

Informatika Fakultatea

Informatika Ingeniaritzako Gradua

▪ Gradu Amaierako Lana ▪
Software Ingeniaritza

Blokeetara orientatutako eskalada aktibitatean
lagunduko duen mugikorrerako aplikazioa

Ander López Collantes

Zuzendaria: Jon Iturrioz Sanchez

2019 - Ekaina

Laburpena

Blokeetara[1] orientatutako eskalataileei zuzendutako aplikazio bat garatzea izan da Gradu Amaierako Lan honen helburua. Aplikazio honek bi fazeta garrantzitsu ditu.

- Sare sozialaren fazeta, non erabiltzaileak normalean sare sozial batek dituen aspektu eta funtzionalitateak erabili ahal izango ditu.
- Eskaladaren fazeta, non erabiltzaileak eskaladarekin zerikusia duten ezaugarriak eskainiko dituen funtzionalitateak eta ezaugarriak erabili ahal izango ditu.

Gaur egun sare sozialeko funtzionalitateak dagoeneko asmatuak daude eta ez da ezer berria, aldiz eskaladako fazetan zentratuta ideia nagusia erabiltzaileak pixkana aplikazioak eskaini dezakeena osatzea izango da.

Oraingo eskaladako aplikazioetan, administratzaileak dira eskalatzeko lekuak sortzen eta kokatzen dituztenak, aplikazio honetan ordea hau erabiltzaileen esku egongo da eta horrela aplikazioaren aberastasuna modu erakargarri batean haziko da.

Gaien Aurkibidea

Laburpena.....	iii
Gaien Aurkibidea.....	v
Irudi eta taulen aurkibidea.....	viii
Irudiak.....	viii
Taulak	ix
1. Sarrera eta motibazioa.....	1
2. Merkatuaren analisia	3
2.1 Mountain Project	5
2.2 Climbing Away.....	6
2.3 Vertical-Life.....	6
2.4 Climb Around	7
3. Kudeaketa.....	9
3.1. Irismena	10
3.1.1. Proiektuaren irismena.....	10
3.1.2. Emangarrien identifikazio eta ezaugarriak	11
3.1.3. LDE diagrama	11
3.1.3.1 LDE diagramaren Atazak eta Ezaugarriak.....	12
3.2. Kudeaketa plana.....	14
3.2.1 Garapen plana	14
3.2.2 Komunikazio plana (interesatuak).....	15
3.2.3 Arriskuak kudeatzeko plana	15
3.2.4 Kalitate plana	16
3.3. Plangintza.....	16
3.3.1 Mugarriak	16
3.3.2 Atazak eta estimazioak.....	18
3.3.3 Plangintzaren Gantt diagrama.....	19
3.4. Jarraipen eta Kontrola	20
3.4.1 Erabilitako komunikazioak.....	20
3.4.2 Proiektuan gertatuko arazoak.....	20
3.4.3 Proiektuan lortutako kalitatea.....	21
3.4.4 Benetako mugarriak.....	21

3.4.5	Egindako atazak	22
3.3.3	Gantt diagrama	23
4.	Eskakizunen bilketa	25
4.1.	Erabilpen kasuak	26
4.1.1	Aktoreak	26
4.1.2	Aktoreak eta erabilpen kasuak	26
4.2.	Gertaera fluxuak.....	28
4.2.	Domeinu eredua	46
5.	Diseinua	49
5.1.	Arkitektura	50
5.1.1	Eredua	50
5.1.2	Bista.....	50
5.1.3	Kontrolatzailea.....	51
5.2.	Datuen egitura	51
5.3.	Kautoketarako datuak	54
5.4.	Sekuentzia diagramak	54
6.	Erabilitako teknologiak.....	57
6.1.	Produktuaren garapen ingurunea	58
6.2.	Teknologia eta Lengoaien azalpenak	58
6.2.1	Android Studio	58
6.2.2	Java.....	59
6.2.3	XML.....	59
6.2.4	Firebase	60
6.2.5	JSON	61
6.2.6	StarUML.....	62
7.	Inplementazioa	63
7.1.	Bista	64
7.2.	Kontrolatzailea	66
7.2.1	Kontrolatzailearen implementazio nagusia	66
7.2.2	Kontrolatzailea eta Firebase.....	68
7.2.2.1	Firebase Authentication	68
7.2.2.2	Firebase Realtime Database	69
7.2.2.3	Firebase Storage.....	71
7.3.	Eredua.....	73
8.	Ondorioak.....	75

8.1. Proiektuaren garapena	76
9. Bibliografia	79
A. Eranskina: Bileren aktak	81
B. Eranskina: Bi proba garrantzitsuenak	91

Irudi eta taulen aurkibidea

Irudiak

2.1 irudia	Mountain Project	5
2.2 irudia	Climbing Away	6
2.3 irudia	Vertical-Life	6
2.4 irudia	Climb Around.....	7
3.1 irudia	LDE diagrama	12
3.2 irudia	Plangintzaren Gantt Diagrama	19
3.3 irudia	Gantt Diagrama.....	23
4.1 irudia	Erabilpen kasuen eredua	27
4.2 irudia	LogIn gertaera fluxua	28
4.3 irudia	Erregistratu gertaera fluxua.....	29
4.4 irudia	Pasahitza ahaztu gertaera fluxua	29
4.5 irudia	LogOut gertaera fluxua	30
4.6 irudia	Profila ikusi gertaera fluxua	30
4.7 irudia	Profila editatu gertaera fluxua	31
4.8 irudia	Erabiltzaileak bilatu gertaera fluxua	31
4.9 irudia	Laguna ezabatu gertaera fluxua.....	32
4.10 irudia	Lagun eskerak gertaera fluxua	32
4.11 irudia	Publikazioak gertaera fluxua	33
4.12 irudia	Publikazioa editatu/ezbatu gertaera fluxua	33
4.13 irudia	Lagunen publikazioak ikusi gertaera fluxua.....	34
4.14 irudia	Publikazioen iritzia gertaera fluxua	35
4.15 irudia	Mapa kontsultatu gertaera fluxua.....	35
4.16 irudia	Blokea sortu gertaera fluxua	36
4.17 irudia	Blokea sortu gertaera fluxua	36
4.18 irudia	Blokearen informazioa ikusi gertaera fluxua	37
4.19 irudia	Gustuko blokeak gehitu gertaera fluxua.....	37
4.20 irudia	Gustuko blokeak bistaratu gertaera fluxua	38
4.21 irudia	Blokea burutu/ez burutu gertaera fluxua.....	39
4.22 irudia	Blokea bilatu gertaera fluxua	40
4.24 irudia	Erronka automatikoki sortu gertaera fluxua	41
4.25 irudia	Erronkan parte hartu gertaera fluxua	41
4.26 irudia	Estatistikak ikusi gertaera fluxua	42
4.27 irudia	Erronkako blokeak bete gertaera fluxua	42
4.28 irudia	Rankinga ikusi gertaera fluxua.....	43
4.29 irudia	Konbertsazioa sortu gertaera fluxua	43
4.30 irudia	Konbertsazioa sortu gertaera fluxua	44
4.31 irudia	Mezuak gehitu gertaera fluxua	44
4.32 irudia	Erabiltzaileak ezabatu gertaera fluxua	45
4.33 irudia	Domeinu eredua	46
5.1 irudia	Proiektuaren arkitektura	50
5.2 irudia	Datu basearen egitura Firebase plataforman	52
5.3 irudia	Firestore Storage karpeten egitura	53

5.4 irudia	Firestore Storage fitxategien egitura.....	53
5.5 irudia	Firestore erabiltzaileen kautotze egitura	54
5.6 irudia	SignIn sekuentzia diagrama	55
5.7 irudia	Erronkak modu adimentsuan sortu sekuentzia diagrama	55
5.8 irudia	Blokea sortu sekuentzia diagrama.....	56
6. 1 irudia	Java klaseak	59
6. 1 irudia	XML kodea	60
6.2 irudia	Datu-basea.....	62
7.1 irudia	RelativeLayout inplementazioa.....	64
7.2 irudia	Liburutegian inplementazioa proiektuan.....	65
7.3 irudia	Bista baten inplementazioa	65
7.4 irudia	Bistako elementuetan irudia ezarri	66
7.5 irudia	Bistako elementuko testuen hizkuntza.....	66
7.6 irudia	Kontrolatzailearen hasierako inplementazioa	67
7.7 irudia	Bistatik bistara mugitzeko metodoa	68
7.8 irudia	Firestore zerbitzuen liburutegiak	68
7.9 irudia	Erabiltzaile baten sorrera	68
7.10 irudia	Datu basearekiko erreferentzia.....	69
7.11 irudia	Datu basearekiko erreferentzia.....	69
7.12 irudia	Datu basea sortutako datu berriekin	70
7.13 irudia	Datu basetik datu bat irakurri	70
7.14 irudia	Query bat sortu	70
7.15 irudia	Erabiltzaileen datuak lortzen.....	71
7.16 irudia	Datu baseko datuen eguneraketa	71
7.17 irudia	Datu baseko datuen ezabaketa.....	71
7.18 irudia	Firestore Storage instantzia	71
7.19 irudia	Firestore Storage fitxategiak igo	72
7.20 irudia	Firestore Storage fitxategiak lortu	72
7.21 irudia	Firestore Storage fitxategiak ezabatu	73
7.22 irudia	Ereduaren inplementazioa Erabiltzaileen datuak lortzeko	73

Taulak

2.1.taula: Merkatuko aplikazioen analisi eta konparaketa.....	5
3.1.taula: Ekintzen mugarriak.....	17
3.2.taula: Atazen mugarriak	17
3.3.taula: Mugarri orokorrak.....	18
3.4.taula: Atazak eta estimazioak	19
3.5.taula: Betetako mugarriak.....	22
3.6.taula: Egindako atazak eta estimazioak.....	22

1

Sarrera eta motibazioa

Sarrera modura, proiektuaren testuingurua, honen aukeraketa, helburuez eta parte hartzaileei buruz hitz egingo da. Proiektu honen hasiera, Pablo Cuenca izeneko Ingeniaritza Mekanika-ko ikasle ohi baten proposamenean hasten da. Bakoitzak bere aldetik merkatuan dauden eskalatzeko aplikazioen azterketa bat egin ondoren, ondorio berdinerara iritsi ginen.

Aplikazio hauek eskalada munduan zentratuak zeudenez, batzuek horretara jotzen zuten soilik, erabiltzaileek konturik ez izatea baizik eta soilik eskalatzeko beharrezkoa den informazioa eskaintzea.

Beste aldetik aplikazio batzuk sare sozial eta eskalatzeko aplikazioen arteko nahasketa bat egiten saiatzen ziren arren erabiltzaile fazeta zertxobait eskasa geratzen zen, erabiltzaileak beste erabiltzaileekin eta batez ere eskalatzeko lekuekin egin zezaketen elkarrekintza mugatua baitdute.

