



FeelingLoadApp

Nom Estudiant: Cristina Hurtado Obiols

Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils

Nom Consultor/a: Francesc D'Assís Giralt Queralt

Professor/a responsable de l'assignatura: Carles Garrigues Olivella

Gener 2019



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>FeelingLoadApp</i>
Nom de l'autor:	<i>Cristina Hurtado Obiols</i>
Nom del consultor/a:	<i>Francesc D'Assís Giralt Queralt</i>
Nom del PRA:	<i>Carles Garrigues Olivella</i>
Data de lliurament (mm/aaaa):	<i>01/2019</i>
Titulació o programa:	<i>Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Bàsquet, RPE, Android, aplicació</i>
Resum del Treball (màxim 250 paraules):	
<p>En aquest projecte, s'ha desenvolupat una aplicació per a dispositius mòbils Android per a la gestió i registre del RPE ("Percepció Subjectiva de l'Esforç") en un equip esportiu, que permet controlar les càrregues d'entrenaments i evitar possibles lesions en els jugadors.</p> <p>La idea neix basada en la pròpia experiència, ja que moltes vegades aquesta recollida de dades s'ha de fer de forma manual i en les poques alternatives existents al mercat per al registre del RPE.</p> <p>Aquesta memòria recull tot el procés d'anàlisi, disseny, implementació i validació de l'aplicació, detallant tots els passos que s'han anat seguint per al desenvolupament.</p>	
Abstract (in English, 250 words or less):	
<p>In this project, we have developed an application for Android Mobile devices for the management and registration of the RPE (Rated Perceived Exertion) in a sports team, which allows the user to control team training loads and avoid possible injuries in the players.</p> <p>The idea is born based on one's own experience, since many times this collection of data must be done manually and in the few alternatives existint in the market</p>	

for the registration of the RPE.

This document collects the entire process of analysis, design, implementation and validation of the application, detailing all the steps followed for the development.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball	1
1.1.1 Importància del control de càrregues.....	1
1.1.2 Identificadors de càrrega interna.....	3
1.1.3 Estudi de mercat.....	5
1.2 Objectius del Treball.....	6
1.3 Enfocament i mètode seguit	7
1.4 Planificació del Treball	7
1.4.1 Possibles obstacles i solucions	8
1.4.2 Metodologia	9
1.4.3 Planificació temporal	9
1.5 Breu sumari de productes obtinguts.....	12
1.6 Breu descripció dels altres capítols de la memòria	12
2. Anàlisi.....	14
2.1 Identificació d'usuaris potencials	14
2.1.1 Fitxes de persones i escenaris	14
2.2 Requisits no funcionals.....	16
2.2.1. Requisits d'aparença	17
2.2.2. Requisits d'estil.....	18
2.2.3. Requisits de facilitat d'ús	18
2.2.4. Requisits de comprensió i cortesia.....	19
2.2.5. Requisits de llançament.....	20
2.2.6. Requisits de suport	20
2.2.7. Requisits de compliment.....	21

2.3 Casos d'ús	21
3. Disseny	27
3.1 Prototip de baixa fidelitat	27
3.2 Guia d'estil.....	28
3.3 Prototip d'alta fidelitat	28
3.4 Arbre de navegació.....	38
3.5 Patrons de disseny	38
3.5.1 Persistència de dades.....	39
4. Implementació	40
4.1 Eines emprades	40
4.2 Estructura del codi	41
4.3 Funcionalitats principals	42
4.3.1 Registre i login.....	42
4.3.2 Gestió de dades	44
4.3.3 Llistes.....	45
4.3.4 Mostrar imatges d'usuari	46
4.3.5 Pestanyes	47
4.3.6 Gràfics.....	48
4.4 Canvis de disseny.....	48
4.4.1 Login i registre	49
4.4.2 Pantalla principal d'entrenador	49
4.4.3 Pantalla de creació o edició de jugador	50
4.4.4 Pantalla de detalls d'entrenament.....	50
4.4.5 Pantalla principal de jugador	51
4.4.6 Registre de RPE.....	51
4.4.7 Menú lateral.....	52
4.5 Problemes sorgits durant el desenvolupament.....	52

5. Valoració de la implementació	53
5.1 Proves unitàries	53
5.2 Proves funcionals	54
5.2.1 Registre – CU1	54
5.2.2 Inici de sessió – CU2	56
5.2.3 Veure jugadors – CU3.....	58
5.2.4 Veure perfil de jugador – CU4.....	58
5.2.5 Afegir jugador – CU5.....	59
5.2.6 Veure entrenaments – CU6	61
5.2.7 Veure dades d’entrenament – CU7	61
5.2.8 Afegir entrenament – CU8	61
5.2.9 Veure estadístiques globals d’entrenament – CU9	63
5.2.10 Veure entrenaments – CU10.....	63
5.2.11 Registrar RPE en un entrenament – CU11	64
6. Conclusions	66
6.1 Assoliment d’objectius	66
6.2 Seguiment de la planificació	66
6.3 Aspectes a millorar	66
6.4 Treball futur	67
7. Glossari	68
8. Bibliografia	69
9. Annexos	70
9.1 Manual d’usuari	70

Llista de figures

Imatge 1: Relació entre càrrega física, rendiment i risc de lesions [1].....	2
Imatge 2: Tipus de càrregues físiques [2].....	3
Imatge 3: Escala de Borg per a la mesura del RPE [3]	4
Imatge 4: Funcionament de la metodologia àgil SCRUM [4]	9
Imatge 5: Desglossament de tasques	11
Imatge 6: Diagrama de Gantt corresponent a la PEC 1	11
Imatge 7: Diagrama de Gantt corresponent a la PEC 2	11
Imatge 8: Diagrama de Gantt corresponent a la PEC 3	11
Imatge 9: Diagrama de Gantt corresponent a l'entrega final	12
Imatge 10: Diagrama de casos d'ús.....	22
Imatge 11: Prototip de baixa fidelitat.....	27
Imatge 12: Guia d'estil de l'aplicació.....	28
Imatge 13: Pantalla inicial.....	29
Imatge 14: Pantalla de login.....	29
Imatge 15: Pantalla principal de l'entrenador.....	30
Imatge 16: Pantalla de nou jugador.....	31
Imatge 17: Llistat de jugadors al afegir nous elements	32
Imatge 18: Pantalla d'edició de jugador	33
Imatge 19: Pantalla per afegir nou entrenament.....	33
Imatge 20: Pantalla de dades d'entrenament	34
Imatge 21: Pantalla de dades globals d'entrenaments.....	35
Imatge 22: Pantalla principal del jugador	35
Imatge 23: Pantalla resum d'entrenament	36
Imatge 24: Pantalla de registre d'RPE.....	37
Imatge 25: Navegació de les diferents pantalles	37

Imatge 26: Arbre de navegació de l'aplicació	38
Imatge 27: Patró MVC [7]	39
Imatge 28: Comunicació client-servidor mitjançant Firebase.....	39
Imatge 29: Repositori GitHub de l'aplicació implementada	41
Imatge 30: Estructura del codi.....	41
Imatge 31: Estructura dels models utilitzats	42
Imatge 32: Integració de Firebase a Android Studio.....	43
Imatge 33: Usuaris registrats a l'aplicació de Firebase.....	43
Imatge 34: Registre d'usuaris a Firebase Database	44
Imatge 35: Pantalla de registre.....	49
Imatge 36: Pantalla principal d'entrenador amb la nova columna afegida.....	49
Imatge 37: Pantalla d'edició de jugador	50
Imatge 38: Pantalla de detalls d'entrenament amb la nova informació afegida.....	51
Imatge 39: Pantalla principal del jugador	51
Imatge 40: Pantalla de registre de l'RPE	52
Imatge 41: Menú lateral afegit	52
Imatge 42: Tests unitaris	53
Imatge 43: Execució dels tests unitaris	54
Imatge 44: Pantalla de registre i formulari amb dades introduïdes.....	55
Imatge 45: Registre del nou usuari a Firebase Auth	55
Imatge 46: Registre del nou usuari a Firebase Database	55
Imatge 47: Formulari de registre amb missatge d'error.....	56
Imatge 48: Missatge d'error a l'intentar registrar un usuari que ja existeix	56
Imatge 49: Pantalla d'inici de sessió, introducció de dades i pantalla principal de l'usuari.....	57
Imatge 50: Missatge d'error al no enviar totes les dades del formulari	57
Imatge 51: Missatge d'error al introduir dades errònies.....	57

Imatge 52: Llistat de jugadors d'un entrenador	58
Imatge 53: Pantalla de detalls de jugador	58
Imatge 54: Actualització de dades de jugador a Firebase Database	59
Imatge 55: Pantalla principal de jugador amb les dades modificades	59
Imatge 56: Formulari per afegir nou jugador	60
Imatge 57: Creació de nou jugador a Firebase Database	60
Imatge 58: Formulari per afegir jugador amb missatge d'error.....	60
Imatge 59: Llistat d'entrenaments d'un entrenador	61
Imatge 60: Pantalla amb les dades d'un entrenament	61
Imatge 61: Pantalla per afegir nou entrenament.....	62
Imatge 62: Pantalla per introduir nou entrenament amb missatge d'error	62
Imatge 63: Llistat d'entrenaments del jugador amb el nou entrenament afegit	63
Imatge 64: Pantalla de resum global d'estadístiques d'entrenaments.....	63
Imatge 65: Pantalla principal del jugador, amb el seu llistat d'entrenaments	64
Imatge 66: Formulari per registrar RPE per part d'un jugador	64
Imatge 67: Registre de RPE a la base de dades	65
Imatge 68: Pantalla de detalls d'entrenament després de registrar un RPE	65
Imatge 69: Pantalla de registre.....	70
Imatge 70: Pantalla de login.....	70
Imatge 71: Pantalla principal amb llistat de jugadors i formulari per afegir nou jugador.....	71
Imatge 72: Llistat d'entrenaments d'un entrenador	72
Imatge 73: Pantalla amb les dades d'un entrenament	72
Imatge 74: Pantalla per afegir nou entrenament.....	73
Imatge 75: Pantalla de resum global d'estadístiques d'entrenaments.....	73
Imatge 76: Pantalla principal del jugador, amb el seu llistat d'entrenaments	74
Imatge 77: Formulari per registrar RPE per part d'un jugador	74

Imatge 78: Menú lateral amb opció de tancament de sessió..... 75

1. Introducció

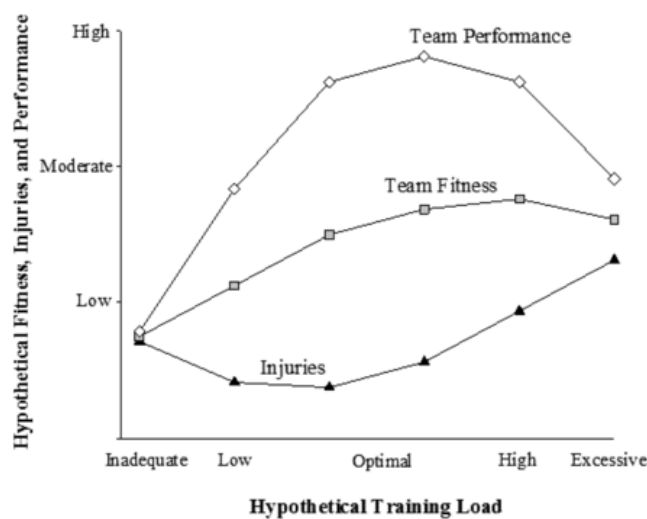
Aquest projecte correspon a un Treball de Fi de Màster (TFM) a la Universitat Oberta de Catalunya. En aquest document podem trobar tot el procés que s'ha seguit per a realitzar el projecte. Al principi podrem trobar tota la part corresponent a la gestió del projecte, que inclou els objectius, l'abast i la planificació temporal del projecte. A continuació, trobarem les principals decisions de disseny del sistema software. A partir d'aquí, podrem trobar els aspectes bàsics relacionats amb la implementació del sistema. Per últim, trobarem les valoracions i els resultats obtinguts a les proves de test, així com les conclusions finals del projecte.

1.1 Context i justificació del Treball

En qualsevol esport d'alt nivell, una bona planificació de la temporada i dels entrenaments és bàsica per a poder assolir els objectius plantejats a principi de temporada. En un nivell professional, l'obtenció de bons resultats és indispensable, i per això és necessari intentar controlar tots els aspectes que poden afectar a l'equip per tal d'intentar optimitzar tots els recursos disponibles.

1.1.1 Importància del control de càrregues

En una activitat on una condició física òptima és necessària, un aspecte que penalitza molt a l'equip són les lesions. Perdre jugadors durant un període de temps, pot penalitzar molt el rendiment general de l'equip, per tant, és molt important intentar minimitzar el risc de lesions controlant tots aquells aspectes que es poden monitoritzar i controlar per part dels entrenadors.





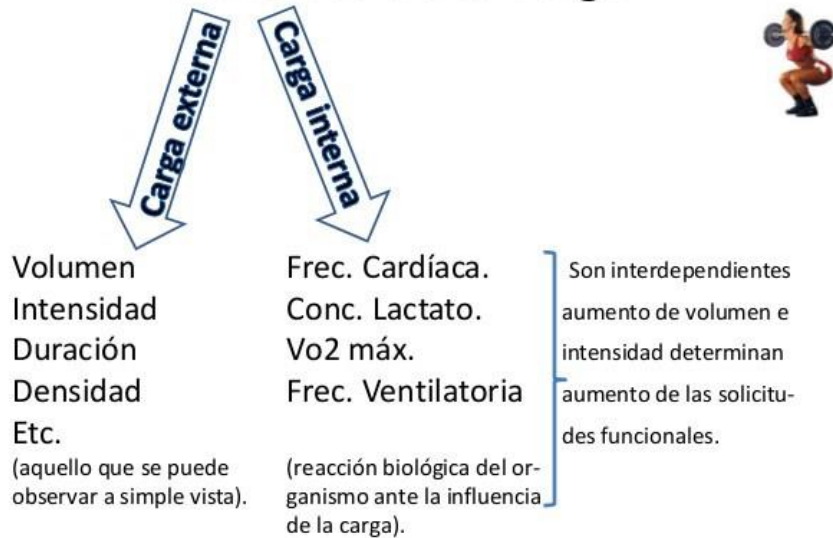
Imatge 1: Relació entre càrrega física, rendiment i risc de lesions [1]

Es per això que un aspecte a tenir molt en compte a la hora de planificar les sessions d'entrenament és la càrrega física que tenen els jugadors en cada moment, per tal de poder adaptar la intensitat de cada entrenament.

Podem dividir la càrrega física en dos tipus [2]:

- Càrrega externa: són tots aquells indicadors de l'entrenament realitzat, que es poden comptabilitzar de manera fàcil i objectiva. Les mesures que ens serveixen per a la càrrega externa són la duració, el volum, la intensitat, la densitat i el descans.
- Càrrega interna: podem definir la càrrega interna com l'efecte que té la càrrega externa sobre l'organisme. En aquest cas, la càrrega interna ens pot ajudar a saber com els jugadors reaccionen a cada tipus de càrrega externa, i saber si el nivell que s'aplica a l'entrenament és l'adequat. Per a poder quantificar la càrrega interna, podem utilitzar variables objectives (freqüència cardíaca, lactat...) i variables subjectives (RPE).

Índices de la carga



Imatge 2: Tipus de càrregues físiques [2]

Combinant els resultats d'ambdues càrregues i monitoritzant a tots els components de l'equip, podem veure ràpidament si els entrenaments planificats són adequats o si en canvi, estem aplicant un entrenament massa dur i en conseqüència, estem posant en risc de lesió els nostres jugadors. Tenir la capacitat de monitoritzar les càrregues físiques ens ajudarà a prendre decisions al moment i adaptar les diferents sessions d'entrenament i no sobre-carregar els jugadors.

La manera de monitoritzar la càrrega externa consistiria simplement en detallar els exercicis que componen cada entrenament, de tal manera que poguéssim saber, per a cada sessió, quina càrrega de temps i d'intensitat hi ha.

1.1.2 Identificadors de càrrega interna

Per a portar aquest control de càrregues internes, un dels mètodes més utilitzats és el que s'anomena "Percepció Subjectiva de l'Esforç" (RPE en anglès), que consisteix simplement en quantificar de manera subjectiva per part de l'esportista, la intensitat d'un entrenament segons una escala de l'1 al 10. Aquesta escala d'intensitats s'anomena escala de Borg [3].

Rating of Perceived Exertion (RPE Scale)	
10	Maximal
9	Really, Really, Hard
8	Really Hard
7	
6	Hard
5	Challenging
4	Moderate
3	Easy
2	Really Easy
1	Rest

Imatge 3: Escala de Borg per a la mesura del RPE [3]

Portant un control d'aquesta mesura després de cada sessió, es pot controlar el cansament de l'esportista, podent ajustar i dosificar les càrregues dels futurs entrenaments.

Aquest mètode és un dels més utilitzats actualment ja que ens permet conèixer la sensació dels jugadors, i es poden recollir les dades de manera fàcil, sense necessitat de cap aparell extern. El problema, però, és el mètode de recollida de dades. A partir de la pròpia experiència personal, s'ha observat que aquest recull de dades actualment es fa per part del preparador físic de l'equip de dues maneres diferents:

- Recollint totes les dades dels jugadors en un mateix full després de l'entrenament i entregant-lo al preparador físic, que haurà de passar totes aquestes dades al seu ordinador per a poder comparar-los amb altres sessions i analitzar els resultats. El problema d'aquest mètode és que les dades de cada esportista no són privades, i que les dades d'uns poden afectar en les decisions dels altres.
- Enviant un missatge privat a través del mòbil cada jugador al preparador físic, que haurà de recollir un a un els resultats, comprovant que no falti cap dada i passar totes les dades al seu ordinador.

Com podem veure, els dos mètodes requereixen una feina de recollida i tractament de dades per part del preparador físic. En aquest procés, primer de tot s'ha d'assegurar que tots els membres de l'equip emplenen la seva dada i posteriorment passar aquestes dades a un format que li permeti tenir-les totes

agrupades, de tal manera que durant aquest procés es poden produir errors a l'hora de copiar les dades d'un costat a altre.

S'ha de tenir en compte que el preparador físic moltes vegades no està present a tots els entrenaments, amb la qual cosa, no li es possible recollir les dades *in situ* al final de la sessió, i ha d'esperar a que tots els jugadors li enviïn aquesta dada per missatge.

A part d'això, per part dels jugadors, han de recordar després de cada entrenament que ha d'enviar el missatge al preparador físic. La mesura de l'esforç, per a que sigui el més fiable possible, ha d'indicar-se just al acabar l'entrenament, que és quan el jugador té més present la sensació de cansament.

Amb tot això, podem observar que el procés de recollida i processament de la percepció subjectiva d'esforç necessita d'un mètode ràpid i eficaç per a recollir les dades de tots els jugadors després de cada entrenament i per a poder visualitzar els resultats obtinguts.

D'aquesta necessitat neix la idea d'implementar una aplicació que permeti a un preparador físic recollir les dades dels seus jugadors després de cada entrenament. El preparador físic només s'hauria d'encarregar d'afegir els entrenaments per a que tots els seus jugadors poguessin visualitzar-los i introduir el seu valor d'esforç segons l'escala de Borg. Aquests resultats els podria visualitzar el preparador físic i analitzar-ne els resultats.

