https://developers.google.com/appengine/docs/java/overview/> > [Novembre 2012]

Hello Android, 3rd Edition. Introducing Google's Mobile Development Platform.
Ed Brunette. Susannah Davidson [Novembre 2012]

Android Programming Tutorial v3.2. Android programming tutorials
Mark L. Murphy [Novembre 2012]

StackOverflow - FAQ programació Android
Portal amb preguntes més freqüents de dubtes i errors en la programació Android.
<<http://stackoverflow.com/questions/tagged/android>> [Novembre 2012]

GITHUB - Repositori de desenvolupament col·laboratiu per Android
<<http://mobile.github.com/>> [Novembre 2012]

9. Annex I

Disseny enquesta per l'anàlisi de requeriments en el DCU:

Enquesta

Disseny centrat en l'usuari.
L'objectiu és recopilar informació i obtenir el perfil d'usuari per a una aplicació mòbil Android.

*** Required**

Edat*

Sexe*

☐ M

☐ F

Realitza algun tipus d'esport?

☐ Marxa

☐ Correr

☐ Bicicleta

☐ Altres

☐ Cap

Realitza curses periòdicament?

☐ Si

☐ No

Busca habitualment informació de curses?

- ☐ Sí
- ☐ No

Té dificultat per trobar i/o gestionar informació relativa a curses?

- ☐ Sí
- ☐ No

Pot anomenar quina/es dificultats?

D'on treu la informació de les curses?

- ☐ Webs
- ☐ Revistes
- ☐ TV
- ☐ Boca a boca
- ☐ Altres mitjans

Quins interessos té amb relació a la informació sobre curses?

Disposa d'un smartphone o dispositiu mòbil semblant?

dispositiu amb connectivitat a Internet i capacitat d'usar aplicacions.

- ☐ Sí
- ☐ No

Quina plataforma és?

Sistema operatiu del mòbil

Android ▼

On usa el mòbil?

- ☐ Casa
- ☐ Feina
- ☐ Arreu

Ús del mòbil...

1 2 3 4 5

Esporàdic ☐ ☐ ☐ ☐ ☐ Freqüent

Descarrega i usa aplicacions per mòbil?

- ☐ Sí
- ☐ No

Usa alguna aplicació per mòbil relacionada amb l'esport?

- ☐ Sí
- ☐ No

Pot descriure-la breument?

Li sembla interessant una aplicació per mòbil relacionada amb informació de curses?

1
2
3
4
5

Gens
☐
☐
☐
☐
☐
Molt

Quines funcionalitats desitjaria incorporar a l'aplicació?
 Calendari de curses, informació de cursa detallada, afegir resultats de curses, compartir informació a la xarxa, etc.

Per contestar l'últim apartat, li recomanem que visiti la web www.runedia.com
 Per poder veure totes les funcionalitats al complet pot accedir amb el següent usuari:
 Correu: runedia@runedia.com i clau: runedia

Coneixia el portal de Runedia?
www.runedia.com

☐ Si
☐ No

Quines són per tu les funcionalitats més importants del portal?
www.runedia.com

Coneix alguna aplicació o portal web similar?
 Indiqui el nom

Li sembla interessant una aplicació per mòbil d'aquestes característiques?

1
2
3
4
5

Gens
☐
☐
☐
☐
☐
Molt