Aplikazio hauek ordea gauza bat dute komunean, bertan azaltzen den informazioa gehien bat aplikazio sortzaileak edo administratzaileak osatzen dutela. Pablo-k berak esanda, normalean pertsona hauek faltan botatzen dute ezagutzen dituzten eskalatzeko zonak bisitatu eta gerora beraien aplikazioetara igotzeko aukera hori, horrela gainera beste eskalatzaileek aukera izango lukete leku berri hauek bisitatu eta probatzeko. Leku asko daude munduan zehar eta hauek guztiak pertsona gutxi batzuekin betetzea zaila denez, zergatik ez egin erabiltzaileek beraiek asebetetzea ezaugarri hauek?

Idea honekin buruan hasi zen dena, behin aplikazioa sortua dagoela baliteke zertxobait "hutsa" egotea baina etorkizunera begira oso aplikazio aberatsa atera daiteke hemendik eta gaur egun dauden eskalatzailerik prestatuak zein hasi berriak direnentzako oso interesgarria izan daiteke, azken finean erabiltzaileen arteko eta aplikazioaren arteko elkarrekintzan oinarritzen da.

Dokumentu hau, aipatu berri den proiektuaren memoria izango da, Ander López Collantes izeneko Software Ingeniaritza espezialitateko ikasleak egina. Lan hau guztiz beregain egin da, Jon Iturrioz irakaslea tutore bezala eta Pablo Cuenca proiektuaren ideia printzipala eduki duenarekin batera.

Memoria hau zortzi atalez osatua egongo da. Lehenengo kapituluak proiektuaren garapenarekin hasi baino lehen egin diren kudeaketei eta analisisiei buruz hitz egingo da. Gainontzekoek garatutako aplikazioaren informazio teknikoagoa izango dute oinarri moduan.

2

Merkatuaren analisia

Proiektuarekin hasteko merkatuaren azterketa bat egin da. Analisi hau burutzeko gaur egun Google-ko Play Store merkatu digitalean dauden eskaladara zuzendutako bost aplikazio garrantzitsuenak aztertu dira. Bakoitzak dituen ezaugarri zein funtzionalitateak kontuan hartuta beraien artean eta proiektu honetarako sortu den aplikazioaren artean konparaketa bat egin da, beti ere merkatuko beste aplikazioak ahulago diren ezaugarri eta funtzionalitateetan zentratuz.

Hurrengo tauletan ikusten da aplikazio bakoitzean aztertu diren ezaugarri garrantzitsuenen arteko konparazioa aplikazioen artean:

Hurrengo taulan ikusten da aplikazio bakoitzean aztertu diren ezaugarri garrantzitsuenen arteko konparazio:

	Mountain Project	Climbing Away	Vertical-Life	Climb Around	Aprieta (Nire App)
Albisteak	X		X		
Blokeak GEO lokalizazioa bidezko bilaketa	X	X	X		X
Blokeak izenen bidezko bilaketa	X	X	X		X
Herrialdeka dauden zonak (baloratuak)	X				
Blokeari buruzko informazioa + argazkiak+ bideoak + iritziak	X	X (Iritziak eta bideoak ez)	X (Iritziak eta bideoak ez)	X	X
Profila	X		X		X
Erabiltzaileen bilaketa			X		X
Foroa	X				
Ranking			X		X
Erronka propioak sortu edo besteetan parte hartu			X(Parte hartu soilik)		X
Erronkak sortu GEO lokalizazioa eta zailtasunaren arabera					X
Lagunak gehitzeko aukera			X		X
Erabiltzaileen artean chat-a					X
Gustuko blokeak gordetzeko aukera		X	X		X

Blokeak baloratu	X		X		X
Mapa bidezko bilaketa	X	X	X	X	X
Maparen bidez blokea gehitzeko aukera					X
Publikazioak gehitu			X(Argazkiak soilik)		X
Blokeko bideoak baloratu					X
Erronkak baloratu					X
Gida liburuak		X			

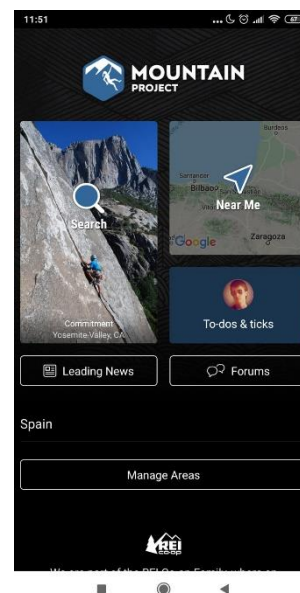
. 2.1.taula: Merkatuko aplikazioen analisi eta konparaketa

Behin goiko taularen bidez beste aplikazioen arteko konparaketa ikusita, orain aplikazio bakoitzaren analisia egingo da banan-banan, berain ezaugarri nagusiak eta faltan botatzen diren funtzionalitateei buruz hitz egiten.

2.1 Mountain Project

Mountain Project-ek dituen ezaugarri garrantzitsuenak bere albiste eta foroak dira batik bat erabiltzaile arteko elkarrekintza sustatzen dutenak, bestalde mundu guztiko herrialdeetan ditu banatuak eskalatzeko zonak eta bertan herrialdeak kalifikatu eta blokeen zailtasun media ateratzen dute. Hau asko bidaiatzen duten eskalatzailentzat oso egokia dela iruditzen zaigu.

Aplikazio honen ezaugarri batzuk faltan bota dira, alde batetik erabiltzaileek saioa irekitzeko aukera duten arren ondoren ez dute beraien artean elkar ekintza asko eskatzen duten funtzionalitateirik, gehienbat iritziak eman, foroetan parte hartu, argazkiak igo eta balorazioak burutzea.



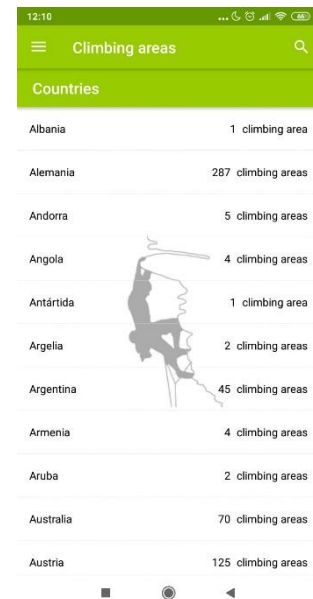
. 2.1 irudia Mountain Project

Azkenik, eskalatzeko zona gutxi ikus daitezke, oso urruti joan gabe, Euskal Herri osoan ez dago eskalatzeko zonarik erregistratua aplikazioan, honen arrazoietakoa bat aplikazioaren administratzaileak soilik gehitzen dituztelako da eta beraz lan zaila mundu guztiko zona guztiak lortzeko.

2.2 Climbing Away

Aplikazio hau da lauetatik funtzionalitate gutxien eskaintzen dituen. Nahiz eta hau horrela izan mapa bidezko bilaketa egiten ari garenean filtro oso interesgarriak ditu beste aplikazioek ez dutena. Filtro hauen bidez eskalatzeko bloke edo paretaren zailtasuna eta ze motako eskalada egin nahi den adierazita ezaugarri horiek betetzen dituzten eskalatzeko zonak bilatzen ditu.

Ezaugarri hau alde batera utzita, faltan botatzen da erabiltzaile fazeta. Aplikazio honen bidez ezin da erabiltzaile batekin saio hasi eta beraz ezin dira eskalatzeko zona berriak gehitu, argazkiak igo, iritziak eman eta horrelako beste hainbat funtzionalitate.



Countries	Number of climbing areas
Albania	1 climbing area
Alemania	287 climbing areas
Andorra	5 climbing areas
Angola	4 climbing areas
Antártida	1 climbing area
Argelia	2 climbing areas
Argentina	45 climbing areas
Armenia	4 climbing areas
Aruba	2 climbing areas
Australia	70 climbing areas
Austria	125 climbing areas

. 2.2 irudia Climbing Away

Eskalatzeko ezaugarriari begira hauek mapa bidez, bilatzaile bidez edo herrialdeka bila daitezke. Kasu honetan aldiz ezin izango dira baloratu, argazkiak igo edo iritziak eman eta gainera herrialdeak eskalatzeko zonak sailkatzeko bakarrik balioko dute, ez dute Mountain Project-en antzera herrialdeak baloratzen edo eta bertako eskalatzeko zonen mediak erakusten.

Aipatzekoa da azken eguneraketa batean tresna interesgarri bat jarri dutela. Tresna hauek liburu gidak eta eskalatzeko zailtasunen gradu mailak erakusten dituzten funtzionalitateak dira. Hauen bidez eskalatzaileak informazio gehiago edukiko du.

2.3 Vertical-Life

Aztertu diren aplikazio guztietatik gehien eskaintzen duen aplikazioa da hau. Hasteko bi eskalatzeko zona eskaintzen ditu: naturarekin bat kalean eta bestea esparru itxi batean (eskalada-murruetan erdaraz *rocódromo*).

Garatu nahi dugun aplikazioak erronkak sortu eta hauetan parte hartzea nahi dugu, hau da, erabiltzaileek erronkak sortu eta beste erabiltzaileek erronka hauetan parte hartzea. Aplikazio honetan erronkak administratzaileek sortzen dituzte eta gero erabiltzaileek parte hartzen dute nahi izan ezker.



. 2.3 irudia Vertical-Life

Gure aplikazioaren antzera, aplikazio honek ere Ranking global bat du, aplikazioko erabiltzaile guztiek osatzen dutena.

Eskalatzeko zonen bilaketa gure aplikazioaren antzera egiten du baina kasu honetan ere ez du zona gehiago gehitzea eskaintzen, hau administratzaileen esku dago. Gainera bloke edo eskalatzeko zona bat egiteko lehenik eta behin ordaindu egin beharko da gero aplikazioan eginga bezala agertzea nahi bada. Nahiz eta gustuko blokeetara gehitu daitekeen ordaindu gabe, beste funtzionalitate guztiak ordaindu beharrekoak dira: baloratu eta blokea egin.

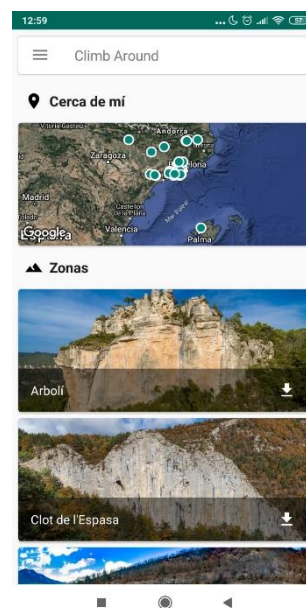
Aplikazio honek erabiltzaile aldetik, gure aplikazioan egin nahi dugunaren antzekoa du; lagunak gehitzea eta bakoitzaren estatistikak ikustea baina bestalde ez du erabiltzaileen arteko mezuak elkar trukatzeko aukera eskaintzen eta gainera publikazioak igotzerakoan argazkiak soilik eskaintzen ditu.