Aquesta idea podríem aplicar-la a tots els mètodes que serveixin per recollir dades sobre la càrrega interna, de tal manera que la nostra aplicació es podria anar enriquint amb més mètodes per fer-la més completa.

1.1.3 Estudi de mercat

Després d'investigar el mercat d'aplicacions, s'han trobat poques aplicacions que responen a aquesta necessitat:

- **RPE BorgScale PRO:** aplicació només per iOS i en anglès, que mesura el RPE per a un sol esportista. Preu: 3.99 lliures esterlines. [<https://itunes.apple.com/US/app/id1050933282?mt=8>]
- **FIT TESTS Qtests:** aplicació Android que mesura la fatiga a partir de 3 indicadors diferents: l'escala de Borg, el perfil d'estats d'ànim (POMS) i l'índex de Hooper. És gratuïta. [<https://apkpure.com/fit-tests-qtests/com.mac.lloyd.fittests.qtests>]
- **Quanter:** aplicació que permet recollir i visualitzar dades de RPE i altres qüestionaris de càrrega interna amb la seva versió més bàsica i gratuïta. El que no permet és detallar la càrrega externa, i si volem les opcions més

avançades, augmenta el preu.
[<https://play.google.com/store/apps/details?id=com.kvantia.quantar>]

- **Loadness Athlete/Loadness Trainer:** és la mateixa aplicació, però es necessita una o altre depenent de si qui la utilitza és entrenador o jugador. Permet recollir les dades de RPE, però el que fa es exportar les dades i enviar-les en format Excel a través d'un correu electrònic.
[<https://play.google.com/store/apps/details?id=com.loadness.athlete>]
[<https://play.google.com/store/apps/details?id=com.loadness.trainer>]

Tal com podem comprovar, actualment no existeixen gaires aplicacions relacionades amb el tema. La idea, a més, seria especialitzar una aplicació per a dissenyar-la específicament per a equips de bàsquet, ja que es podrien incloure opcions per al disseny d'entrenaments, com una pissarra electrònica.

Cal, doncs, desenvolupar una aplicació que permeti cobrir la necessitat anteriorment explicada i solucionant els problemes de les aplicacions ja existents.

1.2 Objectius del Treball

L'objectiu principal del projecte és la implementació d'una aplicació Android per a la gestió i monitorització de la càrrega física d'un equip professional de bàsquet.

L'equip tècnic tindrà la possibilitat de gestionar la seva plantilla i les seves sessions d'entrenament, així com visualitzar les dades corresponents a la càrrega física de cadascun dels components de l'equip.

Els jugadors/es, per la seva part, podran respondre els diferents qüestionaris sobre la càrrega física directament des del seu dispositiu mòbil, per tal que els membres del cos tècnic puguin rebre les dades automàticament.

Per tal de simplificar l'assoliment de l'objectiu final, el podem dividir en subobjectius més senzills:

- Identificar els usuaris mitjançant un sistema de login, i permetre distingir entre jugadors/es i entrenadors, ja que segons el perfil de cada usuari, tindrà disponible unes opcions o unes altres.
- Possibilitar als entrenadors gestionar la seva plantilla i afegir jugadors al seu equip.
- Permetre als entrenadors afegir entrenaments al calendari.
- Enviar automàticament els diferents qüestionaris sobre les càrregues físiques als jugadors al finalitzar cada entrenament, per tal que puguin completar-los.

- Possibilitat d'introduir informació sobre lesions passades per part dels entrenadors, o molèsties per part dels jugadors.
- Visualització de les dades rebudes per part dels entrenadors.

1.3 Enfocament i mètode seguit

Després de definir els objectius del projecte, és necessari decidir la estratègia que es seguirà per a dur a terme el seu desenvolupament.

Tal com hem vist anteriorment, no hi ha cap aplicació actualment que ofereixi exactament les funcionalitats necessàries per a complir els objectius plantejats. Per això mateix, es decideix desenvolupar una aplicació nova per a poder resoldre el problema plantejat.

Així doncs, com a solució final es planteja desenvolupar una aplicació multidispositiu. En aquest cas, degut al temps que hi ha per desenvolupar el projecte i als recursos disponibles, s'optarà per a l'aplicació mòbil per a dispositius Android, adaptant el disseny segons el tipus de dispositiu.

Per al desenvolupament del projecte, s'utilitzaran una sèrie de tecnologies i recursos hardware i software que es detallen a continuació:

Recursos hardware:

- Ordinador portàtil amb sistema operatiu Windows 10
- Telèfon mòbil Honor 8, amb versió Android 7.0
- Tablet Lenovo Tab 4, amb versió Android 7.0

Recursos software:

- Android Studio versió 3.2
- Microsoft Word, per a la realització de la memòria i el manual d'usuari
- Tom's Planner, per a la realització de la planificació
- Microsoft Visio, per a la realització de diagrames

1.4 Planificació del Treball

Una bona planificació serà la clau per a desenvolupar el projecte, ja que ens ajudarà a repartir les tasques en el temps i els recursos que tenim. En primer lloc, haurem de fer una feina per a identificar els possibles obstacles que ens podem anar trobant a mesura que anem desenvolupant el projecte. Identificar aquests obstacles, ens pot permetre pensar per avançat les possibles solucions per a afrontar-los, per tal que, si apareixen, no afectin a la nostra planificació inicial.

Una altra part important de la planificació serà escollir la millor metodologia possible per al projecte, identificant la que millor s'adapta a les condicions i necessitats del mateix.

Per últim, caldrà identificar les tasques en les que es dividirà cadascuna de les fases del projecte, i assignar a cadascuna els recursos i temps necessaris per a complir les diferents entregues.

1.4.1 Possibles obstacles i solucions

Com en tot projecte, durant la realització d'aquest poden sorgir diferents obstacles que dificultin la seva realització. Per això, és important poder detectar quins són aquests obstacles i pensar en la manera de superar-los per tal de que si ens els trobem durant el projecte, siguem capaços de superar-los sense que ens afectin excessivament a la planificació inicial.

Temps per desenvolupar el projecte

El temps per desenvolupar el projecte és curt, i és una de les parts crítiques del projecte, ja que no es podrà ampliar. Per tal de superar aquesta situació, és necessari fixar uns objectius realistes i una planificació acurada per a poder complir-los sense problemes. A part d'això, revisions freqüents amb el tutor per tal de comprovar que els objectius es compleixen i que la planificació es realista i es compleix.

Mala planificació

Relacionat amb el punt anterior, degut al poc temps del que es disposa per a realitzar el projecte, és necessària una planificació realista i acurada, per tal d'assegurar que es compliran tots els objectius. Per comprovar que es segueix la planificació, es realitzaran revisions setmanals i es farà una re planificació si és necessari.

Errors de codificació

Un altre dels problemes crítics amb el que ens podem trobar són els errors en el codi. Aquests errors poden ser a vegades difícils de detectar i, en canvi, poden comportar molts problemes i retards en la planificació, ja que pot ser que triguem molt de temps en detectar l'error. Per tal de minimitzar aquest risc, caldrà anar testejant el software generat contínuament, per comprovar que el nou codi que es va generant funciona correctament.

Relacionat també amb aquest punt, és important centrar la implementació en els punts més importants per a poder aconseguir les funcionalitats principals que ha de tenir l'aplicació i a partir d'aquí, anar ampliant-la amb tots els detalls possibles.

Pèrdua de dades

Aquest risc és un dels que més poden afectar al desenvolupament del projecte. Una pèrdua de dades (ja sigui de codi o de documentació), seria fatal per al projecte, i produiria una desviació molt important, ja que s'hauria de tornar a començar. Per tal d'evitar aquestes pèrdues de dades, és necessari fer diverses còpies de seguretat i guardar-les a diferents llocs per tal de poder recuperar la informació si és necessari. Per a evitar aquests problemes, s'utilitzarà un repositori de codi al núvol, per tal de poder tenir assegurat el codi que anem generant. En quant a la documentació, també es disposarà d'una còpia de seguretat al núvol.

1.4.2 Metodologia

La metodologia que s'utilitzarà en aquest projecte, serà una metodologia àgil. Aquest tipus de metodologia es basa en una sèrie d'iteracions curtes, cadascuna de les quals produeix software funcional. D'aquesta manera, podrem afrontar alguns dels riscos descrits anteriorment.

En primer lloc, podrem anar produint software després de cada iteració. Aquest software serà 100% funcional, sense errors i testejat. Això ens ajudarà a evitar el risc dels errors de codi, ja que amb aquesta metodologia, no arrastrarem errors de codificació durant la realització del projecte.

Les iteracions curtes de la metodologia àgil ens permetran tenir més controlat en quin estat i en quin punt de la planificació es troba el projecte, a part de que podrem detectar abans les desviacions en la planificació i corregir-les abans.



Imatge 4: Funcionament de la metodologia àgil SCRUM [4]

1.4.3 Planificació temporal

El temps de duració del projecte és d'aproximadament 4 mesos. Això suposa poc temps per a treballar, per la qual cosa una bona planificació és bàsica per a l'èxit del projecte.

A més, encara que es presentarà una planificació inicial de tot el projecte, al fer servir una metodologia àgil, amb revisions constants, aquesta planificació pot

anar-se modificant segons els terminis que es vagin complint i les desviacions que vagin apareixent.

El primer que s'ha fet ha estat establir la càrrega total d'hores del projecte, segons els crèdits que el formen. Si tenim en compte que el projecte té un total de 12 crèdits i si estimem que les hores de dedicació personal per cada crèdit és d'unes 25 hores, tenim que el total d'hores de dedicació al projecte seran 300h.

Des de la data d'inici del projecte (19 de setembre del 2018) fins a la data de finalització (2 de gener del 2019), tenim un total de 106 dies, repartits en 69 dies laborables i 37 dies festius o caps de setmana.

Degut a una jornada completa de feina, els dies laborables només es podran dedicar al projecte un total de 2h per dia. Això fa que sigui necessari aprofitar molt bé els dies festius i caps de setmana per a poder complir les fites i seguir la planificació segons les hores estimades.

Per tant, el total d'hores dedicades serà el resultat de la següent operació:

$$(69 \text{ dies laborables} \times 2\text{h al dia}) + (37 \text{ dies festius} \times 4\text{h al dia}) = 286\text{h}$$

Arribant a complir aproximadament el total d'hores de dedicació necessàries.

Per a poder visualitzar la planificació temporal de tot el projecte, s'ha realitzat un diagrama de Gantt, en el que s'ha dividit el projecte en 4 blocs principals, segons les entregues de les diferents PECs. Cadascun d'aquests blocs, a la vegada, està dividit en tasques més específiques.

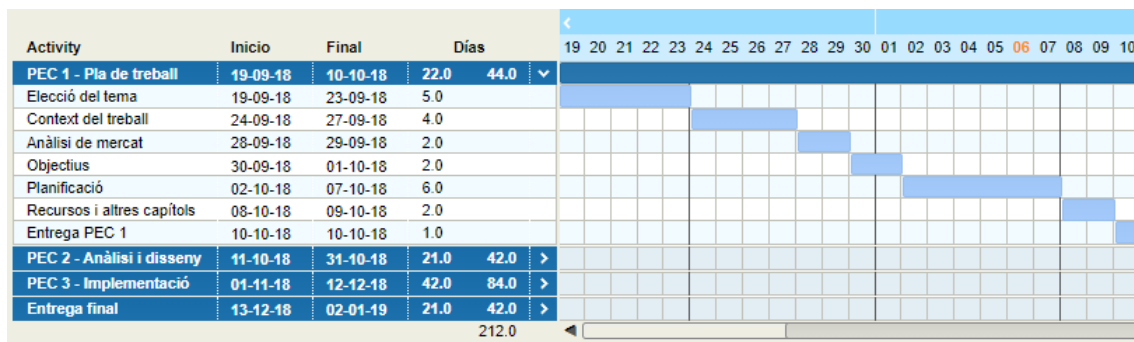
A continuació, es mostra el llistat de tasques amb les seves dates d'inici i de fi, i les hores planificades per a cadascuna:

Tasca	Inici	Fi	Dies	Hores
PEC 1 - Pla de treball	19/09/2018	10/10/2018	22	58
Elecció del tema	19/09/2018	23/09/2018	5	14
Context del treball	24/09/2018	27/09/2018	4	10
Anàlisi de mercat	28/09/2018	29/09/2018	2	6
Objectius	30/09/2018	01/10/2018	2	6
Planificació	02/10/2018	07/10/2018	6	16
Recursos i altres capítols	08/10/2018	09/10/2018	2	4
Entrega PEC 1	10/10/2018	10/10/2018	1	2
PEC 2 - Anàlisi i disseny			21	56
Requisits	11/10/2018	14/10/2018	4	14
Casos d'ús	15/10/2018	20/10/2018	6	14
Disseny d'interfícies	21/10/2018	26/10/2018	6	14
Disseny d'arquitectura	27/10/2018	30/10/2018	4	12
Entrega PEC 2	31/10/2018	31/10/2018	1	2

PEC 3 - Implementació				
			42	112
Iteració 1	01/11/2018	12/11/2018	12	34
Iteració 2	13/11/2018	24/11/2018	12	30
Iteració 3	25/11/2018	06/12/2018	12	32
Proves	07/12/2018	11/12/2018	5	14
Entrega PEC 3	12/12/2018	12/12/2018	1	2
Entrega final				
			21	60
Manual d'usuari	13/12/2018	20/12/2018	8	20
Vídeo	21/12/2018	30/12/2018	10	32
Revisió final	31/12/2018	01/01/2019	2	6
Entrega final	02/01/2019	02/01/2019	1	2

Imatge 5: Desglossament de tasques

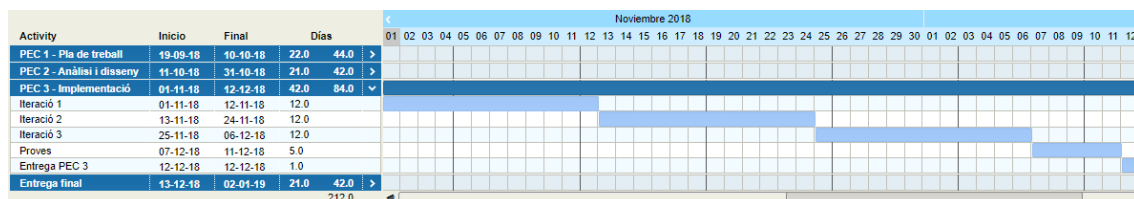
I per últim, es mostren els diagrames de Gantt corresponents a cadascun dels blocs principals:



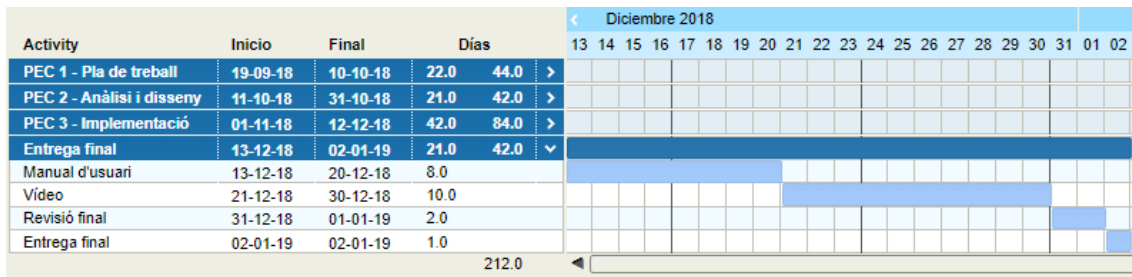
Imatge 6: Diagrama de Gantt corresponent a la PEC 1



Imatge 7: Diagrama de Gantt corresponent a la PEC 2



Imatge 8: Diagrama de Gantt corresponent a la PEC 3



Imatge 9: Diagrama de Gantt corresponent a l'entrega final

1.5 Breu sumari de productes obtinguts

El següent llistat mostra tots els productes obtinguts al finalitzar el projecte, que s'aniran detallant en els pròxims capítols:

- Aplicació Android en format APK
- Codi font de l'aplicació
- Memòria del projecte
- Manual d'usuari de l'aplicació
- Vídeo amb la presentació del projecte

1.6 Breu descripció dels altres capítols de la memòria

A continuació, es detallen la resta de capítols que contindrà la memòria del projecte:

1. **Anàlisi:** en aquest capítol, s'identificaran els diferents usuaris potencials de l'aplicació, així com els diferents escenaris de cadascú. També es definiran els casos d'ús i els requisits, funcionals i no funcionals, que haurà de complir l'aplicació.
2. **Disseny:** a partir de l'anàlisi realitzat al capítol anterior, es procedirà a un disseny d'un prototip de l'aplicació en alt nivell, on es podran veure les diferents pantalles que la formaran. També s'inclourà la navegació entre les diferents pantalles i es realitzarà el disseny de l'arquitectura necessària per a la implementació de l'aplicació.
3. **Implementació:** en aquest punt del projecte, partint de la feina dels capítols anteriors, es procedirà a la implementació de l'aplicació. En aquest capítol, s'explicaran les diferents eines utilitzades per a la implementació de l'aplicació, així com l'estructuració i les solucions utilitzades per a implementar les principals funcionalitats.
4. **Valoració de la implementació:** la fase posterior a la implementació, consistirà en una valoració completa de l'aplicació, realitzant una sèrie de

proves que ens ajudaran a descobrir possibles errors i corregir-los per a poder obtenir el producte final.

- 5. Conclusions:** per últim, es redactaran les conclusions del projecte i es procedirà a revisar la memòria i tots els productes finals del projecte per tal d'eliminar possibles errors.

2. Anàlisi

En aquest apartat, es procedirà a l'anàlisi de l'aplicació a desenvolupar. En primer lloc, s'identificaran els usuaris potencials, així com els diferents escenaris d'ús de cadascú respecte a l'aplicació. A continuació es definiran una sèrie de requisits que l'aplicació haurà de complir, i per últim, es definiran els diferents casos d'ús.

2.1 Identificació d'usuaris potencials

A continuació, es farà una descripció de les possibles parts implicades que podríem trobar en relació al projecte que s'està desenvolupant. Aquestes parts implicades fan referència a totes aquelles persones o organitzacions que puguin tenir algun tipus d'interès en el projecte, o que es puguin veure afectades per aquest.

Després de definir la problemàtica i la solució a desenvolupar, podem observar que tindrem dos tipus d'usuaris diferents: el preparador físic i els jugadors. El preparador físic haurà de ser l'encarregat de planejar els entrenaments de l'equip, de tal manera que els jugadors puguin saber quan han de registrar la sensació de cansament a l'aplicació. A la vegada, serà qui rebi totes les dades introduïdes pels jugadors i les podrà visualitzar de manera ràpida i senzilla.

Per part dels jugadors, l'aplicació els hi oferirà la possibilitat d'introduir l' RPE fàcilment a l'aplicació per a que sigui enviada al preparador físic.

No serà una aplicació utilitzada per un públic molt general, ja que aquesta respon a una necessitat molt específica i per un àmbit molt concret. L'usuari que utilitzi l'aplicació haurà de tenir els coneixements necessaris sobre preparació física necessaris per a poder interpretar els resultats obtinguts.

2.1.1 Fitxes de persones i escenaris

A continuació, es presenten dues fitxes de persona i escenari, reproduint els dos perfils d'usuari anteriorment descrits.

Nom: Victor	
Edat: 29 anys	
Professió: Preparador físic	
Descripció de la persona:	

El Victor té 29 anys, viu en parella però encara no té fills. Viu a Barcelona en un pis de lloguer i diàriament es desplaça a Sant Adrià en metro per les tardes per portar la preparació física del primer equip femení de bàsquet de la ciutat. Prepara minuciosament l'entrenament físic segons el partit de la setmana i el cansament de les jugadores.