Azkenik beste aspektu interesgarri bat ere aipatu behar da. “Entrenamiento” atal bat dauka. Bertan egin diren eskalada zonak erregistratzeaz gain zein egunetan egin diren ere gordetzen da.

2.4 Climb Around

Azken aplikazio honek ere Climbing Away aplikazioaren antzera ez du erabiltzailearen fazeta bat erakusten. Ez du erabiltzaile baten saio irekitzeko funtzionalitaterik eta beraz honek eskaintzen dituen beste ezaugarri guztiak ere ez: publikazioak igo, iritziak eman, balorazioak egin, ... hau da aplikazioko informazioa editatu edo gehitzek aukerak ez dira egongo.

Bestalde ezaugarri oso interesgarriak ditu. Eskalatzeko zona batean eskalatzeko paretak edo harriak banatzen ditu eta bakoitzarekiko oso informazio baliagarriak eskaintzen ditu: hauen altuera, zailtasuna, zonaren egoera,... . Gainera zonen zailtasunen media ere erakusten du.



. 2.4 irudia Climb Around

3

Kudeaketa

Memoriako kapitulu honetan proiektua era egoki batean aurrera eramateko garatu den planifikazioaz hitz egingo da. Kapitulu honek hiru puntu garrantzitsu edukiko ditu:

1. **Irismena**, proiektuaren irismena, emangarriak eta LDE diagrama.
2. **Kudeaketa plana**, garapenaren plana, komunikazioa plana (interesatuak), arriskuak kudeatzeko plana eta kalitatearen plana.
3. **Plangintza**, mugarriak, atazak eta estimazioak eta Gantt diagrama.
4. **Jarraipen eta Kontrola**, plangintza ataleko atazak proiektua garatzen joan den heinean.

3.1. Irismena

Kapitulu honen sarreran aipatu bezala, irismena atalak hiru puntu bereiziko ditu: Proiektuaren irismena, emangarrien identifikazioa eta LDE diagrama izango dira.

3.1.1. Proiektuaren irismena

Proiektu honen helburua eskalada mundura zuzenduta dagoen eta erabiltzaileak hau osatzeko ahalmena edukiko duen aplikazio bat garatzea izango da.

Egin behar dugun aplikazioaren arloan dabiltzan beste merkatuko aplikazioak begiratu ostean erabiltzailea eta aplikazioaren arteko interakzio maila handituko duen aplikazio baten bila gabiltza. Gaur egungo aplikazio hauek azken finean eguneraketan bidez informazio gehiago gehitzen doaz. Gure ustetan honek aplikazioa era batean mugatu egiten du, azken finean pertsona talde bat ari da aplikazioko informazio guztia betetzen. Kasu honetan informazioa aipatzean, eskalada egiteko gune egokiei buruz ari gara.

Aplikazio honek eskalatzailerak eskalatzeko gune bat ezagutzen badute edo eta leku berri bat aurkitzen badute, zona hori aplikaziora gehitzea izango da helburu nagusietako bat. Aipatu bezala mundu guztian zehar erabili behar den aplikazio bat izango balitz, zona hauek guztiak aplikazioan erregistratzea lan oso gogorra izan daiteke pertsona talde txiki baten aurrean, baina hau egiten dutenak erabiltzaileak beraiek badira, modu oso errez eta interaktibo batean lan egin daitekeela pentsatzen dugu.

Beste helburu nagusietako bat GEO lokalizazioaren erabilpena eta erronkak izango dira. Erabiltzailea eta aplikazioaren arteko interakzio hori baino gehiago bilatzen ari gara, erabiltzaileek beraien arteko interakzioa alegia. Horretarako erronka sistema bat sortzea pentsatzen dugu eta honekin batera ranking bat, horrela erabiltzaileek erronkak burutu ahala beraien erabiltzaile maila handituz joango da eta ranking-ean lehenengo postua lortzea izango da beraien helburua.

Proiektu honen aplikazioa hutsetik egingo da eta hasteko *Android Studio*[2] erabiliz eta beraz Java lengoaia erabiltzea pentsatzen da. Erabiliko den datu basea erlazionala edo ez erlazionala izango den pentsatu beharra dago, bakoitzak bere aldetik eman dezaketen onurak eta desabantailak aztertu ondoren erabakiko da hau hala ere Android-en egitea pentsatua dugunez posibilitate handia izango du *Firestore*[3] zerbitzua erabiltzea eta beraz “gure zerbitzaria” Google-en egongo da eta ez da beharrezkoa izango inplementatzea.

Aplikazioaren puntu garrantzitsuak garatzen doazen heinean, funtzionalitate hauek gure lagunen artean probatzea pentsatzen da. Horrela zer hobetuko luketen edo agertu daitezkeen arazoak erakustea lortuko genuke. Behin produktua bere etapa finalera iristen ari denean eta funtzionalitate nagusienak garatuak dituenean eskalatzailerak artean banatzea pentsatzen da, beraien iritzia jakite arren.

Produktua hasiera batean hiru hizkuntzatan egitea pentsatzen da; Euskaraz, Erdaraz eta Ingelesean. Aplikazioaren irismenaren arabera hizkuntza gehiago gehitzea pentsatuko da.

3.1.2. Emangarrien identifikazio eta ezaugarriak

Proiektua burutzeko hiru emangarri egongo dira:

- **Produktua**
Proiektu honek produktu bakarra edukiko du, zehazki Android-eko aplikazio bat. Aplikazio hau ekainaren 23-rako bukatuta egon behar du eta plataforman igota. Aldiz produktua egiteko eskatutako pertsonak (Pablo Cuenca) ez du emangarri honen data zehatz bat eskatu.
- **Memoria**
Proiektuaren kudeaketa eta garapena nola burutu diren azaltzen duen dokumentua ekainaren 23 produktuarekin batera bukatuta eta ADDI plataforman entregatua egon behar da.
- **Defentsa prestatzeko materiala**
Proiektuaren defentsa burutzeko prestatu behar den materiala (presentaziorako materiala, demoak,...) uztailak 1-8 eginda egon beharko luke.

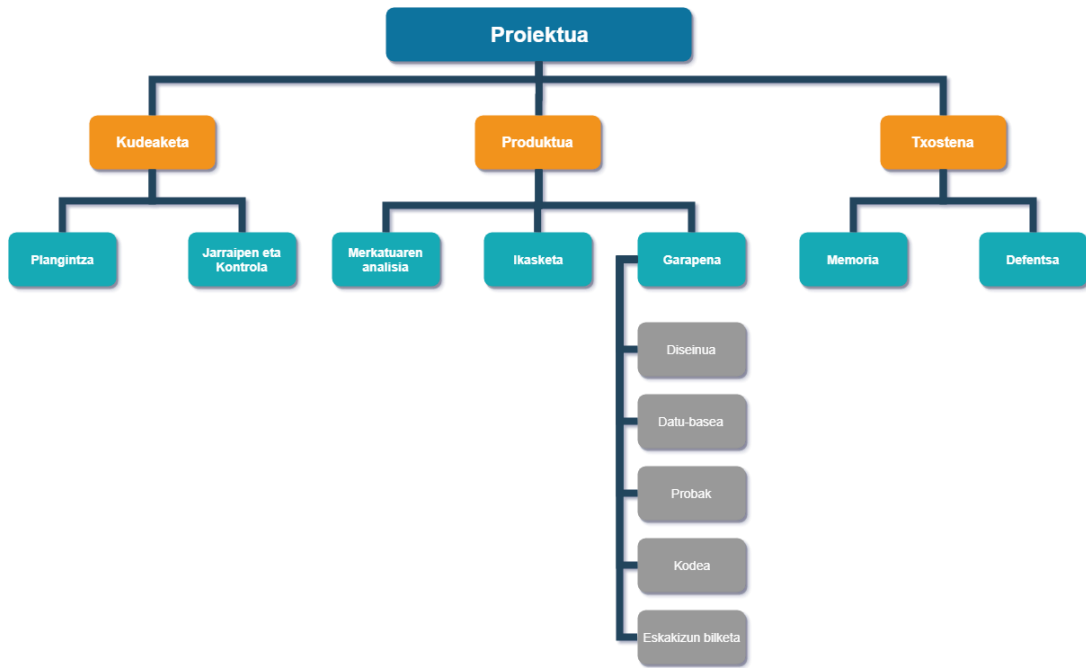
3.1.3. LDE diagrama

Proiektua aurrera emateko hiru atal nagusitan banatuko da. Hasteko proiektuaren kudeaketa edukiko dugu. Bertan proiektuaren plangintza eta jarraipen eta kontrola mantenduko dira, egin beharreko lanen denbora desbiderapenak kontrolatzeko.

Hurrengo atala produktua bera izango da. Honen garapenarekin hasi baino lehen merkatuan dauden antzeko aplikazioen analisi bat egin beharko da, hauen funtzionalitateak aztertuz. Aplikazioa garatzeko lehenik eta behin *Android Studio*-ren erabilera, zein aplikazioa garatzeko erabili beharko diren beste zenbait softwareen erabilera ere ikasi beharko dira azkenik proiektuak garapen on bat eduki dezan.

Bukatzeko, proiektuaren memoria idatzi eta defentsarako aurkezpen bat prestatu beharko da.

Ondorengo irudiak ikus daiteke proiektuaren lanaren deskonposaketaren egitura bat:



. 3.1 irudia LDE diagrama

3.1.3.1 LDE diagramaren Atazak eta Ezaugarriak

Lanaren Deskonposaketa Egituraren diagramatik, proiektua burutzeko egin beharreko azpiatzen lista bat erakutsiko da hemen:

- Kudeaketa
 - Plangintza
 - Irismena
 - LDE diagrama
 - Gantt diagrama
 - Mugarriak
 - Garapen plana
 - Komunikazio plana
 - Atazak eta estimazioak
 - Emangarriak identifikatu
 - Kalitate plana
 - Arriskuen plana
 - Jarraipen eta Kontrola:
 - Desbiderapenen kalkulua
- Produktua
 - Merkatuaren analisisa
 - Ikasketa
 - Android Studio en erabilera

- Firebase-en erabilera
- Garapena
 - Diseinua
 - Datu-basearen aukeraketa eta sorkuntza
 - Aplikazioaren probak
 - Kodea
 - Eskakizun bilketa
- Txostena
 - Memoria idatzia
 - Defentsarako aurkezpena

3.2. Kudeaketa plana

Atal honetan proiektuaren kudeaketa planari buruzko ezaugarriak azalduko dira.