A casa disposa d'ordinador amb fibra òptica per a connectar-se a internet, el qual utilitza diàriament per a realitzar feina del màster que està estudiant. També l'utilitza per veure totes les dades que recopila sobre els entrenaments de l'equip de bàsquet i preparar els continguts de cada sessió física.

També disposa d'un *smartphone* amb connexió a internet, el qual utilitza habitualment per a connectar-se a les seves xarxes socials i per utilitzar les seves aplicacions.

Després de cada entrenament i partit, ha de recollir la sensació d'esforç de cada jugadora, per poder analitzar les dades recollides i ajustar la planificació física en funció d'això. Necessita tenir les dades el més ràpid possible i visualitzar-les fàcilment, per si ha d'ajustar el següent entrenament.

Descripció de l'escenari:

És dimarts a la nit, són les 23 i el Victor està emocionat perquè aquesta setmana les seves jugadores visiten el camp del líder. Vol que totes arribin al partit de dissabte en les millors condicions físiques, per això ha preparat els entrenaments de tota la setmana de manera molt exhaustiva i ja ha planificat tots els entrenaments de la setmana a l'aplicació.

El Victor està esperant a que les noies acabin l'entrenament per poder veure la seva sensació de cansament i ajustar la sessió del dia següent.

Quan acaba de recollir la taula del sopar, el Victor entra a l'aplicació i veu que totes les jugadores ja han introduït les dades de l'entrenament. Aleshores visualitza quins han estat els resultats globals i veu que no cal ajustar res del següent entrenament, ja que el cansament és acord al que havia planificat.

El Victor marxa a dormir content per la bona feina que han fet les seves jugadores i desitjant que arribi dissabte per guanyar al líder.

Nom: Maria	
Edat: 21 anys	
Professió: Estudiant de dret	
Descripció de la persona:	
<p>La Maria té 21 anys i viu a Barcelona amb els seus pares. Pels matins estudia dret a la universitat Pompeu Fabra i per les tardes juga a bàsquet al primer equip de Sant Adrià. Necessita planificar-se molt bé la feina de la universitat perquè acaba molt tard d'entrenar i al dia següent s'aixeca molt aviat pel matí.</p> <p>A casa disposa d'un ordinador de sobretaula amb connexió ADSL que comparteix amb tota la família. Aquest ordinador l'utilitzen més aviat els seus pares, ella utilitza el seu portàtil per fer la feina de la universitat. A part d'això, té un <i>iPhone</i> amb connexió a internet, que utilitza per entrar a les seves xarxes socials i parlar amb els seus amics.</p> <p>Després de cada entrenament i mentre torna a casa amb la seva mare, encara que està molt cansada de tot el dia, ha d'enviar la sensació al seu preparador físic abans d'anar a dormir. Vol fer-ho el més ràpid possible i que no se li obliidi, per tal de poder començar a llegir tots els missatges que té al seu mòbil.</p>	
Descripció de l'escenari:	
<p>És dimarts a la nit, són les 23:30 i la Maria ha acabat d'entrenar. Ha sigut un bon entrenament per part de tot l'equip i l'entrenador està content. Fa uns quants estiraments amb les seves companyes abans d'agafar les coses per anar a la dutxa.</p> <p>Després de la dutxa, es troba amb la seva mare per tornar cap a casa. Agafa la bossa i treu el seu mòbil per mirar quins missatges té. Quan obre el mòbil, veu el recordatori de que ha d'introduir la sensació de cansament per enviar-li al preparador físic. Ho fa a l'instant, de tal manera que ja pot continuar mirant els missatges que té per llegir i entrar a les seves xarxes socials per mirar que han publicat els seus amics.</p>	

2.2 Requisits no funcionals

Els requisits no funcionals o requisits de qualitat són tots aquells requisits que el sistema ha de complir però que no es poden expressar com a forma de casos d'us.

Per mostrar tots aquest requisits utilitzarem la classificació que fa Volere [5] i els mostrarem segons la seva plantilla. Els continguts de la plantilla es troben explicats a continuació:

Requisit #	Identificador del requisit		
Descripció	Descripció del requisit		
Justificació	Justificació de la necessitat del requisit		
Autor	Part interessada per a que el requisit existeixi en el sistema		
Criteri de satisfacció	Condicció que s'haurà de satisfer per a que el requisit es doni com a complert		
Satisfacció del client	Satisfacció del client si es compleix el requisit (de 1 a 5)	Insatisfacció del client	Insatisfacció del client si no es compleix el requisit (de 1 a 5)
Prioritat	Prioritat per a complir el requisit (Baixa - Mitjana - Alta)	Conflictes	Possibles conflictes amb altres requisits

2.2.1. Requisits d'aparença

Requisit #	1		
Descripció	El disseny és atractiu per a l'usuari		
Justificació	Un disseny atractiu fa que els usuaris estiguin més oberts a utilitzar l'aplicació.		
Autor	Propietari		
Criteri de satisfacció	Es farà una prova amb un conjunt de persones i almenys el 90% considerarà que el sistema és atractiu i convida a utilitzar-lo.		
Satisfacció del client	2	Insatisfacció del client	3
Prioritat	Alta	Conflictes	-

2.2.2. Requisits d'estil

Requisit #	2		
Descripció	El disseny de l'aplicació transmet confiança a l'usuari		
Justificació	Un disseny que transmeti confiança atraurà més als usuaris i els convidarà a utilitzar-la.		
Autor	Propietari		
Criteri de satisfacció	Es farà una prova amb un conjunt de persones i almenys el 90% considerarà que el sistema els hi transmet confiança.		
Satisfacció del client	2	Insatisfacció del client	3
Prioritat	Alta	Conflictes	-

Requisit #	3		
Descripció	El disseny de l'aplicació és simple		
Justificació	Un disseny simple permetrà que qualsevol usuari pugui utilitzar-la sense problemes.		
Autor	Propietari		
Criteri de satisfacció	Es farà una prova amb un conjunt de persones i almenys el 90% considerarà que el disseny és simple.		
Satisfacció del client	3	Insatisfacció del client	5
Prioritat	Alta	Conflictes	-

2.2.3. Requisits de facilitat d'ús

Requisit #	4		
Descripció	El funcionament de l'aplicació és fàcil d'entendre		

Justificació	El sistema ha de ser fàcil d'utilitzar per una persona que no hi està familiaritzada.		
Autor	Usuari		
Criteri de satisfacció	Es farà una prova amb un conjunt de persones i almenys el 90% considerarà que el sistema és fàcil d'utilitzar.		
Satisfacció del client	2	Insatisfacció del client	5
Prioritat	Alta	Conflictes	-

2.2.4. Requisits de comprensió i cortesia

Requisit #	5		
Descripció	L'aplicació utilitzarà un llenguatge estàndard, comprensible per a qualsevol usuari		
Justificació	Necessitem utilitzar un llenguatge fàcilment comprensible i que no pugui donar lloc a malentesos o dubtes per part de l'usuari		
Autor	Usuari		
Criteri de satisfacció	Es farà una prova amb un conjunt de persones i almenys el 90% considerarà que el llenguatge utilitzat al sistema és comprensible.		
Satisfacció del client	3	Insatisfacció del client	4
Prioritat	Alta	Conflictes	-

Requisit #	6		
Descripció	La interfície del software ha de fer servir les icones i widgets típics		
Justificació	No fer-ho d'aquesta manera podria portar a confondre l'usuari i portar-lo a equivocar-se		

Autor	Usuari		
Criteri de satisfacció	S'utilitzaran els widgets utilitzats en aplicacions semblants		
Satisfacció del client	1	Insatisfacció del client	4
Prioritat	Mitja	Conflictes	-

2.2.5. Requisits de llançament

Requisit #	7		
Descripció	L'aplicació ha de ser fàcil d'actualitzar a versions més avançades i lliure d'errors		
Justificació	No pot haver-hi errors en l'aplicació permanentment, sinó que aquesta ha de poder actualitzar-se fàcilment		
Autor	Usuari		
Criteri de satisfacció	Es dissenyarà l'aplicació des d' un principi perquè es pugui actualitzar i canviar fàcilment		
Satisfacció del client	2	Insatisfacció del client	4
Prioritat	Alta	Conflictes	-

2.2.6. Requisits de suport

Requisit #	8		
Descripció	L'aplicació disposa d'un manual d'usuari senzill		
Justificació	L'usuari ha de disposar d'una guia que li permeti aprendre el funcionament del sistema de manera fàcil i ràpida		
Autor	Propietari		
Criteri de	Es farà una prova amb un conjunt de persones i almenys el 90%		

satisfacció	considerarà que el manual d'usuari és fàcil d'entendre.		
Satisfacció del client	1	Insatisfacció del client	4
Prioritat	Mitja	Conflictes	-

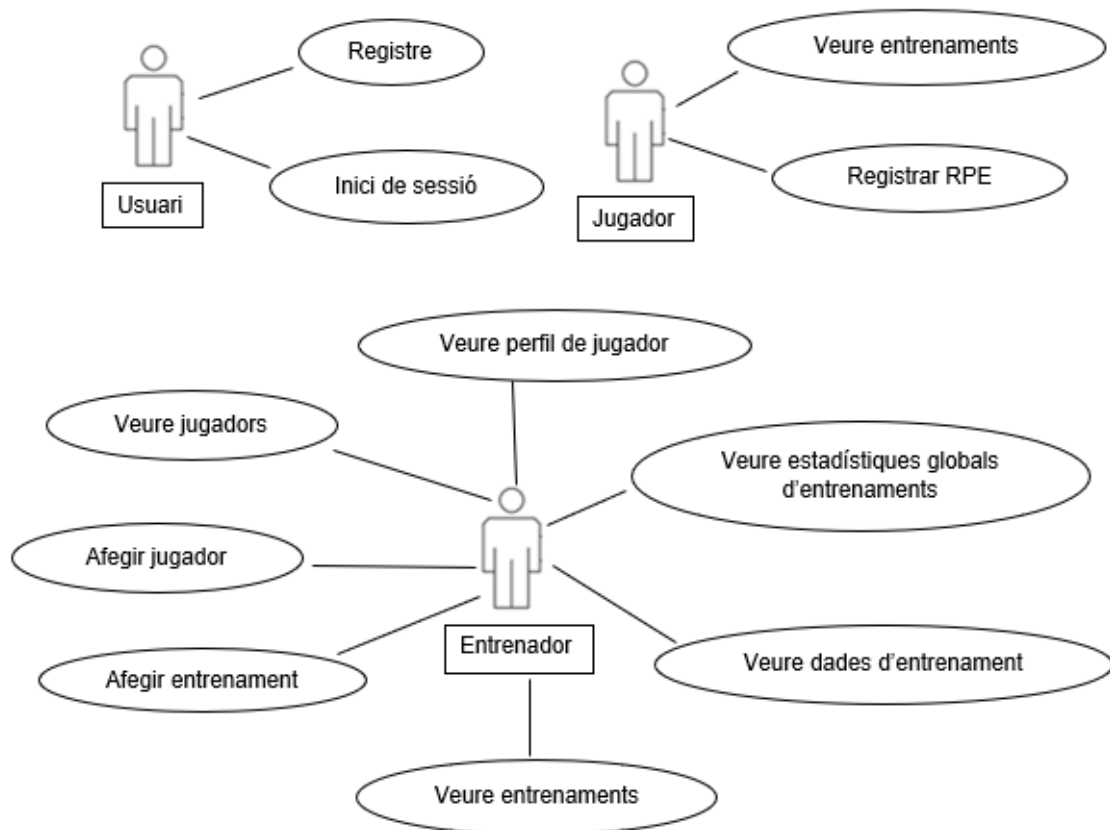
2.2.7. Requisits de compliment

Requisit #	9		
Descripció	El tractament de les dades utilitzades dels usuaris ha de complir la Llei Orgànica de Protecció de Dades (LOPD)		
Justificació	El sistema ha de complir les lleis del país i funcionar de forma legal		
Autor	Propietari		
Criteri de satisfacció	Un advocat contractat confirmarà el compliment de les lleis que pertocuen		
Satisfacció del client	1	Insatisfacció del client	5
Prioritat	Alta	Conflictes	-

2.3 Casos d'ús

Els casos d'ús ens ajudaran a definir els requisits funcionals que ha de complir la nostra aplicació per tal d'assolir els objectius plantejats. Aquests casos d'ús definiran què ha de fer el sistema i com interactuarà amb l'usuari, però no com ho ha de fer.

A continuació, es mostra el diagrama de casos d'ús que recull aquelles funcionalitats que els diferents usuaris potencials, identificats anteriorment, poden realitzar amb el sistema.



Imatge 10: Diagrama de casos d'ús

Una vegada definits els casos d'ús, cal especificar-los. Per a definir els casos d'ús s'utilitzarà la plantilla explicada a continuació.

Identificador: CU-X

Cas d'ús: Nom del cas d'ús

Actor primari: Actor principal del cas d'ús

Precondició: Condició que s'ha de complir per a poder executar el cas d'ús

Disparador: Acció que inicia el cas d'ús

Escenari principal: Interaccions entre l'actor principal i el sistema, que defineixen els passos a seguir per tal de completar amb èxit el cas d'ús

Escenari alternatiu: Possibles camins alternatius que es poden seguir diferents a l'escenari principal del cas d'ús

Prioritat: Baixa-Mitja-Alta

Identificador: CU-1

Cas d'ús: Registre

Actor primari: Usuari

Precondició: -

Disparador: L'usuari es vol registrar a l'aplicació

Escenari principal:

1. L'usuari vol registrar-se a l'aplicació
2. El sistema mostra a l'usuari les dades requerides per al registre
3. L'usuari omple la informació del formulari i envia les dades al sistema
4. El sistema registre l'usuari en una base de dades

Escenari alternatiu:

1. Si les dades introduïdes per l'usuari no són vàlides o l'usuari ja existeix al sistema, no es registraria l'usuari i es tornaria al pas 2

Prioritat: Alta

Identificador: CU-2

Cas d'ús: Inici de sessió

Actor primari: Usuari

Precondició: L'usuari ja està registrat al sistema

Disparador: L'usuari vol iniciar sessió a l'aplicació

Escenari principal:

1. L'usuari vol iniciar sessió a l'aplicació
2. El sistema mostra les dades necessàries per iniciar sessió
3. L'usuari omple les dades i ho indica al sistema
4. El sistema inicia la sessió de l'usuari i mostra la seva pantalla principal

Escenari alternatiu:

1. Les dades introduïdes per l'usuari no són vàlides o l'usuari no està registrat al sistema. No s'iniciaria sessió i es tornaria al pas 2

Prioritat: Alta

Identificador: CU-3

Cas d'ús: Veure jugadors

Actor primari: Entrenador

Precondició: L'entrenador ha iniciat sessió a l'aplicació

Disparador: L'usuari vol veure un dels jugadors del seu equip

Escenari principal:

1. L'usuari indica que vol veure els seus jugadors
2. El sistema mostra un llistat dels jugadors de l'entrenador

Escenari alternatiu: -

Prioritat: Alta

Identificador: CU-4

Cas d'ús: Veure perfil de jugador

Actor primari: Entrenador

Precondició: L'entrenador ha iniciat sessió a l'aplicació i ha llistat els seus jugadors

Disparador: L'usuari vol veure el perfil d'un dels jugadors del seu equip

Escenari principal:

1. L'usuari indica el jugador del qual vol veure el seu perfil
2. El sistema mostra les dades del jugador seleccionat

Escenari alternatiu: -

Prioritat: Mitja

Identificador: CU-5

Cas d'ús: Afegir jugador

Actor primari: Entrenador

Precondició: L'entrenador ha iniciat sessió a l'aplicació

Disparador: L'usuari vol afegir un jugador al seu equip

Escenari principal:

1. L'usuari indica que vol afegir un jugador al seu equip
2. El sistema mostra les dades necessàries per afegir al jugador
3. L'usuari omple les dades necessàries i ho indica al sistema
4. El sistema valida la informació i afegeix el nou jugador

Escenari alternatiu:

1. Les dades introduïdes no són vàlides, el sistema ho indica i es torna al pas 2.

Prioritat: Alta

Identificador: CU-6

Cas d'ús: Veure entrenaments

Actor primari: Entrenador

Precondició: L'entrenador ha iniciat sessió a l'aplicació

Disparador: L'usuari vol veure els entrenaments del seu equip

Escenari principal:

1. L'usuari indica que vol veure els seus entrenaments
2. El sistema mostra un llistat dels entrenaments enregistrats

Escenari alternatiu: -

Prioritat: Alta

Identificador: CU-7

Cas d'ús: Veure dades d'entrenament

Actor primari: Entrenador

Precondició: L'entrenador ha iniciat sessió a l'aplicació i ha llistat els seus entrenaments

Disparador: L'usuari vol veure les dades d'un dels seus entrenaments

Escenari principal:

1. L'usuari indica l'entrenament del qual vol veure les seves dades
2. El sistema mostra les dades de l'entrenament seleccionat

Escenari alternatiu: -

Prioritat: Mitja

Identificador: CU-8

Cas d'ús: Afegir entrenament

Actor primari: Entrenador

Precondició: L'entrenador ha iniciat sessió a l'aplicació

Disparador: L'usuari vol afegir un entrenament al seu equip

Escenari principal:

1. L'usuari indica que vol afegir un entrenament al seu equip
2. El sistema mostra les dades necessàries per afegir l'entrenament
3. L'usuari omple les dades necessàries i ho indica al sistema
4. El sistema valida la informació i afegeix el nou entrenament

Escenari alternatiu:

1. Les dades introduïdes no són vàlides, el sistema ho indica i es torna al pas 2.

Prioritat: Alta

Identificador: CU-9

Cas d'ús: Veure estadístiques globals d'entrenaments

Actor primari: Entrenador

Precondició: L'entrenador ha iniciat sessió a l'aplicació

Disparador: L'usuari vol veure les estadístiques globals dels entrenaments

Escenari principal:

1. L'usuari indica que vol veure les dades globals dels seus entrenaments
2. El sistema mostra les dades globals que tingui enregistrades

Escenari alternatiu: -

Prioritat: Alta

Identificador: CU-10

Cas d'ús: Veure entrenaments

Actor primari: Jugador

Precondició: El jugador ha iniciat sessió a l'aplicació

Disparador: L'usuari vol veure els seus entrenaments

Escenari principal:

1. L'usuari indica que vol veure els seus entrenaments
2. El sistema mostra un llistat dels entrenaments que estan associats a ell

Escenari alternatiu: -

Prioritat: Alta

Identificador: CU-11

Cas d'ús: Registrar RPE en un entrenament

Actor primari: Jugador

Precondició: El jugador ha iniciat sessió a l'aplicació

Disparador: L'usuari vol enregistrar l'RPE en un dels seus entrenaments

Escenari principal:

1. L'usuari indica l'entrenament en el qual vol enregistrar l'RPE
2. El sistema mostra les dades necessàries per a enregistrar l'RPE
3. L'usuari omple les dades i ho indica al sistema
4. El sistema registra l'RPE del jugador

Escenari alternatiu: -

Prioritat: Alta

3. Disseny

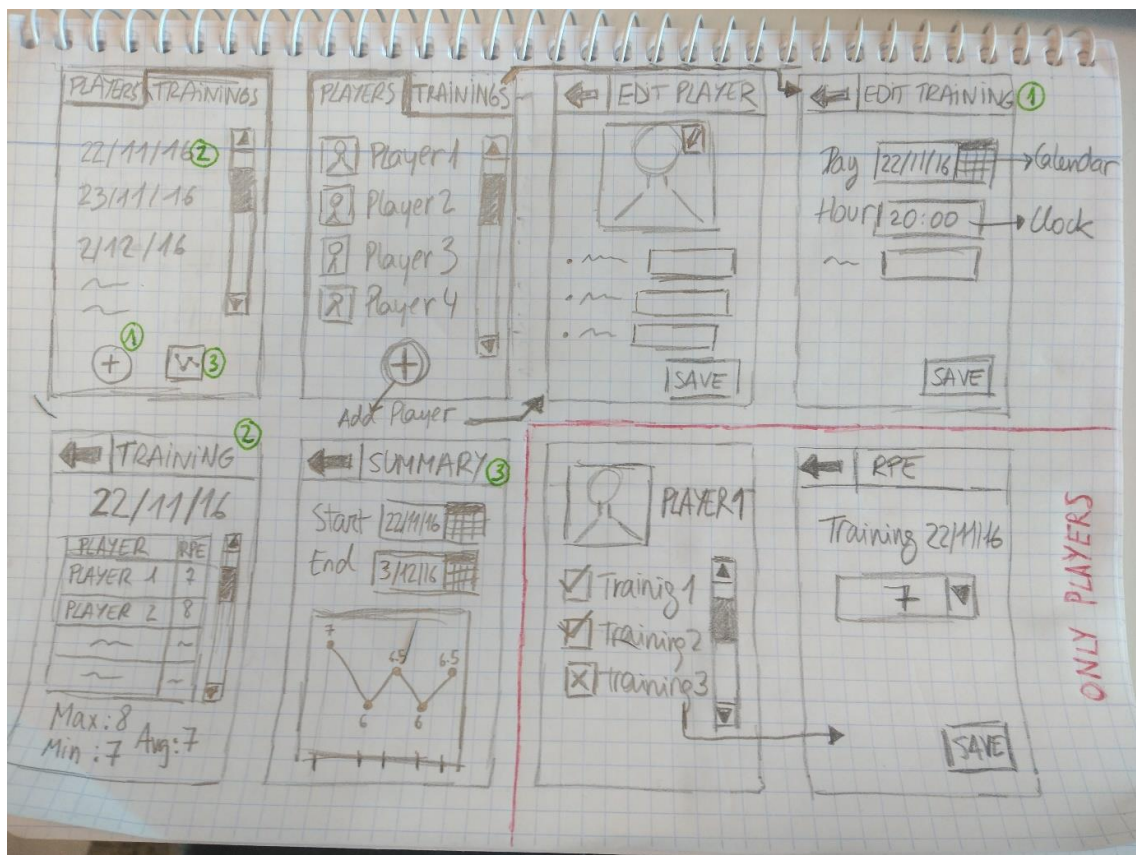
El següent apartat conté la informació relacionada amb el disseny de l'aplicació. En primer lloc, es mostrarà un disseny de les diferents pantalles, a partir d'un prototip i un arbre de navegació general amb totes les pantalles de l'aplicació. Per últim, es comentaran les decisions de disseny de l'arquitectura i dels patrons de disseny escollits per a la implementació de l'aplicació.

3.1 Prototip de baixa fidelitat

Per a iniciar la fase de disseny, el primer pas ha estat realitzar un prototip de baixa fidelitat, mitjançant esbossos a mà de les diferents pantalles plantejades per a l'aplicació.

Tal com es pot veure, les dues imatges de baix a la dreta corresponen a les pantalles que veurien els jugadors i la resta de pantalles serien les que veuria l'entrenador.

A part d'això, s'indica a quina pantalla hauria de dirigir cada acció, ja sigui clicant algun dels botons o clicant sobre algun dels elements que apareixen a les llistes.

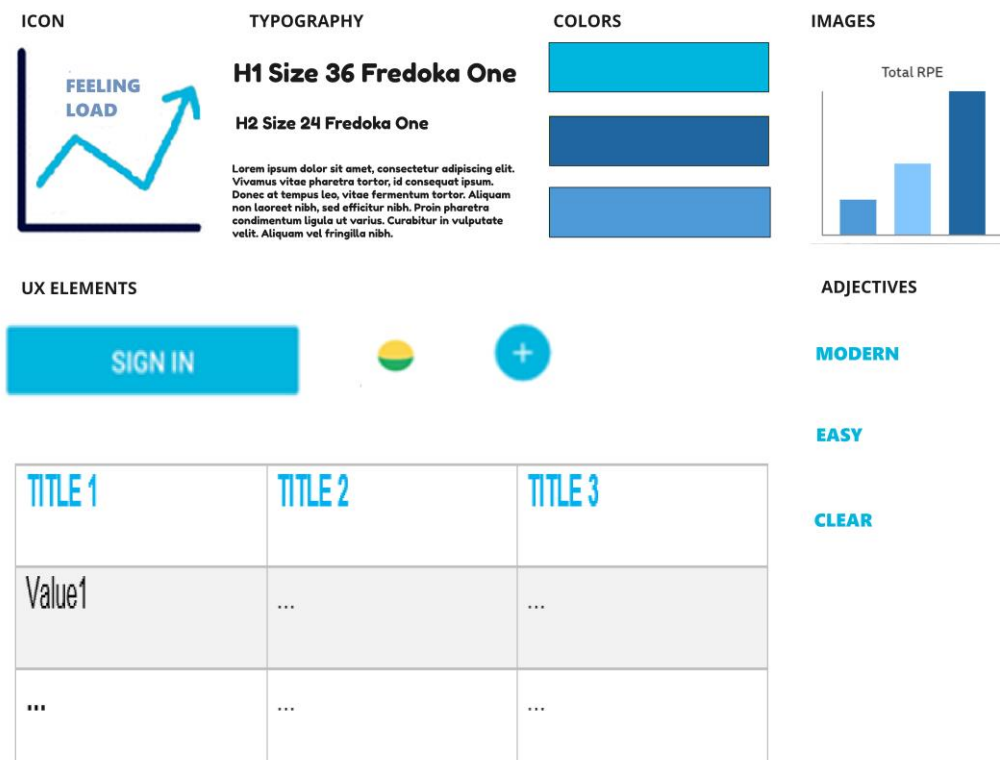


Imatge 11: Prototip de baixa fidelitat

3.2 Guia d'estil

El següent *moodboard* s'ha realitzat focalitzant els elements bàsics de l'aplicació, ja que no es tracta d'un sistema molt complex. Podem veure bàsicament que s'ha definit una gama de colors que es segueix en tots els elements, començant pel logo. Tots els elements destaquen per la senzillesa i la facilitat d'ús que s'espera de l'aplicació.

Amb aquesta guia d'estil, l'objectiu és que a l'hora d'implementar l'aplicació, totes les pantalles i els elements segueixin la mateixa línia, i permeti als desenvolupadors poder afegir elements que seguiran encaixant en l'estil.



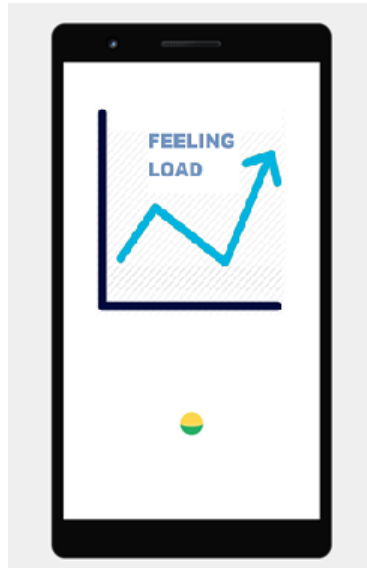
Imatge 12: Guia d'estil de l'aplicació

3.3 Prototip d'alta fidelitat

A continuació, es mostraran les diferents pantalles que formaran l'aplicació, dissenyades a partir d'un prototip funcional, realitzat amb l'eina *JustInMind*.

Pantalla inicial

La primera pantalla que trobaríem és la pantalla de càrrega, que és la pantalla inicial de l'aplicació, que simula la càrrega d'aquesta. Es pot veure el logotip i una icona que simula la càrrega.



Imatge 13: Pantalla inicial

Pantalla de login

Una vegada carregada l'aplicació, la pantalla que apareixerà és la pantalla de login, tal com es mostra a la següent imatge.



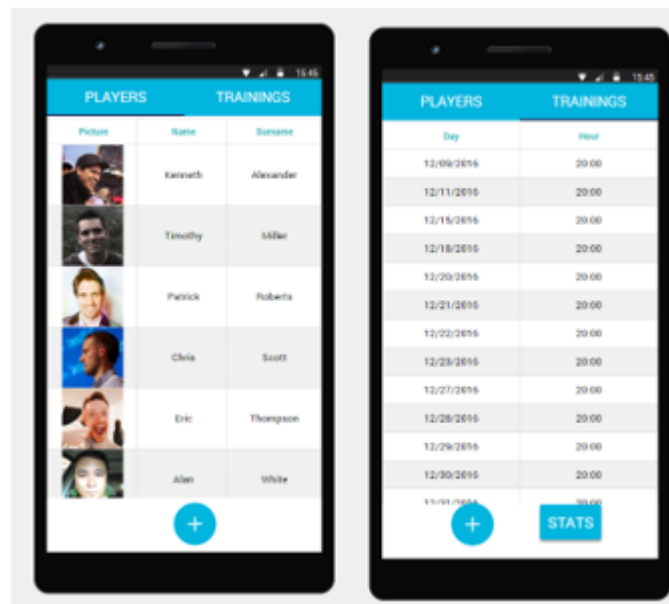
Imatge 14: Pantalla de login

Aquesta pantalla contindrà un formulari amb les dades necessàries per a iniciar sessió. Si l'usuari ja ha iniciat sessió anteriorment, aquesta pantalla s'ometrà i es passarà directament a la pantalla principal de l'usuari. També apareixerà l'opció de registre per a aquells usuaris que comencin a utilitzar el sistema. El formulari per

al registre serà el mateix que per a l'inici de sessió, amb la diferència que s'afegirà un camp que permetrà seleccionar a l'usuari si es tracta d'un entrenador o d'un jugador.

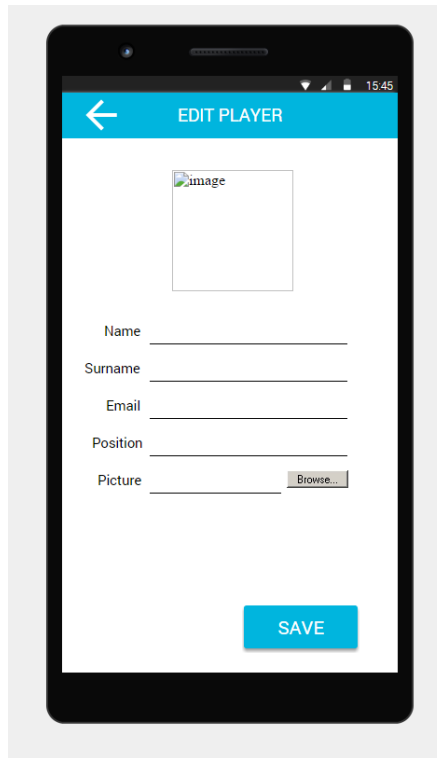
Pantalla principal d'entrenador

Aquesta pantalla és específica per als entrenadors, i conté dues pestanyes, una amb una llista dels seus jugadors i una altra amb una llista dels entrenaments disponibles. Clicant entre una pestanya i una altra, es pot canviar la visualització de la llista.



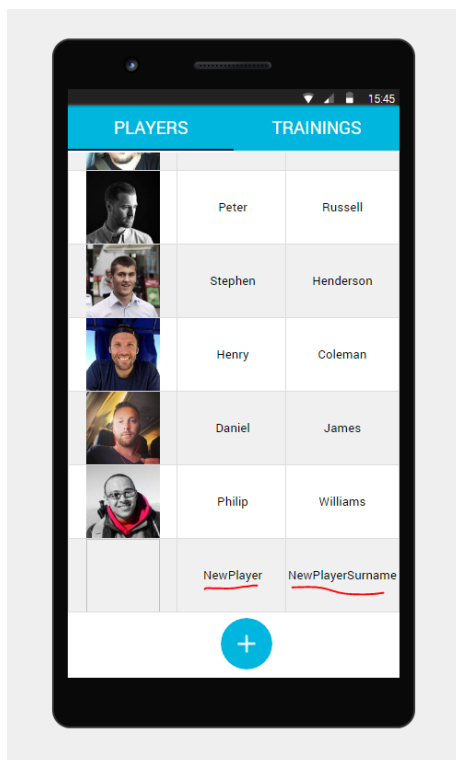
Imatge 15: Pantalla principal de l'entrenador

Tots els elements de les llistes són clicables, i cadascú tindrà el seu comportament concret. A la llista de jugadors, podem veure que hi ha un botó "+". Aquest botó ens permet afegir un nou jugador. Si el cliquem, anirem a la pantalla d'afegir jugador:



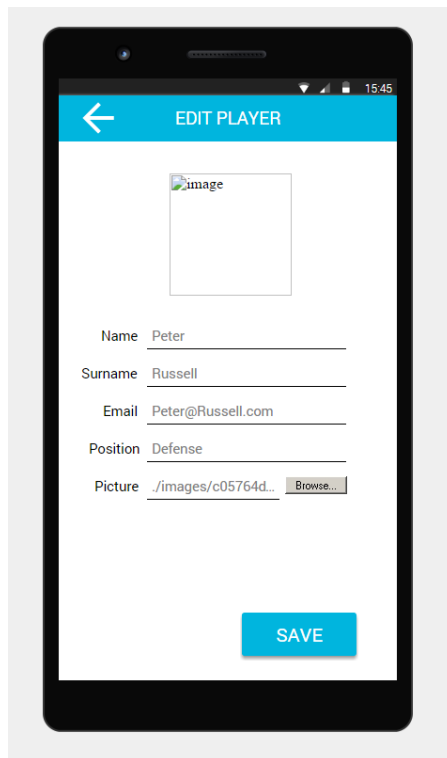
Imatge 16: Pantalla de nou jugador

Aquesta pantalla conté un formulari amb els camps necessaris per afegir un nou jugador, que són: nom, cognom, email, posició de joc i foto. Una vegada completat els camps, només cal clicar el botó "Save" per afegir-lo. Una vegada afegit el jugador, tornarem a la pantalla anterior, i a la llista podrem veure que s'ha afegit una fila amb el nou jugador.



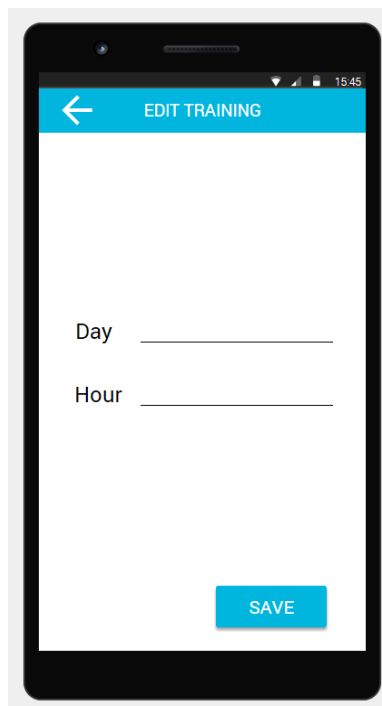
Imatge 17: Llistat de jugadors al afegir nous elements

De la mateixa manera, si cliquem qualsevol dels jugadors de la llista, podrem editar les seves dades. Al seleccionar qualsevol dels jugadors, anirem a la mateixa pantalla on podíem afegir un nou jugador, però en aquest cas, els camps vindran auto completats per defecte. Si modifiquem el jugador, al tornar a la llista podrem veure les noves dades.



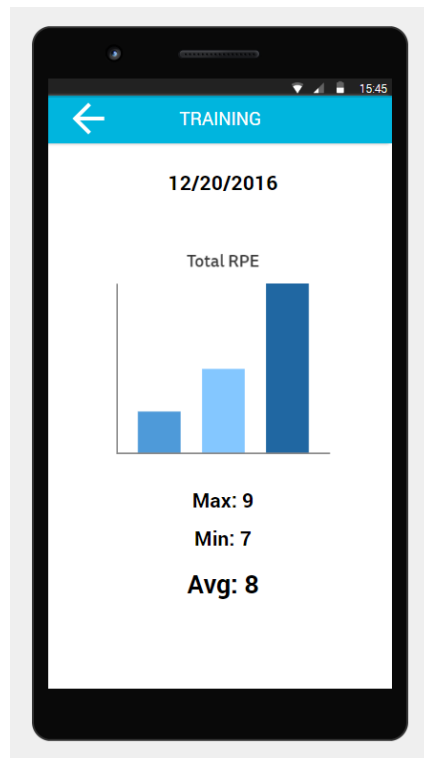
Imatge 18: Pantalla d'edició de jugador

En el cas de la llista d'entrenaments, podem veure que també hi ha un botó “+” per afegir un nou entrenament. Si cliquem aquest botó, anirem a una pantalla amb un formulari per afegir un nou entrenament, on només caldrà introduir la data i la hora d'aquest.



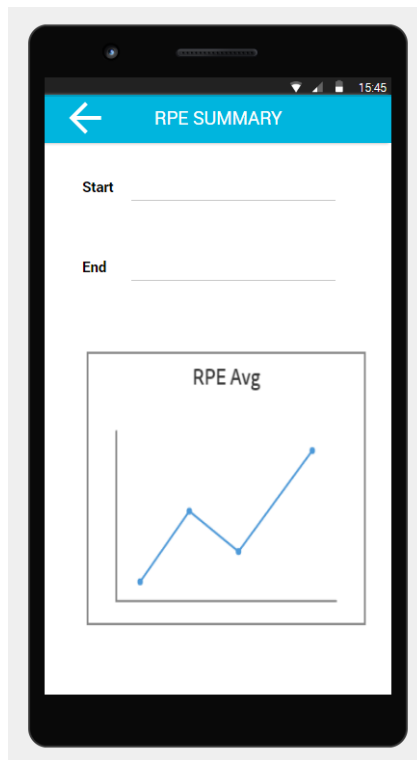
Imatge 19: Pantalla per afegir nou entrenament

Una vegada introduïdes les dades i clicat el botó “Save”, tornarem a la pantalla anterior, i a la llista d’entrenaments disponibles ens apareixerà el nou que hem afegit. Si cliquem sobre un dels entrenaments de la llista, en aquest cas, el que veuríem seria una pantalla amb el resum de les dades de RPE dels diferents jugadors que tenim recollides per aquell entrenament. En aquesta pantalla es mostraria el dia de l’entrenament així com un gràfic amb les dades de RPE. També es podria veure, a mode resum, les principals dades estadístiques de l’entrenament.