3.2.1 Garapen plana

Proiektuaren garapena gauzatzeko jarraitu diren pausuak eta zein plan jarraitu behar dugun adieraziko da hemen.

Hasiera batean urtarrila amaiera eta otsailaren lehen astean hasiko da proiektua. Lehenengo pausua proiektuaren ideia duen pertsonarekin bilera bat gauzatzea izango da. Bertan proiektuaren nondik norakoak zehaztuko dira. Alde batetik proiektuaren parte izango den mugikorrek aplikazioak, garrantzitsuenak edo ezinbestekoak izango diren funtzionalitateak zein diren zehaztu beharko dira.

Behin ezaugarri garrantzitsuenak finkatuak daudela bestelako ezaugarri eta funtzionalitateak zehaztuko dira, beti ere ikasleak proiektuko memoria egiteko denbora ere kontuan hartuta.

Hasiera batean proiektuko aplikazioa hasita zegoen *Ionic* mugikorrek aplikazioak sortzeko tresnarekin. Software honekin bai *Android* zein *IOS* erako aplikazio bat sortu daiteke baina *Android Studio*-rekin alderatuta interneten aurkitu daitekeen informazioa ez da hain aberatsa. Honez gain bi sistema eragileentzako aplikazioa sortzeak denbora gehiago eskatzen du eta plangintzak segurtasun maila bat mantentzeko *Android*-erako egitea soilik erabaki zen. Beraz aplikazioa hasieratik egitea izango da ideia.

Android Studio-rekin programatzea eta teknologia berrien erabilerak ikasteko lehenik eta behin aplikazioak edukiko dituen funtzionalitate ez garrantzitsuekin hastea erabaki da. Epe honetan ez da tutorearekin bilerarik egingo, behin tresna hauek erabiltzen ikasita daudenean eta geratzen den denbora kontuan edukita tutorearekin eta proiektuaren ideia izandako pertsonarekin bilera bat gauzatuko da.

Bilera honetan funtzionalitate garrantzitsuak zatika banatuko dira eta egiten goazen heinean bilera gehiago gauzatuko dira tutorearekin. Garapenaren erritmoaren arabera bilera hauetan zein funtzionalitate egin eta zein alde batera utzi erabakiko da.

Garapenarekin aurrera goazen heinean aplikazioarekin proba batzuk egingo dira. Horretarako pertsona ezberdinei aplikazioa instalatzeko eta probatzeko eskatuko zaie eta noski arazoren bat egon den edo hobekuntzarik egingo luketen ere galdetuko zaie.

Proiektua ekainaren 23rako amaitua eta plataforman igota egon behar duenez, entregatzeko epea baino bi aste edo hiru aste lehenago aplikazioak amaitua egon beharko du, gerora arazoren bat edo gertatzen bada zuzenketak egiteko denbora eduki dezagun.

3.2.2 Komunikazio plana (interesatuak)

Proiektuan parte hartuko duten interesatuak hurrengoak dira:

Ikaslea	Ander Lopez	alopez312@ikasle.ehu.eus
Zuzendaria	Jon Iturrioz	jon.iturrioz@ehu.es
Proiektuaren jabea	Pablo Cuenca	pablo.cuenca.petit@gmail.com
Bezero posibleak		

Zuzendariaren posta elektronikoa bilerak gauzatzeko erabiliko da batik bat, nahiz eta galdera txikiren bat egin behar den momentuan ere erabiliko da. Bileraren eguna aukeratzeaz gain ordurarte egindako lana bidaltzeko ere erabiliko da, zuzendariak begirada bat eman diezaion eta horrela bileran ze aldaketak egitea gomendagarria den hitz egin dezakegun.

Pablarekin hitz egiteko hasiera batean WhatsApp bidez egingo da nahiz eta honekin arazorik balego bere posta elektronikoa erabiliko zen. Lehenengo bilera egin ondoren Ekuador-era doanez Skype bidezko bilerak ere egiteko erabiliko da posta elektronikoa hau.

Bezero posibleak produktuaren garapenean ikaslearen eta Pablaren ezagunak izango dira. Beraiek aplikazioa garatzen doan heinean probei buruzko informazioa pasako dizkigute.

3.2.3 Arriskuak kudeatzeko plana

Atal honetan proiektuaren garapenean zehar aurkitu daitezkeen arriskuei buruz eta hauek nola ekidin daitezkeen hitz egingo da.

- **Egindako lana galtzea:** Egunero egiten diren aurrerapenak, bai memoria aldetik bai kode aldetik, ordenagailu pertsonalean kopia bat edukitzeaz gain Google Drive-en eta kanpoko memoria baten ere gordeko dira. Arazo honi aurre egiteko modu bakarra da, hauetako pausoak jarraitu ezean gero ezin izango baizen egindako ezer errekuaratu eta honek oso arazo larriak sortuko lituzke proiektuan.
- **Denbora arazoak:** Memoria eta proiektuko aplikazioa ekainaren 23-rako bukatua eta plataforman igota egon behar du, baina irailean beste aukera bat edukita proiektuan edukiko lukeen eragina ez da hain handia, nahiz eta komenigarria den ekainerako bukatzea.
- **Funtzionalitate batean trabatuta geratzea:** Orain arte ez da mugikorrerako aplikaziorik egin eta beraz funtzionalitatearen bat ez ateratzea eta denbora asko bertan pasatzea gerta daiteke. Beraz horrelako egoera batean aurkitzen bagara, aukerarik onena beste funtzionalitate bat egiten hastea da. Gainera horrela baliteke garatzen goazen heinean egin gabe geratu den funtzionalitatea egiteko ideia bat edukitzea

- **Helburuak ez lortzea:** Hasieran adierazitako helburuetako bat lortzea funtzionalitate batekin tratatuta geratzearen arazoarekin erlazioa dauka eta beraz arrisku honek eragin handia izango du eta arriskua gertatzeko probabilitatea helburuaren zailtasunaren arabera izango da. Kasu honetan ere helburu osoa lortzeko proiektuan zehar dauden helburu txikiagoak egitea izango da, gerora helburu nagusiagoak egiteko probabilitateak handiagoak izango baitira helburu txikiekin ikasitakoa erabiliz.

3.2.4 Kalitate plana

Proiektu osoaren kalitatea bermatzeko bi zatiren kalitatea minimoa bermatu beharko da. Hasteko **memoriaren** kalitate bermatu behar dugu.

Memoriaren kalitatea bermatzeko, lehenik eta behin ehu-ko web orrialdean dagoen memoria egiteko txantiloia jarraitu behar dugu. Bertan hau egiteko jarraitu beharreko pauso batzuk aurkituko ditugu eta honen kalitate minimoa lortzeko hauek jarraitzea komeni da.

Proiektuaren zati honen kalitatea bermatzen jarraitzeko komenigarria izango da proiektuan egiten diren pauso garrantzitsu guztiak egiten ditugun heinean memoria ere zabalik edukitzea pauso hauek bertan idazteko, batik bat plangintzarekin bat gatozen ikusteko.

Beste zatiari dagokionez, hau da, proiektuko **produktuari** dagokionez, bere kalitate minimoa bermatzeko, helburu bezala finkatu ditugun funtzioak garatu beharko dira. Funtzio hauek garatu ostean denbora baldin badago produktuaren kalitatea hobetze arren funtzionalitate gehigarri batzuk egin daitezke.

3.3. Plangintza

Kudeaketaren zati honetan plangintzaren hiru atal nagusi ikusiko ditugu: mugarriak, atazak eta estimazioak, eta Gantt diagrama.

3.3.1 Mugarriak

Proiektuaren garapenean zehar hainbat mugarri jarri dira. Mugarri hauek proiektuan zehar dauden puntu garrantzitsuak izango dira eta beraz data hauetara iristen garen heinean gure egoera aztertu eta erabakiak hartu beharko dira honen arabera.

Hurrengo taulak ikus daitezke aipatu berri diren mugarrien azalpena:

EKINTZA	DATA
Proiektuaren hasiera	Urtarrilak 28
Proiektuaren inflexio puntua	Martxoak 15
Memoriaren hasiera	Apirilak 15

Produktuaren amaiera	Maiatzak 18
Memoriaren amaiera	Ekainak 5
Aurkezpena prestatzearen hasiera	Ekainak 10
Proiektua ADDI plataformara igo	Ekainak 20
Aurkezpena prestatzen bukatu	Ekainak 28
Defentsa burutu	Uztaila 1-12

. 3.1.taula: Ekintzen mugarrak

Mugarrietan ikus daitekeen bezala Martxoaren 15ean inflexio puntu batekin aurkitzen gara. Inflexio puntu honetan proiektuan zehar egindakoa baloratuko da eta bertako zein puntutan aurkitzen garen aztertuko da, hau amaitzeko geratzen den denbora ere kontuan hartuta. Hau ikusteko zuzendariarekin eta Pablo-rekin bilera bat gauzatuko da eta bertan erabakiko da plangintzan adierazitako ataza guztiak beteko diren edo denboraz ondo ez bagabiltza zein ezaugarri alde batera uztearen erabakia.

Defentsa burutzeko datak Uztailaren 1-12 tartea aipatu da mugarrietan, ez baitdakigu oraindik zein datetan esleituko zaigun defentsaren aurkezpena.

Ekintzen mugarriaz gain, produktuaren garapenak berak dituen beste hainbat atazen mugarrak azalduko dira hurrengo taulan:

ATAZA	DATA
Merkatuaren analisia	Urtarrilak 30
Ikasketa	Maiatzak 18
Garapena	Maiatzak 18
Probak	Otsailak 15 – Maiatzak 18
Diseinua	Otsailak 15
Datu-basea	Otsailak 15 – Maiatzak 18
Eskakizunen bilketa	Otsailak 10

. 3.2.taula: Atazen mugarrak

Ikus daiteke mugarri hauetan ataza batzuen data ez dela finkoa, baina hasteko *Ikasketa* atazaren aipamen bat egin behar da. Ikasketa proiektuaren garapenean zehar emango den gauza bat da. Nahiz eta hasiera batean ikasketa hau proiektuaren hasieran egingo den, honen garapenean zehar erabili behar den tresnen ikasketa gauzatuko da eta beraz gure ustetan ikasketa hau produktuaren amaiera arte gauzatuko da.

Bestalde aipatu bezala bi ataza ditugu data finkorik gabe. Bi kasuetan, bai *Probak* eta bai *Datu-basean* kontuan hartu behar da hauen amaiera ez dela finkoa izango probak beti ere data oien artean gauzatuko direlako amaiera finko bat eduki beharrean eta datu-basearen diseinua aldakorra izan daitekeela produktua garatzen doan heinean.