Imatge 20: Pantalla de dades d’entrenament

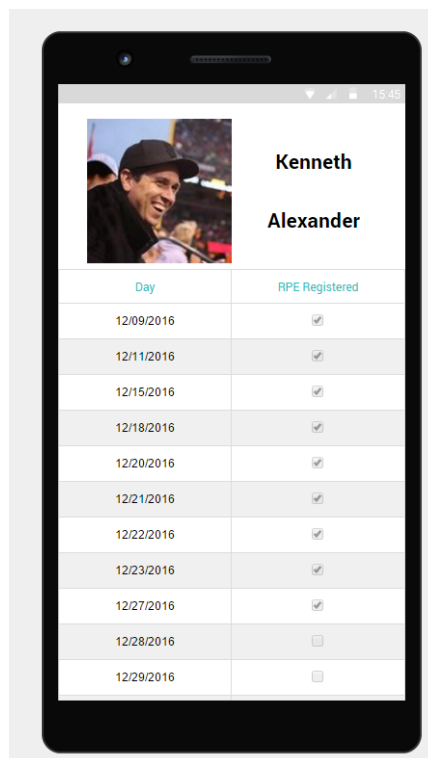
Al costat del botó per afegir un nou entrenament, podem veure que hi ha un botó “Stats”. Si cliquem aquest botó, navegariem fins a una pantalla per veure dades acumulades de diversos entrenaments. En aquesta pantalla podríem seleccionar un rang de dates, i es visualitzarien les dades de RPE per tots els entrenaments continguts entre aquelles dates.



Imatge 21: Pantalla de dades globals d'entrenaments

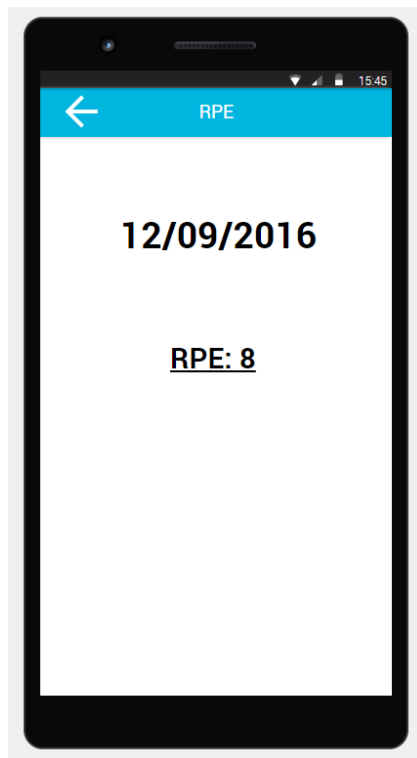
Pantalla principal del jugador

Aquesta pantalla és específica per als jugadors, i conté una llista dels entrenaments disponibles.



Imatge 22: Pantalla principal del jugador

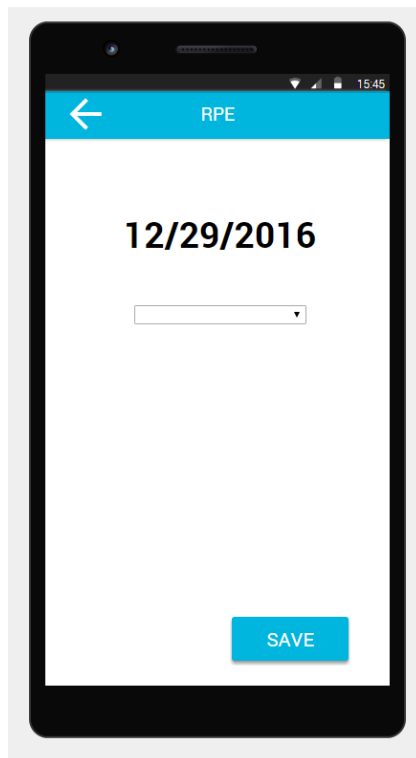
Tal com podem veure, en aquesta pantalla apareixeria el nom i el cognom del jugador que ha accedit a la app, així com la seva foto de perfil. Sota d'aquesta informació, la llista d'entrenaments disponibles, mostrant el dia i si ja s'ha registrat o no el RPE, indicat mitjançant un checkbox. En aquesta pantalla, tots els elements de la llista són clicables, però tenen un comportament diferent segons si el RPE ha estat registrat o no. Si ja ha estat registrat el RPE, al clicar sobre l'entrenament s'obrirà una pantalla amb el resum de les dades.



Imatge 23: Pantalla resum d'entrenament

Tal com podem veure, el que es mostra és simplement el dia de l'entrenament i el RPE registrat per a aquell jugador en concret. No es pot modificar cap d'aquestes dades una vegada introduïdes.

Si, en canvi, cliquem sobre un dels entrenaments que no tenen el RPE registrat, anirem a una pantalla que ens permetrà registrar aquesta dada.



Imatge 24: Pantalla de registre d'RPE

Podem veure que és una pantalla molt semblant a l'anterior, amb la diferència que ara el camp de RPE és editable, mitjançant un dropdown amb valors de l'1 al 10. Una vegada seleccionat el valor, si es clica el botó "Save", es registrarà el RPE per a aquell entrenament i tornarem a la pantalla anterior. Una vegada registrat el RPE podrem veure que aquell entrenament apareixerà a la llista amb el checkbox de RPE marcat.

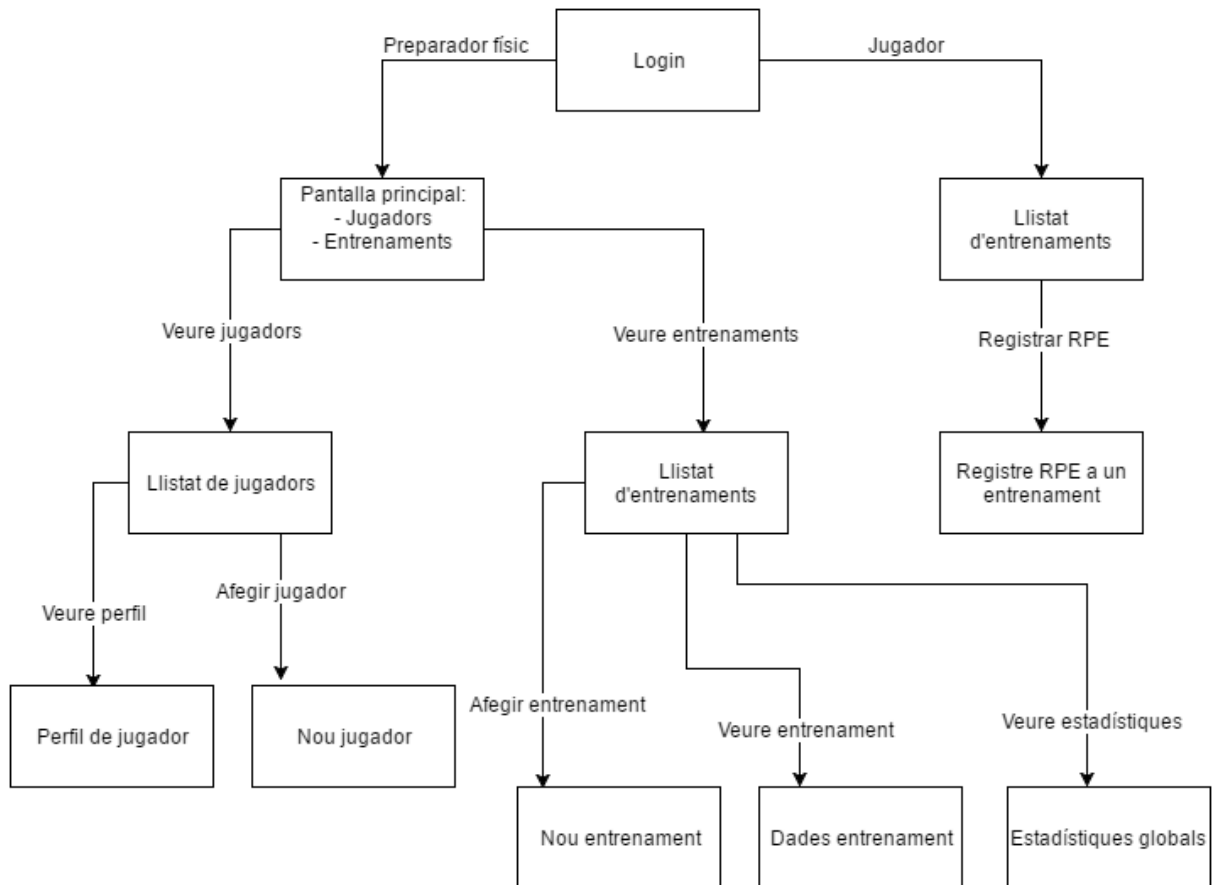
Tal com hem pogut veure, la navegació d'aquesta aplicació només té un nivell de profunditat. A totes les pantalles s'accedeix mitjançant una pantalla principal, que és diferent per als entrenadors i per als jugadors. Totes les pantalles secundaries tenen una capçalera, amb el títol de la pantalla i una fletxa que permet tornar a la pantalla anterior.



Imatge 25: Navegació de les diferents pantalles

3.4 Arbre de navegació

Després de veure el disseny de les diferents pantalles de l'aplicació i la interacció entre elles, podem veure que l'arbre de navegació resultant seria el que es mostra a la següent imatge.



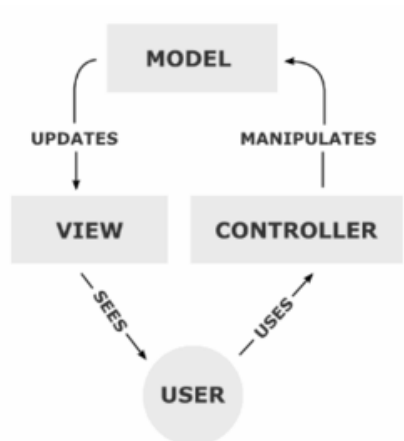
Imatge 26: Arbre de navegació de l'aplicació

3.5 Patrons de disseny

Una vegada dissenyat el prototip i la part visual de l'aplicació, cal determinar quins patrons de disseny que es seguiran per a l'arquitectura del sistema.

En aquest sentit, s'ha optat per el desenvolupament de l'aplicació Android seguint el patró MVC (model-vista-controlador). Aquest patró es basa en la separació de les dades (models) i la lògica de negoci (controladors) de la seva representació visual (vistes) [6].

A la següent imatge es pot veure de manera esquematitzada la representació del patró MVC i com els diferents elements que es defineixen interactuen entre si.

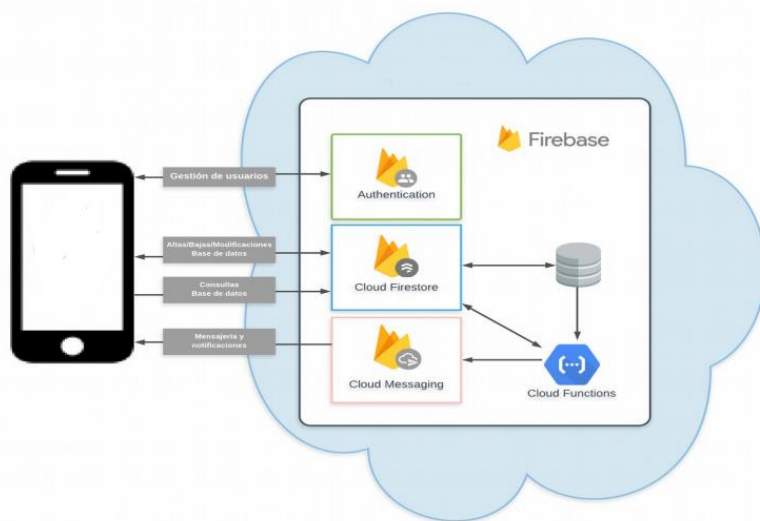


Imatge 27: Patró MVC [7]

3.5.1 Persistència de dades

Per a la persistència de dades i la comunicació client-servidor, s'ha optat per utilitzar Firebase, ja que és una plataforma amb una versió gratuïta que ens permet guardar dades en un servidor cloud, i utilitzar el seu SDK de manera fàcil per a poder realitzar la comunicació entre l'usuari i el servidor.

A part del tema de la persistència i consulta de dades, Firebase també ofereix un servei d'autenticació i gestió d'usuaris, que per a la nostra aplicació serà molt útil a l'hora d'implementar tot el sistema de registre i login.



Imatge 28: Comunicació client-servidor mitjançant Firebase

4. Implementació

En aquest apartat s'explicaran les eines que han estat necessàries per a realitzar el projecte. A part d'això, es mostrarà l'estructura del codi generat i les funcionalitats bàsiques que s'han implementat, explicant les llibreries i tècniques utilitzades per a desenvolupar les principals funcionalitats de l'aplicació.

4.1 Eines emprades

Plataforma i sistema operatiu

Per a la realització d'aquest projecte, s'ha treballat amb un sistema operatiu Windows 10 i amb la plataforma de desenvolupament d' Android Studio 3.2. S'ha escollit aquesta plataforma ja que ofereix un entorn molt pràctic per a desenvolupar aplicacions en el llenguatge escollit i que a més, és l'entorn de desenvolupament oficial que Google proporciona per al desenvolupament d'aplicacions Android.

Llenguatge de programació

El llenguatge escollit ha estat Java, degut a que ja s'havia utilitzat anteriorment, i això reduïa considerablement la corba d'aprenentatge que hagués suposat utilitzar algun altre llenguatge.

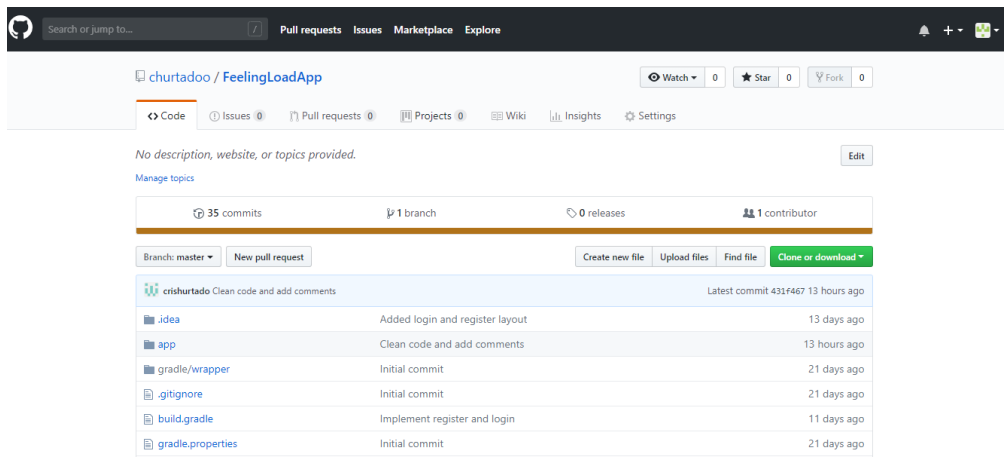
Llibreries

Per a resoldre algun dels problemes plantejats a l'hora d'implementar les diferents funcionalitats de l'aplicació, s'han utilitzat algunes llibreries de tercers, com per exemple per a la realització de les gràfiques, que han permès una gran agilitat en la implementació.

Repositori de codi

Un dels majors problemes que pot aparèixer a l'hora de desenvolupar un sistema software és perdre el codi que es va generant. Això seria un greu problema ja que faria endarrerir molt la feina i podria provocar que el projecte no s'acabés ens els terminis fixats. Per a que això no passi, és molt important guardar aquest codi en un repositori, que a la vegada ens permeti recuperar versions antigues si fos necessari. En aquest cas, s'ha escollit GitHub, un repositori en el núvol que permet anar fent còpies de seguretat del codi quan l'usuari ho desitja i recuperar aquestes versions si fos necessari.

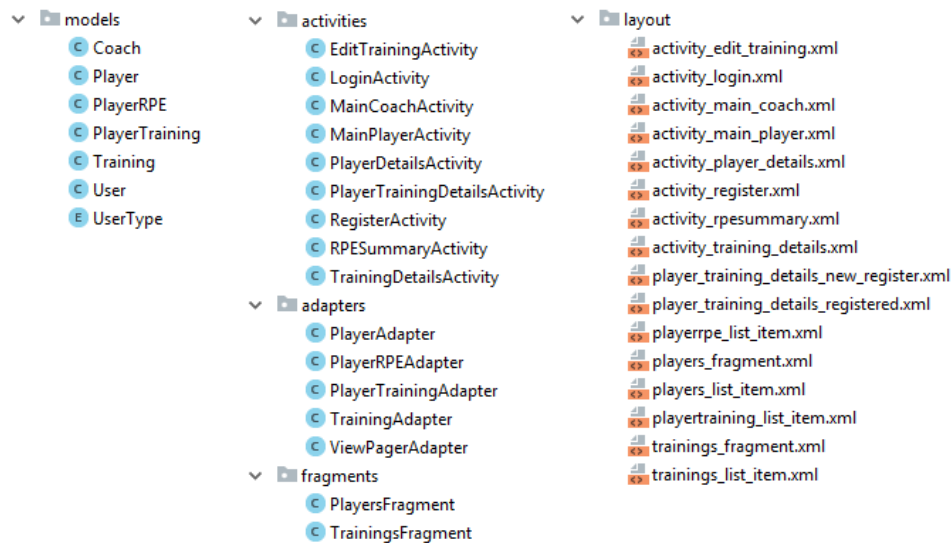
Es pot accedir al repositori públic que conté el codi generat a partir del següent enllaç: <https://github.com/churtadoo/FeelingLoadApp>



Imatge 29: Repositori GitHub de l'aplicació implementada

4.2 Estructura del codi

Tal com s'ha explicat a l'apartat de disseny, s'ha escollit una arquitectura MVC. Per tal de seguir aquestes directrius marcades en el disseny, el codi s'ha estructurat de la següent manera:

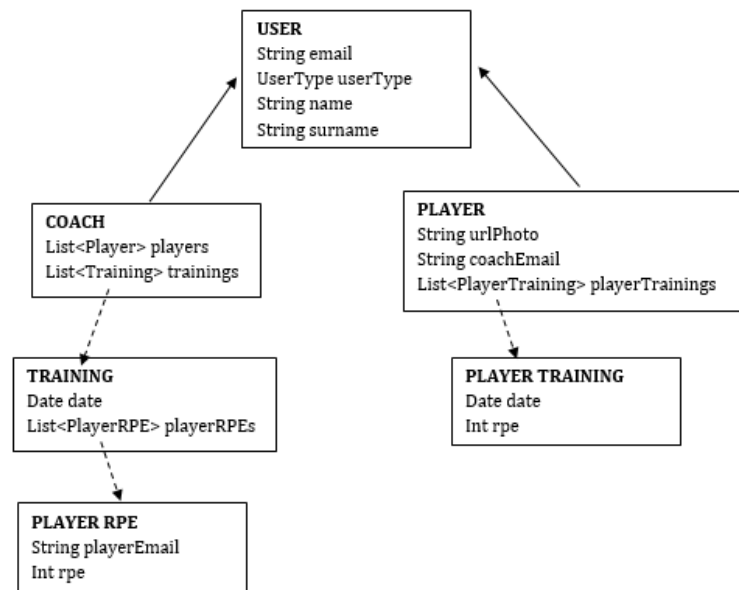


Imatge 30: Estructura del codi

En aquest cas, tindriem els models, que són els encarregats de gestionar tota la lògica de negoci. Per altra banda, tindriem les vistes, que en aquest cas serien els layouts de cada activitat, i per últim, les classes que farien la funció del controladors, serien les pròpies activities, ja que son les que responen als diferents events que es produeixen a la interfície gràfica.

Es poden veure els atributs de cadascun dels models a la següent imatge, que mostra la interacció entre ells. Els models de *Coach* i *Player* son una herència de la classe *User*, ja que això ens permet autenticar tots els usuaris de la mateixa manera i decidir a quin tipus d'usuari correspon mitjançant la propietat *userType*.

A part, també ens permet reutilitzar codi, ja que comparteixen algunes propietats i mètodes.



Imatge 31: Estructura dels models utilitzats

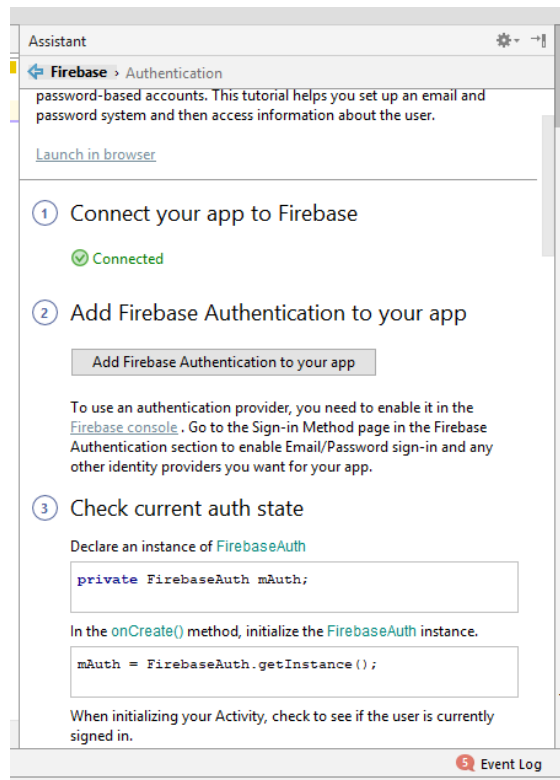
4.3 Funcionalitats principals

A continuació, s'explicarà la implementació de les principals funcionalitats del sistema, quines llibreries s'han utilitzat i es mostraran fragments de codi per a exemplificar-ho.

4.3.1 Registre i login

Per a l'autenticació d'usuaris, s'ha utilitzat Firebase Auth, ja que ens permet de manera fàcil registrar usuaris a partir d'email i contrasenya.

Per a poder utilitzar aquests mètodes, el primer que cal és afegir Firebase al nostre projecte. Utilitzant Android Studio, és molt fàcil aquesta integració, ja que conté una eina que ens permet connectar el nostre projecte:



Imatge 32: Integració de Firebase a Android Studio

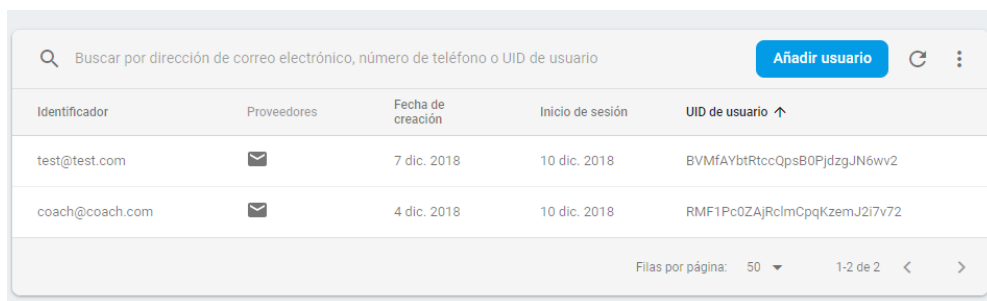
El primer que cal, una vegada connectat el nostre projecte a Firebase, és afegir la referència a l'arxiu build.gradle de la app:

```
implementation 'com.google.firebase:firebase-auth:16.0.5'
```

Una vegada afegida al referència, podem registrar un usuari de la següent manera:

```
mAuth = FirebaseAuth.getInstance();
mAuth.createUserWithEmailAndPassword(email, password)
```

Després de registrar un usuari, podem gestionar-lo també a través de la consola de Firebase, que ens permet veure tots els usuaris registrats, eliminar-los, canviar la contrasenya, etc.



Imatge 33: Usuaris registrats a l'aplicació de Firebase

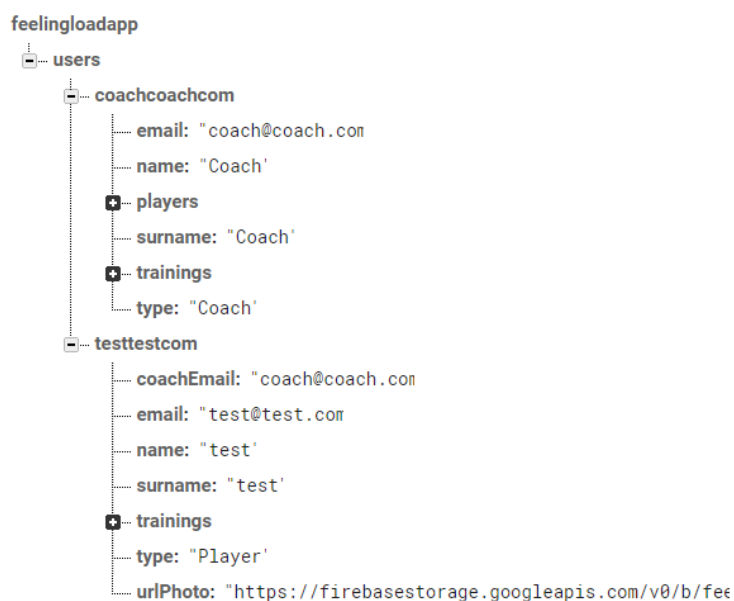
En quant al login, simplement cal cridar al següent mètode de Auth:

```
FirebaseAuth.getInstance().signInWithEmailAndPassword(email, password)
```

4.3.2 Gestió de dades

Una vegada registrats els usuaris, és important identificar de quin tipus d'usuari es tracta, ja que depenent d'això, es redirigirà a una pantalla o a una altra. Una vegada registrat un usuari, el que s'ha fet és generar un perfil de l'usuari a Firebase Database. Aquests usuaris, contindran les dades que hem vist anteriorment als models utilitzats.

A la següent imatge, podem veure una captura amb les dades de dos usuaris registrats, un entrenador i un jugador. Al ser una base de dades no relacional, ens permet tenir estructures diferents per a cada registre.



Imatge 34: Registre d'usuaris a Firebase Database

Per a poder utilitzar Firebase Database al nostre projecte, cal afegir la següent referència a l'arxiu gradle.build:

```
implementation 'com.google.firebase:firebase-database:16.0.5'
```

Un tema important a tenir en compte aquí, és que no necessàriament ha d'existir una relació entre els usuaris registrats i els usuaris creats a la base de dades. Això és així perquè els jugadors poden ser creats per part de l'entrenador. Si es dona el cas que un entrenador crea un jugador que encara no s'ha registrat a l'aplicació, es crearà el registre a la base de dades, però fins que aquell jugador no es registri, no es podrà autenticar i accedir a l'aplicació. Una vegada es registri, el que es farà és crear el registre a Firebase Auth i recollir les dades que ja tenia creades a la base de dades, de tal manera que si l'entrenador li havia afegit entrenaments, per exemple, ja els tindria disponibles.

4.3.3 Llistes

Una de les funcionalitats que es repeteix en diverses pantalles de l'aplicació són les llistes dinàmiques. Per a implementar aquestes llistes, s'ha utilitzar un RecyclerView, que mitjançant un adaptador, ens permet omplir la llista amb els nostres objectes de domini fàcilment.

Per a poder implementar això, el que necessitem és un RecyclerView a la vista on volem mostrar la llista:

```
<android.support.v7.widget.RecyclerView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/playertrainings_list"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"

tools:context="edu.uoc.churtado.feelingloadapp.activities.MainPlayerAc
tivity"
    tools:listitem="@layout/playertraining_list_item"
    android:requiresFadingEdge="vertical"
    android:fadingEdgeLength="40dp"
    android:scrollbars="vertical"/>
```

També necessitem un layout per als items de la llista. Aquest layout ens permetrà combinar les diferents propietats de l'objecte que volem representar, per exemple, combinant camps de text amb imatges.

Per últim, s'ha creat un adaptador, que es una classe que ens permet recollir la llista de dades a mostrar i utilitzar el layout de cada item per mostrar les diferents propietats:

```
public class PlayerTrainingAdapter extends
RecyclerView.Adapter<PlayerTrainingAdapter.ViewHolder> {
    private final List<PlayerTraining> playerTrainings;
```

Veiem a continuació com es mostraria un element d'aquest adaptador:

```
@Override
public void onBindViewHolder(@NonNull final ViewHolder holder, int
position) {
    //Get PlayerTraining to show
    holder.item = playerTrainings.get(position);
    //Get and set background color for list element
    holder.mView.setBackgroundResource(getBackgroundColor(position));
    //Set training date and check if player has registered rpe
    SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy
HH:mm", Locale.UK);

    holder.trainingDate.setText(dateFormat.format(playerTrainings.get(posi
tion).getDate()));

    holder.rpeRegistered.setChecked(playerTrainings.get(position).HasRegis
teredRPE());
```



```

holder.mView.setTag(position);
holder.mView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int currentPos = (int) v.getTag();
        Context context = v.getContext();
        //If element is clicked, go to
        PlayerTrainingDetailsActivity
        Intent intent = new Intent(context,
        PlayerTrainingDetailsActivity.class);
        intent.putExtra(ARG_ITEM_ID, currentPos);
        //Start the new activity
        context.startActivity(intent);
    }
});
}

```

El mètode `onBindViewHolder` ens permet mostrar els diferents elements de la llista que li hem indicat a l'adaptador. Tal com podem veure, també podem gestionar els events de click dels elements. En aquest cas, per exemple, veiem que al clicar un dels elements, s'iniciaria una nova activitat, passant com a argument, la posició de l'element que s'ha clicat, per tal de poder recuperar-lo posteriorment.

4.3.4 Mostrar imatges d'usuari

Una altra de les funcionalitats que es repeteix en diversos punts de l'aplicació, es la de mostrar imatges de perfil de l'usuari.

En primer lloc, veiem com s'han guardat les imatges pujades pels usuaris. Per a fer això, s'ha utilitzat `Firebase Storage`, un gestor d'arxius que ens permet guardar arxius de manera fàcil i obtenir una url per a descarregar aquests arxius. Per a poder utilitzar `Firebase Storage`, el primer pas és afegir la referència:

```
implementation 'com.google.firebase:firebase-storage:16.0.5'
```

Una vegada pujades i guardades les imatges, cal mostrar-les a les diferents pantalles. Per a fer això, s'ha utilitzat la llibreria `Picasso`, que ens permet de manera molt fàcil mostrar imatges a partir d'una Url.

Per a poder utilitzar aquesta llibreria, cal la següent referència:

```
implementation 'com.squareup.picasso:picasso:2.71828'
```

I carregar les imatges amb la següent crida:

```
Picasso.get().load(currentPlayerPhoto).into(playerPhoto);
```

On `playerPhoto` és un `ImageView` de la vista i `currentPlayerPhoto` és la url de la imatge a mostrar.

Per últim, relacionat amb el tractament d'imatges, també s'ha utilitzat una llibreria per a mostrar imatges amb una forma circular, que es mostra en la pantalla

principal del jugador. S'ha decidit utilitzar aquesta llibreria ja que és molt fàcil d'utilitzar i que d'altra manera, no és tan fàcil de resoldre.

La llibreria i la referència utilitzades per això ha estat:

```
implementation 'de.hdodenhof:circleimageview:2.2.0'

<de.hdodenhof.circleimageview.CircleImageView
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/player_photo"
    android:layout_width="120dp"
    android:layout_height="120dp"
    app:civ_border_width="2dp"
    app:civ_border_color="@color/white"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="20dp"/>
```

4.3.5 Pestanyes

A la pantalla principal de l'entrenador, tenim una vista amb pestanyes, que ens permet tenir les llistes de jugadors i d'entrenaments a la mateixa pantalla i canviar d'una a altra fàcilment, sense haver de carregar una nova activitat. Per a poder implementar aquest sistema de pestanyes, el primer que necessitem és un TabLayout i un ViewPager, que serà el que mostrarà el contingut de cada pestanya:

```
<android.support.design.widget.TabLayout
    android:id="@+id/tabs"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:tabMode="fixed"
    android:background="@color/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    app:tabIndicatorColor="@color/colorPrimaryDark"
    app:tabGravity="fill"
    app:tabTextColor="#FFF"
    app:tabSelectedTextColor="@color/colorPrimaryDark" />

<android.support.v4.view.ViewPager
    android:id="@+id/pager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

I per inicialitzar els elements de les pestanyes, l'únic que cal és afegir el següent codi a l'activitat:

```
//Initialize adapter to show tabs
ViewPager viewPager = findViewById(R.id.pager);
ViewPagerAdapter adapter = new
ViewPagerAdapter(getSupportFragmentManager());
// Add Fragments to adapter one by one
adapter.addFragment(new PlayersFragment(), "PLAYERS");
adapter.addFragment(new TrainingsFragment(), "TRAININGS");
viewPager.setAdapter(adapter);

TabLayout tabLayout = findViewById(R.id.tabs);
tabLayout.setupWithViewPager(viewPager);
```

Tal com podem veure, el que es necessita és associar un adaptador per al ViewPager, que es el que mostrarà el contingut. El que fem és crear un Fragment per a cada pestanya. Aquest fragment és el que inicialitzarà les dades que ha de mostrar a la vista.

4.3.6 Gràfics

Un altre dels elements importants de l'aplicació són els gràfics de dades. Per a poder mostrar els gràfics, s'ha utilitzat la llibreria GraphView, que necessita de la següent referència:

```
implementation 'com.jjoe64:graphview:4.2.2'
```

El layout necessari per mostrar el gràfic és molt senzill, i és el mateix indiferentment del tipus de gràfic o de dades que volem mostrar:

```
<com.jjoe64.graphview.GraphView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/rpe_summary" />
```

Per a afegir les dades al gràfic, només cal crear el tipus de gràfic desitjat i afegir-li les dades a mostrar:

```
//Draw bar graph with rpe information
BarGraphSeries<DataPoint> series = new BarGraphSeries<>(new
DataPoint[] {
    new DataPoint(0, getRpeCount(0)),
    new DataPoint(1, getRpeCount(1)),
    new DataPoint(2, getRpeCount(2)),
    new DataPoint(3, getRpeCount(3)),
    new DataPoint(4, getRpeCount(4)),
    new DataPoint(5, getRpeCount(5)),
    new DataPoint(6, getRpeCount(6)),
    new DataPoint(7, getRpeCount(7)),
    new DataPoint(8, getRpeCount(8)),
    new DataPoint(9, getRpeCount(9)),
    new DataPoint(10, getRpeCount(10))
});
graph.addSeries(series);
```

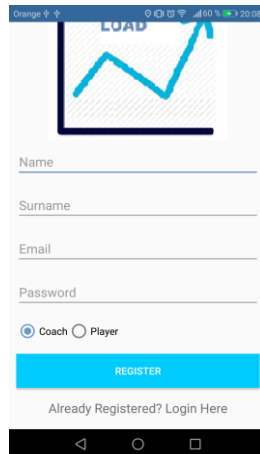
Amb això, en tindríem prou per a mostrar el gràfic. A partir d'aquí, la llibreria ofereix mètodes que ens permeten editar alguns aspectes del gràfic, com el color dels diferents elements o la mida.

4.4 Canvis de disseny

En general, s'ha respectat el disseny presentat al capítol anterior, però durant el desenvolupament s'ha cregut convenient realitzar alguns canvis i afegir alguna informació extra en algunes pantalles. A continuació s'expliquen i es justifiquen els diferents canvis realitzats.

4.4.1 Login i registre

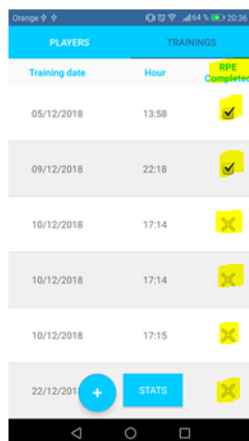
La pantalla de login segueix el mateix disseny que es mostrava al prototip d'alt nivell. La pantalla de registre, que no es va mostrar, conté un formulari per a registrar un nou usuari. En aquest formulari, a part dels camps de email i contrasenya, s'han afegit tres camps més que permeten completar la informació de l'usuari. Aquests camps són el nom, el cognom i una opció per escollir el tipus d'usuari, entrenador o jugador.



Imatge 35: Pantalla de registre

4.4.2 Pantalla principal d'entrenador

Aquesta pantalla mostra les dades dels jugadors i entrenaments associats a l'entrenador, amb els botons per afegir noves dades. L'única diferència és que a la llista d'entrenaments, s'ha afegit un nou camp a cada element, que mostra una icona segons si per a aquell entrenament, tots els jugadors han registrat el seu RPE o no. Aquesta informació s'ha afegit per a poder oferir visualment, una dada important per a l'entrenador de manera ràpida, sense haver d'anar als detalls de l'entrenament.

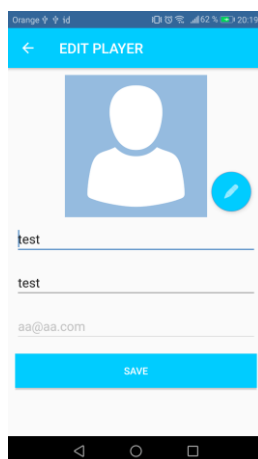


Imatge 36: Pantalla principal d'entrenador amb la nova columna afegida

4.4.3 Pantalla de creació o edició de jugador

En aquest formulari, s'ha suprimit el camp "Position", ja que és una informació que s'ha decidit no guardar per tal de simplificar el model, ja que no aportava una dada imprescindible. A part d'això, s'ha canviat el botó per seleccionar una imatge i s'ha col·locat al costat de la imatge que es mostra.

Per últim, indicar que si ens trobem editant un jugador que ja existia, tots els camps són editables excepte el camp email.

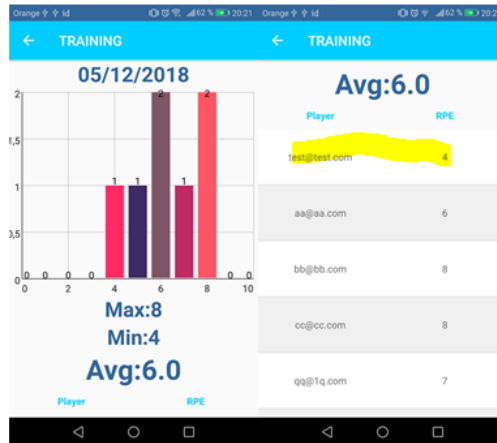


Imatge 37: Pantalla d'edició de jugador

4.4.4 Pantalla de detalls d'entrenament

En aquesta pantalla, a part de les dades de l'entrenament que ja es mostraven al prototip, s'ha decidit afegir una llista resumint l'RPE registrat per cada jugador. Aquesta informació s'ha afegit, ja que és important no només tenir una visió global de les dades registrades, sinó també saber quin és el registre de cada jugador, i si algun dels jugadors encara no ha registrat la dada.

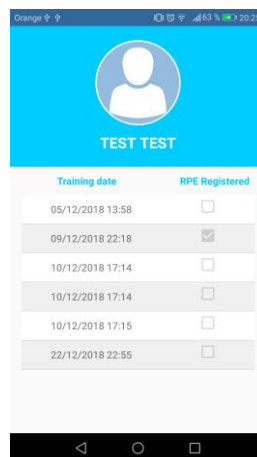
Per mostrar totes aquestes dades en la mateixa pantalla, el que s'ha fet és que la vista tingui un scroll vertical, de tal manera que podem visualitzar totes les dades.



Imatge 38: Pantalla de detalls d'entrenament amb la nova informació afegida

4.4.5 Pantalla principal de jugador

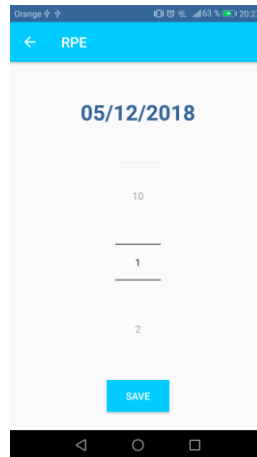
En aquesta pantalla, l'únic que s'ha canviat ha estat la presentació de les dades del jugador, que es mostren davant d'un fons del color principal de l'aplicació, amb una imatge arrodonida.



Imatge 39: Pantalla principal del jugador

4.4.6 Registre de RPE

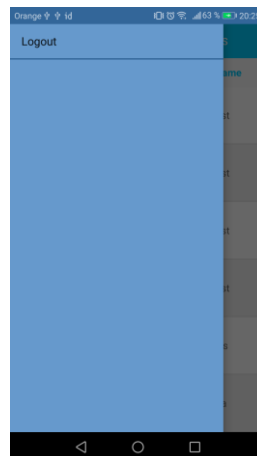
En aquesta pantalla, on el jugador enregistra el RPE per a un entrenament concret, l'única diferència amb el prototip és que en comptes de mostrar un desplegable amb les opcions disponibles, es mostra un selector de número, que fa més fàcil aquest recull de dades.



Imatge 40: Pantalla de registre de l'RPE

4.4.7 Menú lateral

Tant a la pantalla principal del jugador com a la de l'entrenador, s'ha afegit un menú lateral, al qual s'ha afegit una opció per tancar la sessió.



Imatge 41: Menú lateral afegit

4.5 Problemes sorgits durant el desenvolupament

El problema principal sorgit durant la implementació, ha estat el desconeixement de Firebase, que s'ha utilitzat tant per a la gestió d'usuaris com per a la gestió de dades. El fet de que hi hagi dos tipus d'usuaris i que a més comparteixen dades entre si, ha estat el que més ha dificultat el desenvolupament.

Per a solucionar-ho, s'ha optat per repetir dades entre els usuaris, i no optimitzar les cerques dintre de Firebase amb els índexs. Aquesta solució, encara que no és la més eficient i recomanable, ha servit per poder complir els terminis i acabar una aplicació funcional, i amb coherència de dades.

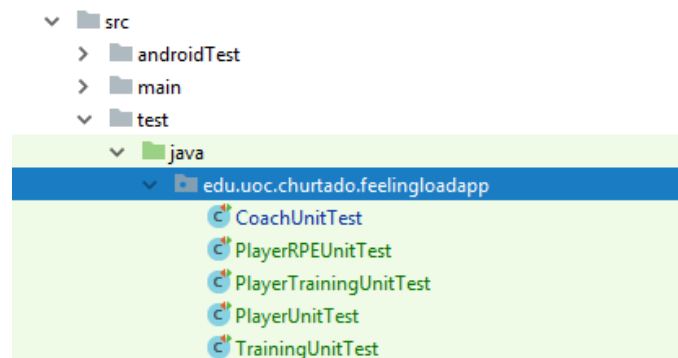
5. Valoració de la implementació

Una vegada realitzada la implementació de l'aplicació, cal valorar i provar-la per tal d'assegurar-nos que compleix tots els requisits que es van definir en l'apartat d'anàlisi. Per a fer això, el que s'ha fet és implementar una sèrie de tests unitaris sobre els models implementats, per tal d'assegurar que la lògica de negoci que contenen aquests models és correcta.

Per una altra banda, s'han fet una sèrie de proves sobre les diferents funcionalitats, analitzant quin són els requisits i casos d'ús que s'han complert.

5.1 Proves unitàries

Mitjançant la llibreria JUnit, s'han realitzat una sèrie de tests unitaris sobre els models que contenen lògica de negoci. Podem trobar aquests tests dins la carpeta **app > src > test**.



Imatge 42: Tests unitaris

Veiem com a exemple un d'aquests tests. Dins de la classe *CoachUnitTest*, tenim el següent test:

```
@Test
public void registerRpe_isCorrect() {
    Coach coach = new Coach();

    Player player = new Player();
    player.setEmail("test@test.com");
    player.setName("Test");
    player.setSurname("Surname");
    coach.addPlayer(player);

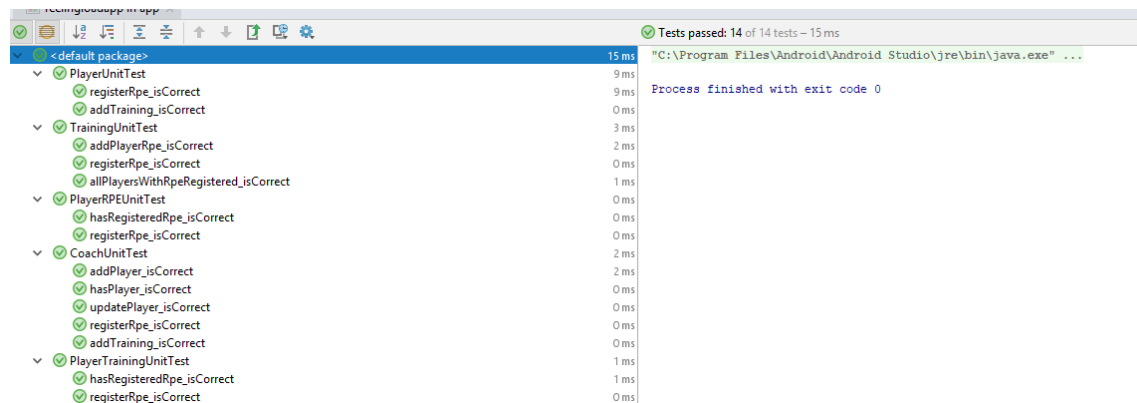
    Date date = new Date();
    coach.addTraining(date);

    coach.registerRpe(date, 7, "test@test.com");

    assertEquals(7,
coach.getTrainings().get(0).getRPEs().get(0).getRPE());
}
```


Tal com podem veure, el que fem és crear una instància de l'objecte sobre el qual volem aplicar el test, en aquest cas, la classe *Coach*. A partir d'aquí, realitzem les operacions necessàries per a poder testejar el mètode que volem, en aquest cas, el mètode és *registerRpe*. Una vegada cridat el mètode, el que fem és comprovar que realment es compleix el resultat esperat, en aquest cas, comprovem que s'ha registrat un rpe amb valor 7 per al jugador "test@test.com".

Si executem tots els tests implementats, podem veure que tots s'executen correctament i es produeix el resultat esperat.



Imatge 43: Execució dels tests unitaris

Tenir aquests tests unitaris ens permet assegurar que la nostra lògica de negoci seguirà funcionant correctament si es fan canvis en algun punt de l'aplicació.

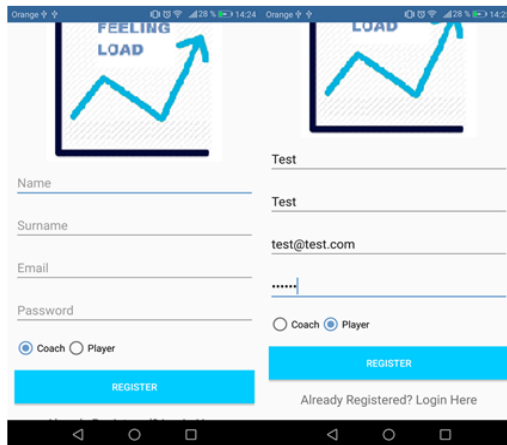
5.2 Proves funcionals

El segon conjunt de proves han estat sobre les diferents funcionalitats de l'aplicació, per tal de garantir que es compleixen tots els requisits i casos d'ús especificats a l'apartat de disseny.

Per a la realització de les proves, el que s'ha fet és agafar cada cas d'ús i seguir els passos indicats a l'escenari principal, per tal de comprovar si realment es complien les condicions indicades. També s'han realitzat proves sobre els escenaris alternatius, si es que n'hi havia.

5.2.1 Registre - CU1

L'aplicació ens mostra el formulari de registre, amb les dades requerides. Una vegada omplert el formulari amb les dades, i clicant el botó "Register", comprovem que el nou usuari s'ha registrar correctament a la base de dades.

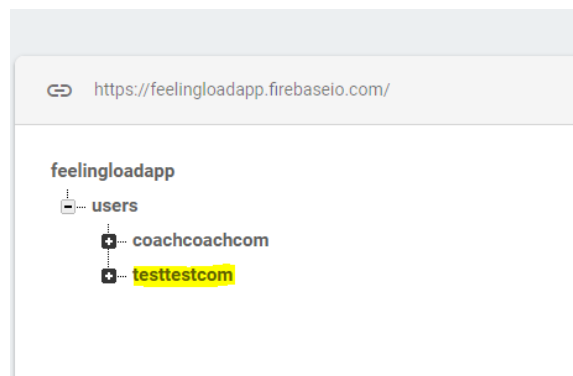


Imatge 44: Pantalla de registre i formulari amb dades introduïdes

Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
coach@coach.com	✉	4 dic. 2018	11 dic. 2018	RMF1Pc0ZAJRclmCpqKzemJ2i7v72
test@test.com	✉	12 dic. 2018	12 dic. 2018	qliiWIKo5ysRQAX0thhbqqiyWOBx1

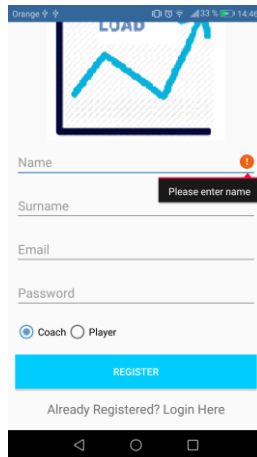
Filas por página: 50 1-2 de 2

Imatge 45: Registre del nou usuari a Firebase Auth



Imatge 46: Registre del nou usuari a Firebase Database

Si l'usuari no envia alguna de les dades, es mostra un missatge d'error al camp corresponent.



Imatge 47: Formulari de registre amb missatge d'error

Si l'usuari ja existeix al sistema, es mostra un missatge d'error.



Imatge 48: Missatge d'error a l'intentar registrar un usuari que ja existeix

5.2.2 Inici de sessió – CU2

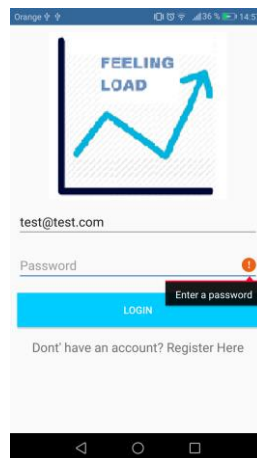
L'aplicació ens mostra el formulari de inici de sessió, amb les dades requerides, que són el correu i la contrasenya. Una vegada omplert el formulari amb les dades d'un usuari existent, i clicant el botó "Login", comprovem que el sistema inicia la sessió i ens mostra la pantalla principal de l'usuari.

Si agafem com a exemple l'usuari registrat en el cas anterior, que hem comprovat que existeix al sistema, comprovem el seu inici de sessió.



Imatge 49: Pantalla d'inici de sessió, introducció de dades i pantalla principal de l'usuari

Igual que en el cas anterior, si alguna de les dades del formulari no s'envia, es mostra un missatge d'error en el camp en qüestió.



Imatge 50: Missatge d'error al no enviar totes les dades del formulari

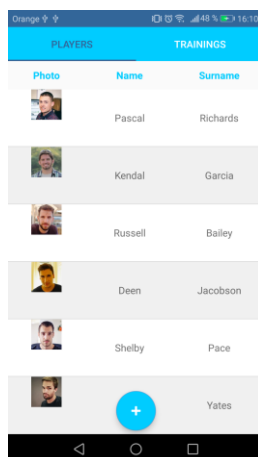
Per últim, si introduïm les dades d'un usuari que no s'ha registrat prèviament, es mostra un missatge d'error.



Imatge 51: Missatge d'error al introduir dades errònies

5.2.3 Veure jugadors – CU3

Una vegada un usuari de tipus entrenador ha iniciat sessió, se li mostrarà la seva pantalla principal, que conté una llista dels seus jugadors.

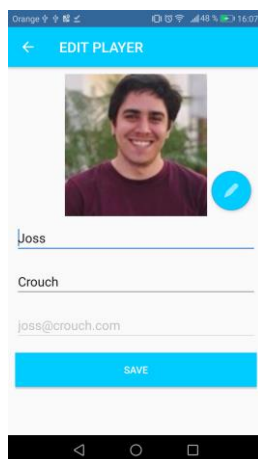


Imatge 52: Llistat de jugadors d'un entrenador

5.2.4 Veure perfil de jugador – CU4

Un usuari de tipus entrenador que ha iniciat sessió, a través de la seva pantalla principal on hi ha un llistat dels seus jugadors, pot visualitzar el perfil del jugador clicant qualsevol dels elements de la llista.

Una vegada clicat, es mostrarà una pantalla amb les dades del jugador, i a més es permetrà editar-les.



Imatge 53: Pantalla de detalls de jugador

Si, per exemple, canviem el nom i el cognom del jugador per "Test" i "Test", podem comprovar que si editem les dades del jugador, aquestes es modifiquen a la base de dades.

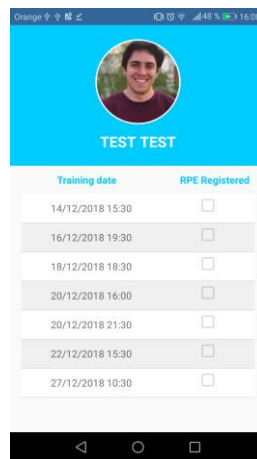
```

josscrouchcom
  coachEmail: "coach@coach.com"
  displayName: "Test Test"
  email: "joss@crouch.com"
  name: "Test"
  surname: "Test"
  trainings
    type: "Player"
    urlPhoto: "https://firebasestorage.googleapis.com/v0/b/fee

```

Imatge 54: Actualització de dades de jugador a Firebase Database

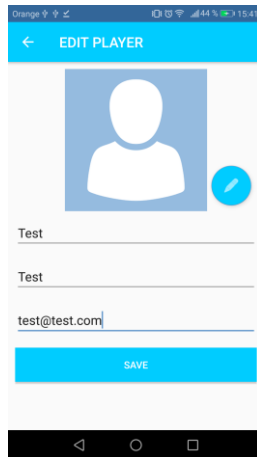
També podem comprovar que si editem alguna de les dades del jugador i després iniciem sessió amb aquest mateix jugador, les dades apareixen actualitzades.



Imatge 55: Pantalla principal de jugador amb les dades modificades

5.2.5 Afegir jugador – CU5

Un usuari de tipus entrenador que ha iniciat sessió, a la seva pantalla principal on apareixen els seus jugadors, pot afegir un nou jugador clicant el botó “+”. Una vegada clicat, apareixerà un formulari amb les dades necessàries per a afegir un nou jugador. Un cop introduïdes les dades i clicant el botó “Save”, es guardarà el nou jugador.



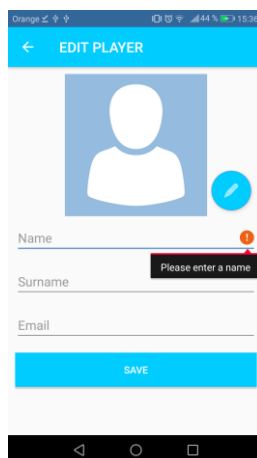
Imatge 56: Formulari per afegir nou jugador

Podem comprovar que s'ha afegit el nou jugador a la base de dades.