Amaitzeko proiektu osoko mugarrak taula batean elkartuko dira dataren arabera ordenaturik:

EKINTZA / ATAZA	DATA
Proiektuaren hasiera	Urtarrilak 28
Merkatuaren analisia	Urtarrilak 30
Eskakizunen bilketa	Otsailak 10
Diseinua	Otsailak 15
Proiektuaren inflexio puntua	Martxoak 15
Memoriaren hasiera	Apirilak 15
Probak	Otsailak 15 – Maiatzak 18
Datu-basea	Otsailak 15 – Maiatzak 18
Ikasketa	Maiatzak 18
Garapena	Maiatzak 18
Produktuaren amaiera	Maiatzak 18
Memoriaren amaiera	Ekainak 5
Aurkezpena prestatzearen hasiera	Ekainak 10
Proiektua ADDI plataformara igo	Ekainak 20
Aurkezpena prestatzen bukatu	Ekainak 28
Defentsa burutu	Uztaila 1-12

. 3.3.taula: Mugari orokorrak

3.3.2 Atazak eta estimazioak

Irismen atalean azaldutako LDE diagraman oinarrituta lan-paketeetan antolatutako lanaren denbora-estimazioa ikus daiteke hurrengo taulan. Proiektua egiteko guztira 345 bat ordu estimatzen dira. Lan-pakete hauek aztertutako denbora gehien pasatukoa *Produktua* atalean izan dela argi dago azken finean tresna berrien ikasketa eta proiektuko produktuaren garapenak osatzen dute lan-pakete hau batik bat. Bestalde denbora gutxien eskatu duen lan-paketea *Kudeaketarena* izan da.

ATAZAK	ORDUAK
Kudeaketa	25
Plangintza	15
Jarraipen eta Kontrola	10
Produktua	240
Merkatuaren analisia	65
Ikasketa	45
Garapena	130

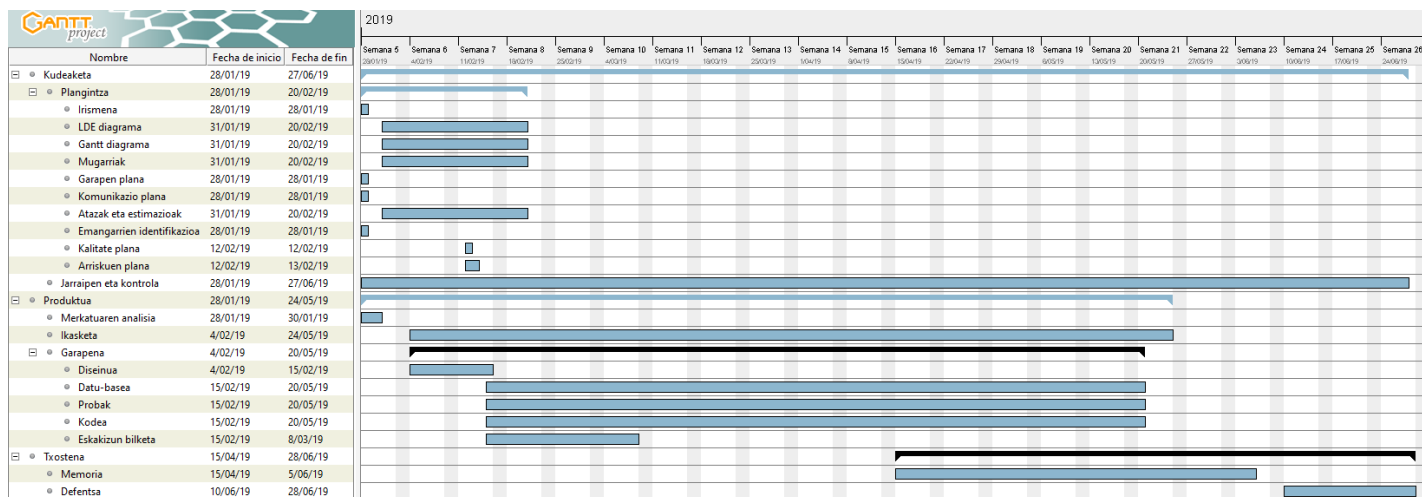
Diseinua	10
Datu-basea	35
Probak	10
Kodea	55
Eskakizun bilketa	20
Txostena	80
Memoria	65
Defentsa	15
GUZTIRA	345

. 3.4.taula: Atazak eta estimazioak

Taula honetan ikusitako guztia ordu estimazio batzuk besterik ez dira, gerora baliteke estimazio hauek ez betetzea eta jarritako ordu kopuruak aldaketa batzuk edukitzea. Normalean arazoren baten ondorioz ordu aldaketa nabariak agertu daitezke eta beraz hasieran estimatutako orduak baino gehiago behar izango ditugu.

3.3.3 Plangintzaren Gantt diagrama

LDE diagraman aipatutako lan-paketeak eta bertako atazen azpi-atazak hurrengo irudian agertzen den Gantt diagraman jarri dira.



. 3.2 irudia Plangintzaren Gantt Diagrama

3.4. Jarraipen eta Kontrola

Kudeaketaren atal honetan, plangintzan egindako estimazioak eta planeatutako atazen jarraipen eta kontrol azalduko dugu. Plangintza atalean ikusitako puntu guztiak berrikusiko dira, hauek nola bete diren ikusteko asmotan.

3.4.1 Erabilitako komunikazioak

Planeatu den bezala, posta elektronikoa tutorearekin bilerak egiteko erabili da batik bat. Lehenik eta behin egindakoaren txosten bat bidali zaio begirada bat emateko. Hau egin ondoren tutorearekin hitz egiteko bilera bat adostu eta hau gauzatu dugu bere bulegoan, hasieran bidalitakoa nola hobeto, ze aldatu,... eta momentuko egoera ebaluatzen gauzei buruz hitz egiteko.

Posta elektronikotik egindako galderak soilik GrAL-eko gestiorako izan dira; nire tutore bezala esleitzeko eta defentsa eskatzeko.

Bestalde Pablo-rekin hitz egiteko aurreikusi bezala WhatsApp-a izan da tresnarik erabiliena, gehienbat aplikazioa garatzen zioan heinean hau pasa eta probatu zezan, gerora ze ezaugarri hobetu edo ageriko akatsei buruz hitz egiteko. Nahiz eta hau horrela izan, bilera batzuk ere egin ditugu Skype bidez, berak pentsatuak zeuzkan ideiak argitzeko eta nik ere ideia horiekin adibide batzuk emanez adostasun batera iristeko.

3.4.2 Proiektuan gertatuko arazoak

Atal honetan proiektuaren garapenean zehar aurkitu daitezkeen arriskuei buruz eta hauek nola ekidin daitezkeen hitz egingo da.

- **Egindako lana galtzea:** Esan bezala, egin den guztiaren kopia bana gordetzen joan gara, Google Drive-en, ordenagailuan bertan eta kanpo memoria batean.

Proiektuaren garapenean zehar ez da inolako galerarik izan baina *Android Studio*-n kode zatiak aldatzean ez bazaigu atera den emaitza gustatzen eta beraz hasieran geneukan kodea lortu nahi badugu ezin izango dugu hau egin plataforma honek eskaintzen duen "Deshacer" aukerarekin, mugatua baitdago. Hau ekiditeko, aldaketa handi bat egin behar zen bakoitzean kopia berri bat gordetzen zen berriz hasierako egoerara itzuli behar bazen.

- **Denbora arazoak:** Proiektuko aplikazio garatzerakoan zein memoriaren garapenean ez da inolako denbora arazorik egon, nahiz eta honen garapenean atal batzuk pentsatutakoa baino denbora gehiago eskatu, hasieratik pentsatua eta planeatua zeuden helburu guztiak lortu dira.
- **Funtzionalitate batean trabatuta geratzea:** Aplikazioaren garapenean hainbat arazoren aurrean topatu gara baina planeatu bezala funtzionalitate batekin trabatua ez geratzeko

beste batzuei eman zaie lehentasuna eta gerora atzean geratu zirenak egitea lortu dira. Azken finean honen garapenean zehar gauza berriak ikasten joan garez hasiera batean trabatuak geratzen ginen kode zatia konpontzea gauza “erraza” izan da.

3.4.3 Proiektuan lortutako kalitatea

Plangintzan kalitate planaren atalean aipatu bezala, proiektu honen kalitatea bi zatitan banatzen da, **memoriaren kalitatea** eta **produktuaren kalitatea**.

Memoria aldetik, ehu-ko web orrian zegoen GrAL-eko memoriaren txantiloietan adierazitako baldintza guztiak bete dira. Gainera atal bakoitzaren azalpena eta irakurlearentzat lagungarri izango diren adibide eta irudiekin memoria ondo azaldua eta irakur erraza dela iruditzen zaigu. Ezaugarri hauek kontuan hartuta memoriaren kalitatea bermatu dezakegu.

Produktu aldetik, hasieran ezarri ziren helburu guztiak bete dira eta gainera hau garatzen hari zen bitartean pentsatutako beste ezaugarri txiki batzuk ere garatu dira honen kalitatea hobetuz. Honez gain aplikazioaren ezaugarri bakoitza erabiltzaile askoren artean probatu dira egoera desberdinetan eduki zitzakeen errore guztiak konpondu ahal izateko. Beraz produktuaren kalitatea ere bermatu dezakegu aipatu berri dugunarekin.

3.4.4 Benetako mugarrak

Plangintzan estimatutako ekintzen eta atazen mugarrak banan-bana ikusi ziren atal horretan baina oraingoan zuzenean biak batera ikusiko ditugu ez baitdago aldaketa handirik. Hasteko proiektuko produktuaren garapenean hasiera ez zen oso erraza izan eta beraz eskatutako ezaugarrien garapena mantsoa izan zen.

Honen ondorioz proiektuaren inflexio puntua eta hasiera batean garapena amaitzeko mugarrak zertxobait atzeratu behar izan ziren, hurrengo taulan ikus daitekeen bezala. Baina honek ez du proiektuan zehar inpaktu handirik izan azken finean egindako plangintzan horrelako atzerapenak aurreikusitak zeuden.