```
7
{
  "coachEmail": "coach@coach.com",
  "displayName": "Test Test",
  "email": "test@test.com",
  "name": "Test",
  "surname": "Test",
  "type": "Player",
  "urlPhoto": "https://firebasestorage.googleapis.com/v0/b/fee"
}
```

Imatge 57: Creació de nou jugador a Firebase Database

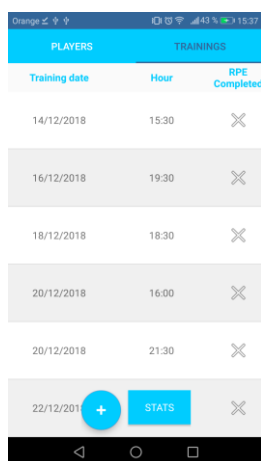
Si alguna de les dades del formulari no s'introdueix, el sistema mostrarà un missatge d'error.



Imatge 58: Formulari per afegir jugador amb missatge d'error

5.2.6 Veure entrenaments – CU6

Una vegada un usuari de tipus entrenador ha iniciat sessió, se li mostrarà la seva pantalla principal, que conté dues pestanyes, una amb el seu llistat de jugadors i una altra amb el seu llistat d'entrenaments. Clicant a la pestanya d'entrenaments, se li mostraran els seus enregistrats.



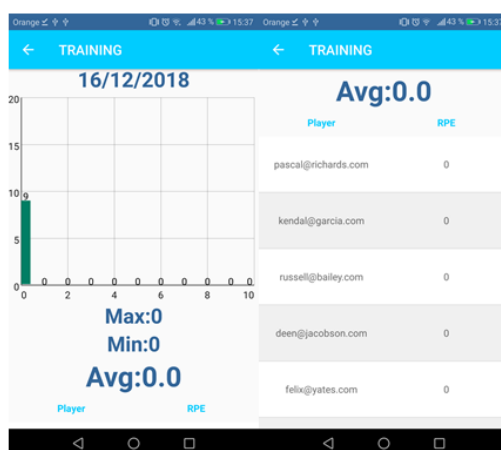
Training date	Hour	RPE Completed
14/12/2018	15:30	✕
16/12/2018	19:30	✕
18/12/2018	18:30	✕
20/12/2018	16:00	✕
20/12/2018	21:30	✕
22/12/2018		+

Imatge 59: Llistat d'entrenaments d'un entrenador

5.2.7 Veure dades d'entrenament – CU7

Un usuari de tipus entrenador que ha iniciat sessió, a través de la seva pantalla principal on hi ha un llistat dels seus entrenaments, pot visualitzar les dades d'aquell entrenament clicant qualsevol dels elements de la llista.

Una vegada clicat, es mostrarà una pantalla amb les dades de l'entrenament.

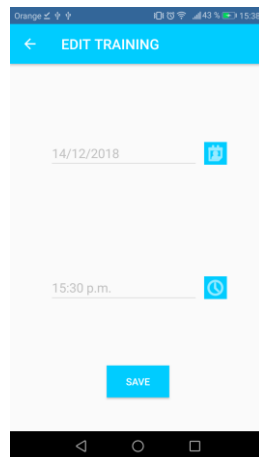


Imatge 60: Pantalla amb les dades d'un entrenament

5.2.8 Afegir entrenament – CU8

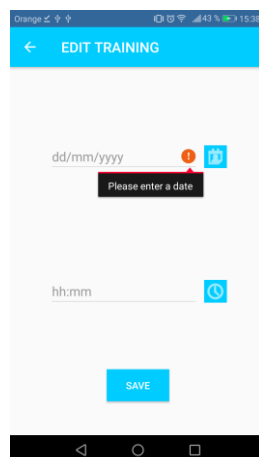
Un usuari de tipus entrenador que ha iniciat sessió, a la seva pantalla principal on apareixen els seus entrenaments, pot afegir un nou entrenament clicant el botó "+".

Una vegada clicat, apareixerà un formulari amb les dades necessàries per a afegir un nou entrenament. Un cop introduïdes les dades i clicant el botó “Save”, es guardarà el nou entrenament.



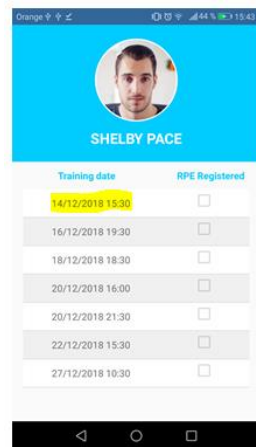
Imatge 61: Pantalla per afegir nou entrenament

Si alguna de les dades introduïdes no es correcta, es mostrarà un missatge a l'usuari.



Imatge 62: Pantalla per introduir nou entrenament amb missatge d'error

També podem comprovar que al afegir un entrenament, aquest també s'afegeix a la llista d'entrenaments disponibles per als jugadors d'aquell entrenador. Podem comprovar això iniciant sessió amb algun dels jugadors i comprovant que l'entrenament apareix al seu perfil.

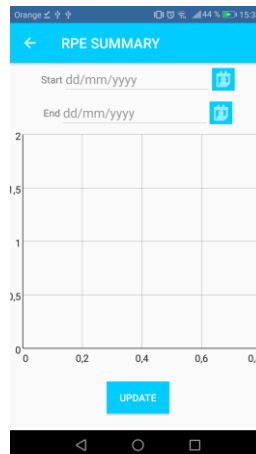


Imatge 63: Llistat d'entrenaments del jugador amb el nou entrenament afegit

5.2.9 Veure estadístiques globals d'entrenament – CU9

Un usuari de tipus entrenador que ha iniciat sessió, a través de la seva pantalla principal on hi ha un llistat dels seus entrenaments, pot visualitzar les dades globals dels seus entrenaments, clicant el botó “STATS”.

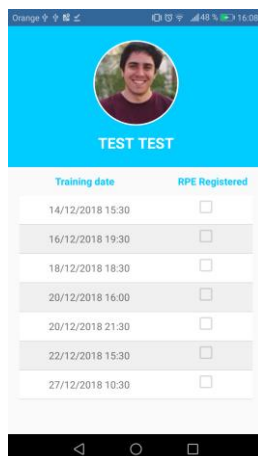
Una vegada clicat, es mostrarà una pantalla que permetrà seleccionar un rang de dates, i un botó “Update” per a actualitzar les dades del gràfic segons les dates seleccionades.



Imatge 64: Pantalla de resum global d'estadístiques d'entrenaments

5.2.10 Veure entrenaments – CU10

Una vegada un usuari de tipus jugador ha iniciat sessió, se li mostrarà la seva pantalla principal, que conté una llista dels seus entrenaments.



Imatge 65: Pantalla principal del jugador, amb el seu llistat d'entrenaments

5.2.11 Registrar RPE en un entrenament – CU11

Una vegada un usuari de tipus jugador ha iniciat sessió, se li mostrarà la seva pantalla principal, que conté una llista dels seus entrenaments. Si cliquem sobre un entrenament que no tingui registrat l'RPE, es mostrarà un formulari per a registrar-lo.

Si en aquest formulari seleccionem un valor i cliquem el botó "Save", es registrarà aquell RPE en aquell entrenament.



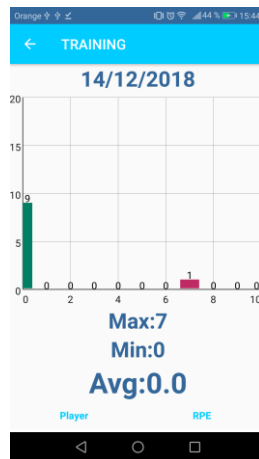
Imatge 66: Formulari per registrar RPE per part d'un jugador

Podem comprovar que s'ha registrat correctament l'RPE comprovant el valor a la base de dades.



Imatge 67: Registre de RPE a la base de dades

També podem comprovar que aquell valor és visible per al seu entrenador, visualitzant les dades d'aquell entrenament concret en la pantalla de l'entrenador.



Imatge 68: Pantalla de detalls d'entrenament després de registrar un RPE

6. Conclusions

Una vegada realitzat tot el projecte, incloent l'anàlisi de resultats, cal avaluar en general tot el desenvolupament del projecte, centrant-nos sobretot en si s'han assolit els objectius inicials que es van plantejar i quins aspectes es poden millorar o quins aspectes queden pendents per a futurs treballs.

6.1 Assoliment d'objectius

Primer de tot, recordem que l'objectiu principal del projecte era la implementació d'una aplicació Android per a la gestió i monitorització de la càrrega física d'un equip professional de bàsquet.

Després de la realització de l'anàlisi, disseny, implementació i valoració de l'aplicació, explicats en els apartats anteriors d'aquesta memòria, podem concloure que s'ha assolit l'objectiu principal plantejat.

Els casos d'ús que es van definir, permetien arribar a l'objectiu, i tal com hem comprovat a l'apartat de la valoració, s'han realitzat un conjunt de proves que demostren que tots aquests casos d'ús es compleixen.

6.2 Seguiment de la planificació

En general, podem dir que la planificació inicial s'ha seguit tal com estava plantejat, a excepció d'alguna petita desviació en alguna de les tasques concretes, però que s'ha acabat compensant amb altres tasques que al final han consumit menys hores que les plantejades.

Un dels aspectes claus per al seguiment de la planificació, ha estat les entregues de les diferents PECS, que obligaven a portar la feina al dia i al tenir uns objectius molt concrets en cada una, obligaven a seguir la planificació per a poder entregar cada part a temps.

Per altra banda, sobre els possibles obstacles que ens podíem trobar durant la realització del projecte, exposats en l'apartat de planificació, podem dir que no s'han produït i que per tant, no ha calgut cap reajustament dels temps planificats inicialment.

6.3 Aspectes a millorar

Tal com s'ha comentat a l'apartat de la implementació, un dels aspectes més millorables de l'aplicació desenvolupada és la gestió de les dades. Degut al poc temps per a realitzar la implementació, el que es va fer va ser optar per una aplicació funcional, més que buscar el màxim rendiment. Així doncs, caldria

millorar l'accés i la gestió de les dades, que es fa mitjançant Firebase Database, per tal de millorar el rendiment.

En quant a la part funcional de l'aplicació, un dels aspectes a millorar seria la part del jugador. Ara mateix, s'ofereix al jugador una pantalla principal amb tots els seus entrenaments disponibles per a registrar el RPE. Tot i així, no seria necessari mostrar sempre tots els entrenaments, ja que només té sentit registrar l'RPE després de realitzar l'entrenament, no a entrenaments futurs. En aquest sentit, es podria millorar aquesta pantalla del jugador, mostrant només aquella informació dels entrenaments dels quals pot registrar l'RPE o ressaltant quins són els pròxims entrenaments.

En aquest sentit, i tenint en compte els subobjectius que es van plantejar inicialment, podem veure que hi ha un d'aquests subobjectius que al final, per falta de temps, es va descartar ja que no impactava en l'assoliment de l'objectiu principal. Aquest subobjectiu era:

- Possibilitat d'introduir informació sobre lesions passades per part dels entrenadors, o molèsties per part dels jugadors.

Per tant, es podria ampliar la funcionalitat del jugador per a que es pogués introduir aquesta informació per després ser visualitzada per part de l'entrenador.

Per últim, també es podrien afegir algun tipus de notificacions automàtiques, com per exemple, que aparegués una notificació al jugador de quan pot registrar un determinat RPE en un entrenament. Aquesta funcionalitat recordaria al jugador que ha de registrar un determinat RPE al acabar un entrenament i permetria agilitzar tot el procés de recollida de dades.

6.4 Treball futur

L'aplicació desenvolupada actualment, seria una aplicació 100% funcional, tot i així, hi haurien diferents vies per a afegir funcionalitats noves, a part dels aspectes a millorar comentats en l'apartat anterior.

Tal com s'explica en el primer capítol de la memòria, l'RPE no és l'únic indicador utilitzat per al control de càrregues. Així doncs, es podrien afegir diferents formularis més per tal que el jugador els omplís per tal de ser analitzats per l'entrenador.

Si volguéssim introduir aquests nous formularis, caldria replantejar la funcionalitat actual del jugador, i redissenyar la seva pantalla principal per tal de poder incloure aquestes noves funcionalitats.

7. Glossari

- **RPE:** Rating of Perceived Exertion. És una mesura subjectiva de la sensació d'esforç realitzada per part d'un esportista després d'una sessió d'entrenament. Es mesura segons una escala, anomenada escala de Borg, que compren valors del 1 al 10.

8. Bibliografía

[1] Gabbett T], The training—injury prevention paradox: should athletes be training smarter and harder?, Br J Sports Med [2016;50:273-280].

<<https://bjsm.bmj.com/content/50/5/273>>

[2] entrenamientodeportivo1blog.wordpress.com, *La Carga: Tipos de Carga y Componentes* – [07/11/2016]

<<https://entrenamientodeportivo1blog.wordpress.com/2016/11/07/la-carga-tipos-de-carga-y-componentes/>>

[3] simplifaster.com, *The Use of RPE in Team Sports* – [30/01/2017]

<<https://simplifaster.com/articles/use-rpe-team-sports/>>

[4] diegocalvo.es, *Metodología SCRUM (Metodología ágil)* – [07/04/2018]:

<<http://www.diegocalvo.es/metodologia-scrum-metodologia-agil/>>

[5] volere.co.uk, *Requirements Specification Template* – [2017]:

<<http://www.volere.co.uk/template.htm>>

[5] medium.com, *Android y el patrón MVC* – [24/03/2017]:

<<https://medium.com/@fahedhermoza/notas-del-libro-programaci%C3%B3n-con-android-the-big-nerd-ranch-cap-2-46d9071d9e7e>>

[6] wikipedia.org, *Modelo-Vista-Controlador* – [25/10/2018]:

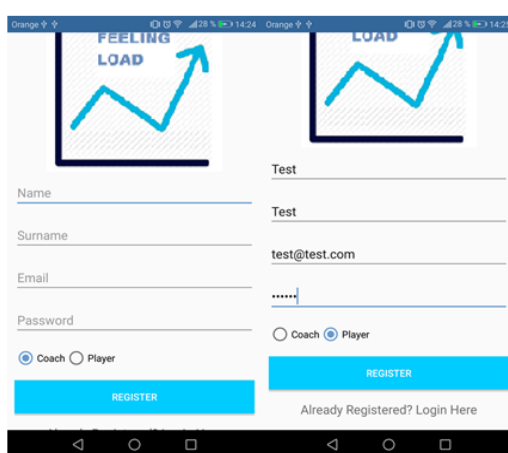
<<https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>>

9. Annexos

9.1 Manual d'usuari

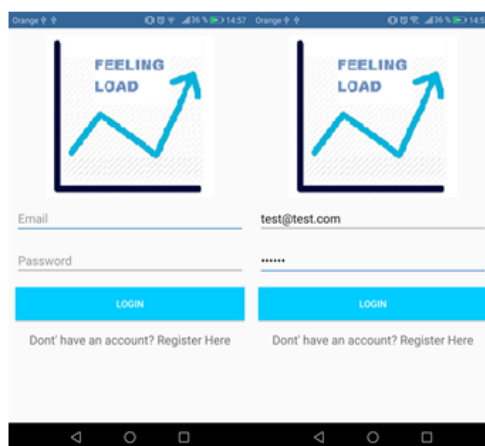
Registre i login

La primera pantalla que ens apareixerà serà la de login/registre. Aquesta pantalla, ens permetrà iniciar sessió si ja tenim un usuari registrar, o registrar un nou usuari. Per a registrar un nou usuari, només cal omplir les dades del formulari (totes són requerides). Si ja existeix un usuari amb aquell email o alguna de les dades no és correcte, es mostrarà un missatge d'error.



Imatge 69: Pantalla de registre

En la pantalla de login, l'aplicació ens mostra el formulari de inici de sessió, amb les dades requerides, que són el correu i la contrasenya. Una vegada omplert el formulari amb les dades d'un usuari existent, i clicant el botó "Login", el sistema inicia la sessió i ens mostra la pantalla principal de l'usuari.

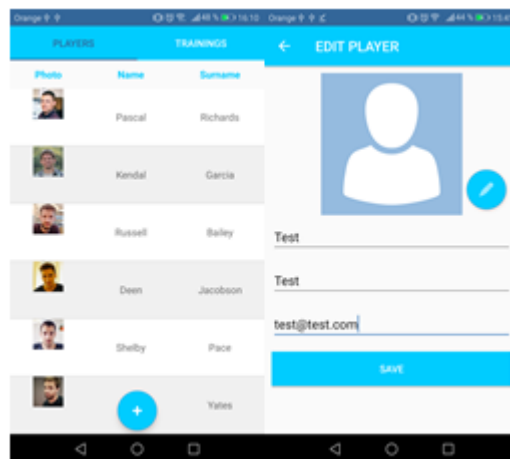


Imatge 70: Pantalla de login

Pantalla principal d'entrenador

Si iniciem sessió amb un usuari entrenador, ens mostrarà la seva pantalla principal, que conté dues pestanyes, una per veure un llistat dels seus jugadors, i una per veure el llistat dels seus entrenaments.

Des de la pestanya de jugadors, es pot clicar qualsevol dels elements de la llista per a poder visualitzar les seves dades i/o editar-les. Tanmateix, també hi ha un botó “+” que ens permet afegir un nou jugador. Tan si cliquem un jugador existent com si cliquem el botó per afegir un nou jugador, es mostrarà el mateix formulari. La diferència és que si venim d’un jugador ja existent, les dades es mostraran omplertes.



Imatge 71: Pantalla principal amb llistat de jugadors i formulari per afegir nou jugador

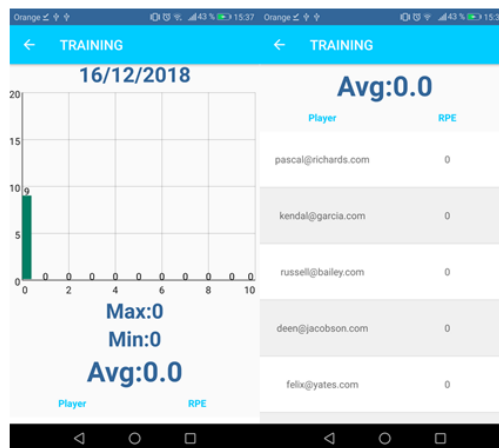
En el formulari per afegir un nou jugador, si alguna de les dades no és correcta, es mostrarà un missatge d’error. Tampoc podrem afegir dos jugadors amb el mateix correu. Una vegada guardat el nou jugador, apareixerà a la llista de jugadors disponibles.

Des de la pestanya d’entrenaments, el primer que veurem és un llistat dels entrenaments ja creats. Dels entrenaments, es mostrarà el dia, la hora i una imatge que ens indica si tots els jugadors han registrar l’RPE per a aquell entrenador o no.

Training date	Hour	RPE Completed
14/12/2018	15:30	X
16/12/2018	19:30	X
18/12/2018	18:30	X
20/12/2018	16:00	X
20/12/2018	21:30	X
22/12/2018		X

Imatge 72: Llistat d'entrenaments d'un entrenador

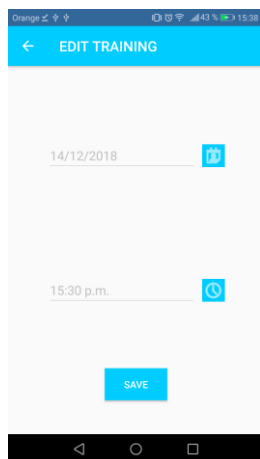
Si cliquem qualsevol dels entrenaments de la llista, tinguin o no complert el registre de RPE, se'ns mostrarà una pantalla amb les dades recollides per a aquell entrenament. El que ens apareixerà és una gràfica amb els RPEs registrats, i una taula amb la correspondència de jugadors i el seu RPE.



Imatge 73: Pantalla amb les dades d'un entrenament

Des de el llistat d'entrenaments, disposem de dos botons, un botó "+" que ens permet afegir un nou entrenament, i un botó "Stats", que ens permet visualitzar dades acumulades de diversos entrenaments.

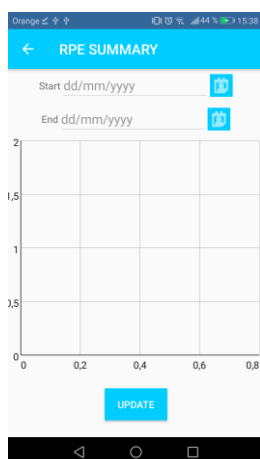
Si cliquem sobre el botó "+", ens apareixerà un formulari per a afegir un nou entrenament, on podem escollir el dia i la hora.



Imatge 74: Pantalla per afegir nou entrenament

Si alguna de les dades introduïdes no es correcta, es mostrarà un missatge a l'usuari.

Si cliquem sobre el botó “Stats”, ens apareixerà una pantalla on podem escollir un rang de dates per visualitzar les dades de tots els entrenaments compresos entre aquelles dates. L'únic que cal és escollir les dues dates i clicar el botó “Update”.

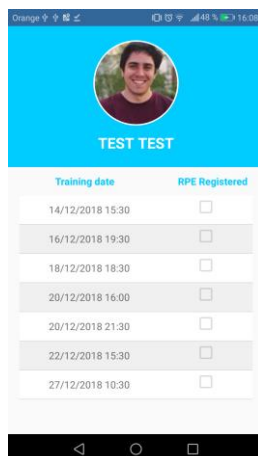


Imatge 75: Pantalla de resum global d'estadístiques d'entrenaments

Pantalla principal del jugador

Si iniciem sessió amb un usuari de tipus jugador, la seva pantalla principal ens mostrarà una llista dels entrenaments que tingui associats. Per a que apareguin entrenaments en un jugador, ha d'existir un usuari de tipus entrenador que tingui afegit aquell jugador a la seva llista. A partir de llavors, tots els entrenaments que afegeixi l'entrenador, apareixeran a la pantalla principal del jugador.

De la mateixa manera, quan el jugador registri l'RPE en algun dels entrenaments, aquesta dada es podrà visualitzar en la sessió de l'entrenador.



Imatge 76: Pantalla principal del jugador, amb el seu llistat d'entrenaments

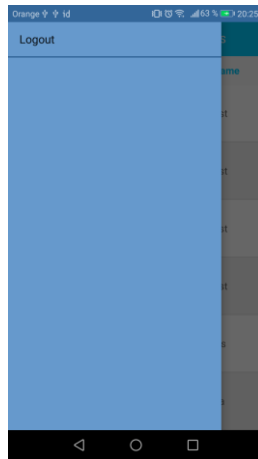
La llista d'entrenaments ens mostra quins entrenaments tenen ja registrat l'RPE i quins no. Si cliquem sobre un entrenament que ja tingui registrat l'RPE, simplement es mostrarà el valor registrat, no es podrà modificar. Si cliquem sobre un entrenament sense RPE registrat, es mostrarà un formulari per a poder escollir l'RPE.



Imatge 77: Formulari per registrar RPE per part d'un jugador

Tancament de sessió

Tant a la pantalla principal del jugador com a la de l'entrenador, podem tancar la sessió. Per a fer-ho, l'únic que cal és desplegar el menú lateral, que conté la opció de tancar la sessió.



Imatge 78: Menú lateral amb opció de tancament de sessió