EKINTZA / ATAZA	ESTIMATUTAKO DATAK	BENETAKO DATAK	DESBIDERAPENAK
Proiektuaren hasiera	Urtarrilak 28	Urtarrilak 28	-
Merkatuaren analisisa	Urtarrilak 30	Urtarrilak 30	-
Eskakizunen bilketa	Otsailak 10	Otsailak 10	-
Diseinua	Otsailak 15	Otsailak 15	-
Proiektuaren inflexio puntua	Martxoak 15	Apirilak 15	-31 egun
Memoriaren hasiera	Apirilak 15	Apirilak 15	-
Probak	Otsailak 15 – Maiatzak 18	Otsailak 15 – Maiatzak 28	-10 egun
Datu-basea	Otsailak 15 – Maiatzak 18	Otsailak 15 – Maiatzak 28	-10 egun
Ikasketa	Maiatzak 18	Maiatzak 28	-10 egun

Garapena	Maiatzak 18	Maiatzak 28	-10 egun
Produktuaren amaiera	Maiatzak 18	Maiatzak 28	-10 egun
Memoriaren amaiera	Ekainak 5	Ekainak 5	
Aurkezpena prestatzearen hasiera	Ekainak 10	Ekainak 10	
Proiektua ADDI plataformara igo	Ekainak 20	Ekainak 20	
Aurkezpena prestatzen bukatu	Ekainak 28	Ekainak 28	
Defentsa burutu	Uztaila 1-12	Uztaila 1-12	

. 3.5.taula: Betetako mugarrak

3.4.5 Egindako atazak

Proiektuaren plangintzan adierazitako atazen denbora estimazioak bete diren aztertuko dugu orain, egindako estimazioak, egindako benetako orduak eta hauen arteko desbiderapenak erabiliz egingo da hau hurrengo taulan ikus daitekeen bezala.

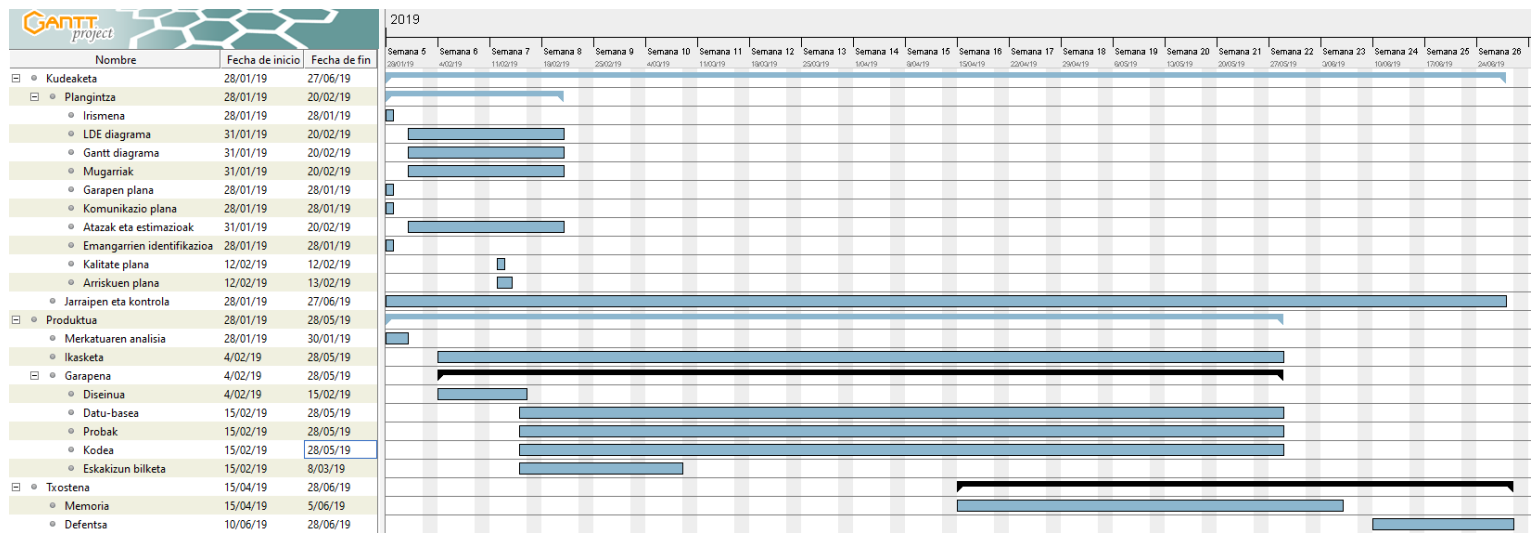
ATAZAK	ESTIMATUTAKO ORDUAK	BENETAKO ORDUAK	DESBIDERAPENAK
Kudeaketa	25		-
Plangintza	15		-
Jarraipen eta Kontrola	10		-
Produktua	240		-60
Merkatuaren analisia	65		-
Ikasketa	45	55	-20
Garapena	130	140	-40
Diseinua	10	10	-
Datu-basea	35	40	-5
Probak	10	10	-
Kodea	55	75	-35
Eskakizun bilketa	20	5	-
Txostena	80		-
Memoria	65	65	-
Defentsa	15	10	-
GUZTIRA	345	360	-15

. 3.6.taula: Egindako atazak eta estimazioak

Taulan ikus daitekeen antzera plangintzan aurre ikusitako 345 orduetatik ez gara oso urruti ibili baina azkenean garapenean gertatuko atzerapenak izan direla eta hamabost ordu gehiago erabili behar izan dira. Beraz guztira 360 orduko proiektu baten aurrean gaude.

3.3.3 Gantt diagrama

Mugarrietan egin diren aldaketak oso handiak izan ez diren arren plangintzan eraturtako Gantt diagramak aldaketa txiki batzuk eduki dituen hurrengo irudian ikusiko dira hauek:



. 3.3 irudia

Gantt Diagrama

4

Eskakizunen bilketa

Memoriako atal honetan aplikazioaren eskakizun bilketaren zatia aurkezten da. Bertan aplikazioaren Erabilpen Kasuak, Gertaera fluxua eta Domeinuaren Eredua azalduko dira. Gertaera fluxuaren aldetik aipatu beharrekoa da, gertaera bakoitzaren azalpena ulertzen lagunduko duen aplikazioaren argazki batzuk erabili direla .

4.1. Erabilpen kasuak

Erabilpen kasuak ondo ulertzeko lehenik eta behin bertan azaltzen diren aktoreen azalpen bat egingo da:

4.1.1 Aktoreak

Erabilpen kasu honetan hiru aktore mota agertuko dira; Administratzailea, Logeatutako erabiltzaileak eta erregistratu gabeko erabiltzaileak. Aktore mota bakoitzak bere erabilpen kasuan edukiko ditu nahiz eta aktore batzuk besteekin erabilpen batzuk partekatu.

Hemen ikusten dira aktore bakoitzaren azalpena eta beraien zeregina:

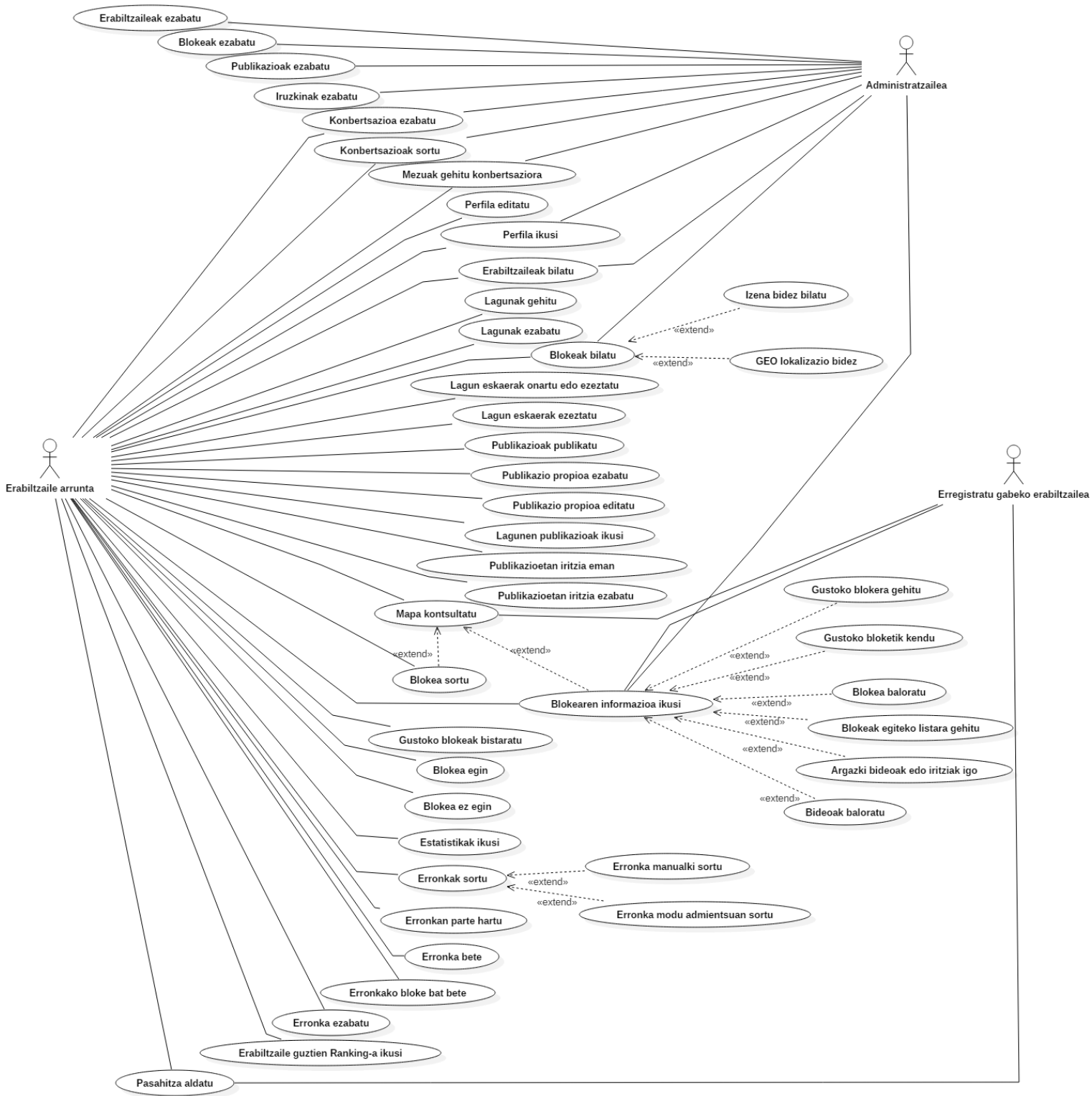
- Administratzailea: Ez da erabilpen kasu gehien dituen aktorea baina logeatutako erabiltzaile arrunt batek baino baimen handiagoak ditu, azken finean edozein erabiltzaile arruntek sortutako edozer ezabatzeko eta erabiltzaile arrunt bat ezabatzeko baimena ere badu.
- Erabiltzaile arrunta: Erabilpen kasu gehien dituen aktorea da, azken finean proiektuko produktua den aplikazioko funtzionalitate guztiak (administratzaileenak ezik) gauzatu ditzazke.
- Erregistratu gabeko erabiltzaile: Azkenik aktore hau aplikazioan erregistratua ez badago erregistratu eta logeatzeko aukera edukiko du. Gainera aplikazioan konturik ez badu edo logeatu gabe badago, aplikazioan aurkitzen diren blokeak bilatzeko aukera izango du, bai bilatzaile arrunta erabiliz, zein mapa kontsultatuz. Aldiz kontua badu baina saioa hasi gabe baldin badago, bere pasahitza aldatzeko aukera izango du.

4.1.2 Aktoreak eta erabilpen kasuak

Behin aktore bakoitza definitu ondoren, atal honetan erabilpen kasuen eredia ikusiko dugu. Hurrengo irudian ikus daitekeen moduan erabilpen kasu gehien dituen erabiltzaile arrunta izango da, azken finean aplikazioa aktore honi dedikatua dago eta beraz aplikazioak eskaintzen dituen funtzionalitate gehienak erabili ahal izango ditu.

Bestalde hiru aktoreen artean elkar konpartitzen dituzten erabilpen kasu batzuk ere aurkitzen ditu.

Beste aktore garrantzitsu bat administratzailea dela ikus daiteke, bere erabilpen kasu gehienak aplikazioko alde garrantzitsuenen gestiora zuzenduak daudela ikus daiteke beheko irudi honetan:



. 4.1 irudia Erabilpen kasuen eredua

4.2. Gertaera fluxuak

Atal honetan gertaera fluxuei buruz hitz egingo da eta fluxu bakoitza hobeto ulertzeko aplikazioaren interfazeen irudiez laguntzen saiatuko gara.

1. Gertaera fluxua: Login

Fluxu printzipala

-**Erabiltzailea:** Login egiteko datuak sartzen ditu (emaila eta pasahitza) eta sartzeko eskaera egin.

-**Sistema:** Sartutako datuak datu basean dauden erabiltzaileen emaila eta honi dagokion pasahitzarekin kontrastatu eta erabiltzailea aplikazioaren GUI printzipalera bideratzen du.

Fluxu alternatiboa: Erabiltzailea ez da existitzen

-**Sistema:** Sartutako emaila ez dago erregistratua errorea erakusten du.

Fluxu alternatiboa: Pasahitz desegokia

-**Sistema:** Sartutako pasahitza ez da egokia errorea erakusten du.

Fluxu alternatiboa: Bete beharreko atalak ez bete

-**Sistema:** Eskatutako datu guztiak bete behar den errore mezua erakusten du. Kasu honetan emaila eta pasahitza



. 4.2 irudia Login
gertaera fluxua

2. Gertaera fluxua: SignIn

Fluxu printzipala

-**Erabiltzailea:** SignIn egiteko beharreko datuak sartzen ditu; profileko argazkia, emaila, pasahitza, pasahitza errepikatua, erabiltzaile izena, pertsonaren izena eta herrialdea. Behin hau eginda erregistratzeko eskaera egiten du.

-**Sistema:** Sartutako datuak kontuan hartuta erabiltzailea datu basean erregistratzen du.

Fluxu alternatiboa: Erabiltzailea dagoeneko existitzen da

-**Sistema:** Sartutako emaila dagoeneko erregistratua dagoela adierazten du.

Fluxu alternatiboa: Bete beharreko atalak ez bete

-**Sistema:** Eskatutako datu guztiak bete behar den errore mezua erakusten du. Kasu honetan profileko irudia, emaila, pasahitza, pasahitz errepikatua, erabiltzaile izena, izena eta herrialdea izango dira.

Fluxu alternatiboa: Pasahitzak ez datoz bat

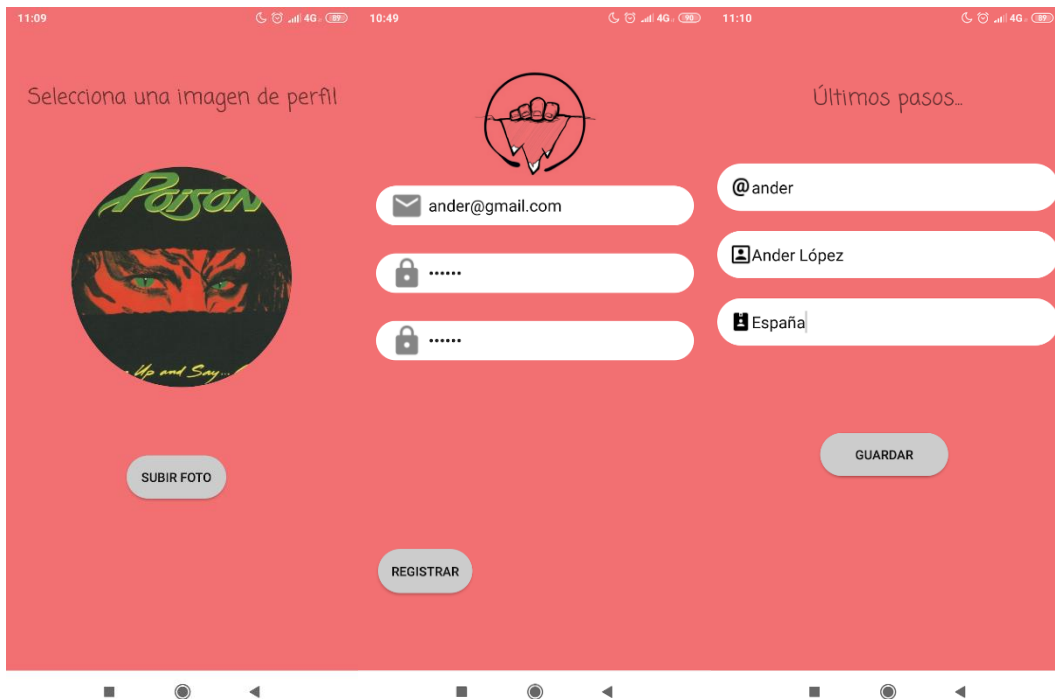
-**Sistema:** Pasahitzak berdinak ez diren errorea botatzen du.

Fluxu alternatiboa: Emailak ez du patroia betetzen

-**Sistema:** Sartutako emailaren formatua ez da zuzena errorea erakusten du.

Fluxu alternatiboa: Erabiltzaile izena dagoeneko existitzen da

-**Sistema:** Sartutako erabiltzaile izena datu basean existitzen dela errorea erakusten du.



.. 4.3 irudia

Erregistratu gertaera fluxua

3. Gertaera fluxua: Pasahitza aldatu

Fluxu printzipala

-**Erabiltzailea:** Email bat idatzi eta sistemari pasahitza aldatzeko eskaera bidaltzen dio.

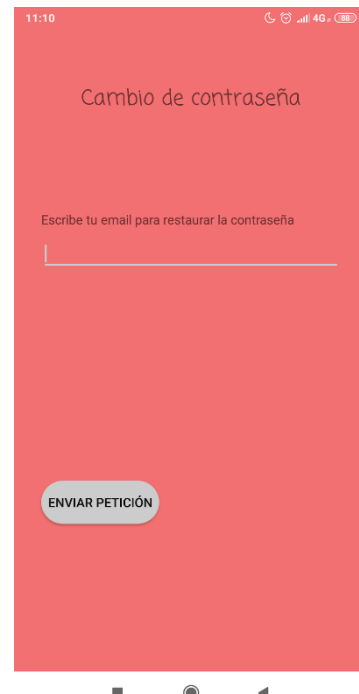
-**Sistema:** Sartutako emaila datu basean existitzen dela egiaztatu eta pasahitza aldatzeko email bat bidaltzen dio.

Fluxu alternatiboa: Emaila ez da existitzen

-**Sistema:** Sartutako emaila ez dago datu basean erregistratua errorea erakusten du.

Fluxu alternatiboa: Ez da ezer idatzi

-**Sistema:** Email bat idazteko abisua erakuste du.



.. 4.4 irudia

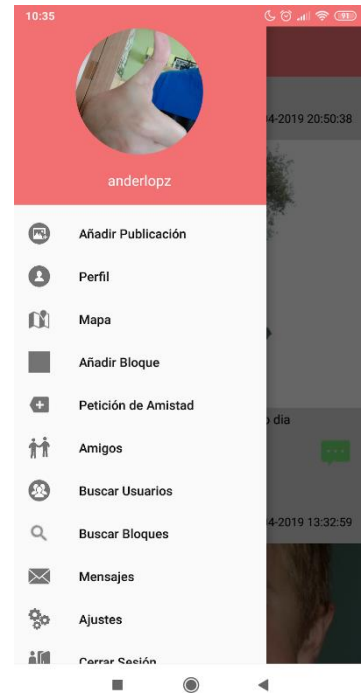
Pasahitza ahaztu gertaera fluxua

4. Gertaera fluxua: LogOut

Fluxu printzipala

-**Erabiltzailea:** LogOut edo saio ixteko eskaera egiten du.

-**Sistema:** Erabiltzailearen saio ixten du.



.. 4.5 irudia

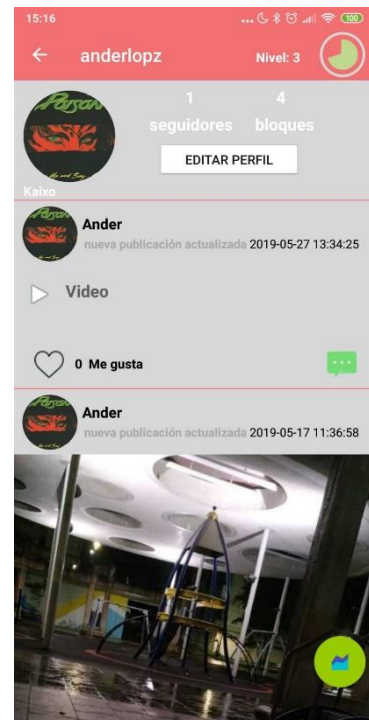
LogOut gertaera fluxua

5. Gertaera fluxua: Profila ikusi

Fluxu printzipala

-**Erabiltzailea:** Profil bat ikusteko eskaera egiten du honek bere erabiltzailearen identifikazio zenbakia eta ikusi nahi duen profilaren identifikazio zenbakia emanez.

-**Sistema:** Eskaera egin duen erabiltzailearen identifikazioa, ikusi nahi den profilaren erabiltzailearen identifikazio zenbakiarekin alderatu eta honen arabera profil propioa edo beste erabiltzaileen profila erakusten du.



.. 4.6 irudia

Profila ikusi

gertaera fluxua

6. Gertaera fluxua: Profila editatu

Fluxu printzipala

-**Erabiltzailea:** Datu pertsonalak aldatzeko GUI-ra joateko eskaera egiten du bere erabiltzaile identifikazio zenbakia emanaz.

-**Sistema:** Erabiltzailearen identifikazioa aztertu eta eskaera onartzen du. Behin bideratuta identifikazioarekin bat datorren erabiltzailearen datuak erakusten ditu.

-**Erabiltzailea:** Aldatu nahi diren datuak aldatu eta aldaketa gauzatzeko eskaera egiten du.

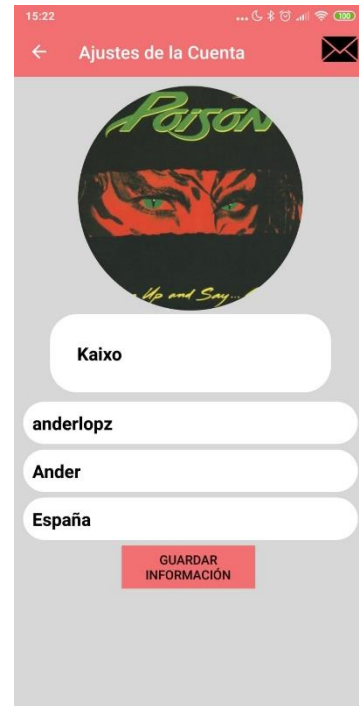
-**Sistema:** Eskaera jaso eta aldatu berri diren datuak datu basean gordetzen ditu.

Fluxu alternatiboa: Datu hutsak

-**Sistema:** Beharrezkoak diren datuak hutsak dauden errorea erakusten du. Kasu honetan erabiltzaile izena, izena, herrialdea edo profileko irudia

Fluxu alternatiboa: Erabiltzaile izena errepikatua

-**Sistema:** Aldatu nahi den erabiltzaile izena dagoeneko existitzen da datu basean errorea erakusten du.



.. 4.7 irudia Profila editatu gertaera fluxua

7. Gertaera fluxua: Erabiltzaileak bilatu

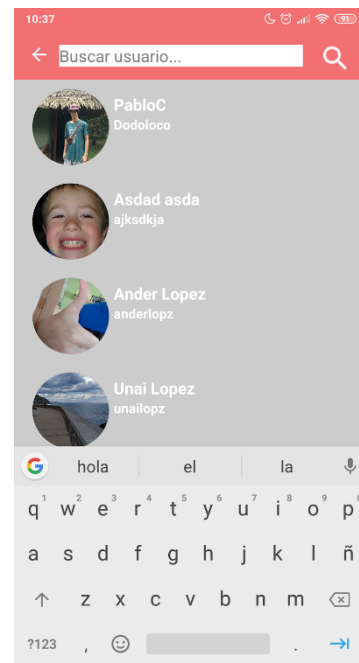
Fluxu printzipala

-**Erabiltzailea:** Erabiltzaileak bilatzeko GUI-ra joateko eskaera egiten du

-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du.

-**Erabiltzailea:** Erabiltzaile baten izena sartu eta bilaketa eskaera egiten du.

-**Sistema:** Sartutako datuak datu basean begiratu eta bat egiten badute erabiltzaile lista bat erakusten du.



.. 4.8 irudia Erabiltzaileak bilatu gertaera fluxua

8. Gertaera fluxua: Lagunak gehitu

Fluxu printzipala

-**Erabiltzailea:** Erabiltzailearen profilerara joateko eskaera egiten du. (ikus 5 Gertaera fluxua)

-**Sistema:** Eskaera onartu eta identifikazioak alderatu ondoren onen profilerara bideratzen du.

-**Erabiltzailea:** Lagun eskaera bat egiten du bere erabiltzaile identifikazio zenbakia emanaz.

-**Sistema:** Erabiltzailea identifikazio bidez datu basean bilatu eta lagun eskaera iristarazten dio bidalitako erabiltzailearen identifikazioarekin batera.

9. Gertaera fluxua: Laguna ezabatu

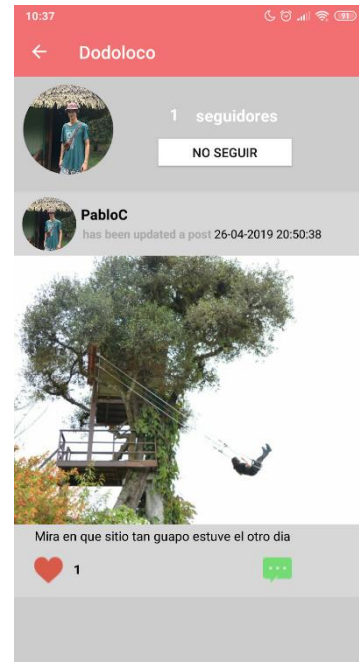
Fluxu printzipala

-**Erabiltzailea:** Erabiltzailearen profilerako joateko eskaera egiten du.

-**Sistema:** Eskaera onartu eta erabiltzailearen identifikazioa aztertu ondoren onen profilerako bideratzen du.

-**Erabiltzailea:** Lagunetatik kentzeko eskaera egiten du, profilaren identifikazio zenbakia emanaz.

-**Sistema:** Erabiltzailearen lagunaren listatik profilaren identifikazio zenbakia duen erabiltzailea ezabatzen du.



.. 4.9 irudia Laguna ezabatu gertaera fluxua

10. Gertaera fluxua: Lagun eskaera onartu

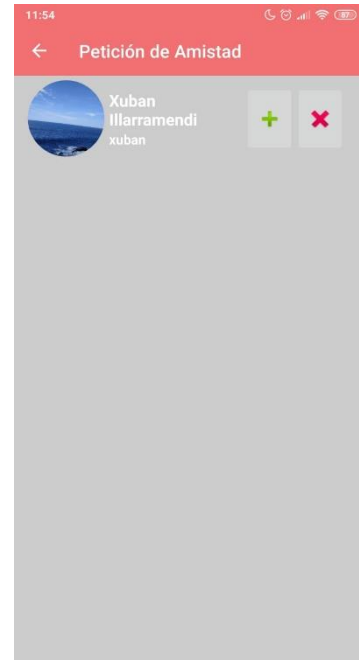
Fluxu printzipala

-**Erabiltzailea:** Lagunen eskaerak ikusteko eskaera bat egiten du, bere erabiltzaile identifikazio zenbakia emanaz.

-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du. Bertan, emandako identifikazio zenbakiak dituen eskaerak bistaratzen ditu.

-**Erabiltzailea:** Dagoen eskaera onartzen du.

-**Sistema:** Eskaeran zegoen erabiltzailea, uneko erabiltzailearen lagunaren listan sartzen du eta berdin beste erabiltzailearekin.



.. 4.10 irudia Lagun eskerak gertaera fluxua

11. Gertaera fluxua: Lagun eskaera ezeztatu

Fluxu printzipala

-**Erabiltzailea:** Dagoen eskaera ezeztatzen du.

-**Sistema:** Datu basetik lagun eskaera ezabatzen du.

12. Gertaera fluxua: Publikazioak publikatu

Fluxu printzipala

-**Erabiltzailea:** Publikazioak gehitzeko interfazera joateko eskaera egiten du bere erabiltzaile identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du.

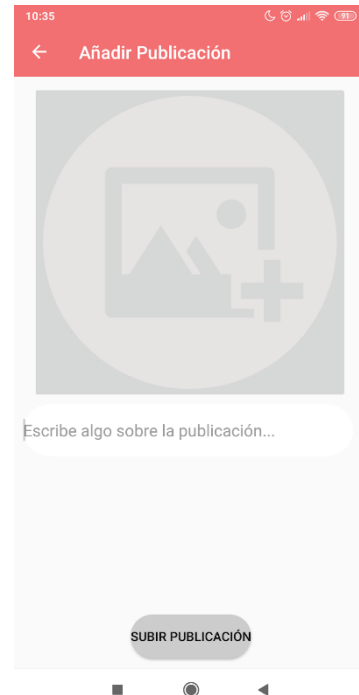
-**Erabiltzailea:** Publikazio mota aukeratzen du. "Irudi", "Bideo" edo "Testu". Egindako aukeraketaren arabera datu mota batzuk sartuko dira eta hauek igotzeko eskaera egingo da lehen emandako erabiltzaile identifikazioa eta publikazioari jarritako identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta bete beharreko atalak identifikatu ondoren publikazioa datu basera igotzen du.

Fluxu alternatiboa: Beharrezko datuak faltan

-**Sistema:** Publikazio mota Testua bada eta deskribapena hutsa badago abisua erakusten du.

Publikazio mota Irudi edo Bideo bada multimedia igotzeko abisua erakusten du.



4.11 irudia Publikazioak gertaera fluxua

13. Gertaera fluxua: Publikazio propioa ezabatu

Fluxu printzipala

-**Erabiltzailea:** Publikazioak ikusteko interfazera joateko eskaera egiten du bere eta publikazioaren identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta erabiltzailearen identifikazioa publikazioaren jabearen identifikazioarekin identifikatu ondoren hau dagokion publikaziora bideratzen du.

-**Erabiltzailea:** Publikazioa ezabatzeko eskaera egiten du.

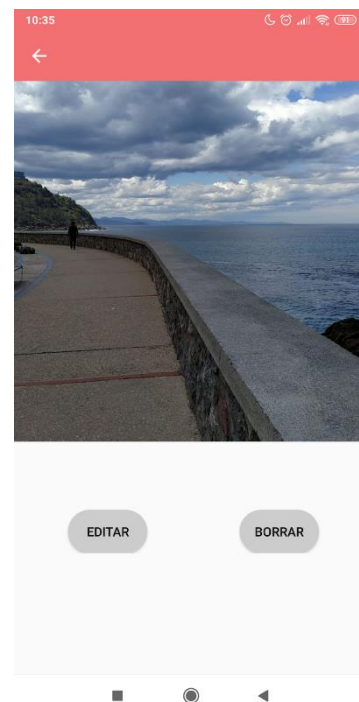
-**Sistema:** Eskaera onartu eta publikazioa datu basetik ezabatzen du.

14. Gertaera fluxua: Publikazio propioa editatu

Fluxu printzipala

-**Erabiltzailea:** Publikazioa ikusteko interfazera joateko eskaera egiten du bere eta publikazioaren erabiltzaile identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta erabiltzailearen identifikazioa publikazioaren jabearen identifikazioarekin identifikatu ondoren hau bideratzen du.



4.12 irudia Publikazioa editatu/ezbatu gertaera fluxua

-**Erabiltzailea:** Publikazioa editatzeko eskaera egiten du.

-**Sistema:** Eskaera onartu eta aldatu nahi dituen datuak erakusten dizkio, hauen gainean aldaketak egin ditzan.

-**Erabiltzailea:** Publikazioaren datuak aldatu eta aldaketa gordetzeko eskaera egiten du.

-**Sistema:** Eskaera onartu eta aldatu dituen datuak datu basean gordetzen ditu.

15. Gertaera fluxua: Lagunen publikazioak ikusi

Fluxu printzipala: Lagun guztien publikazioak

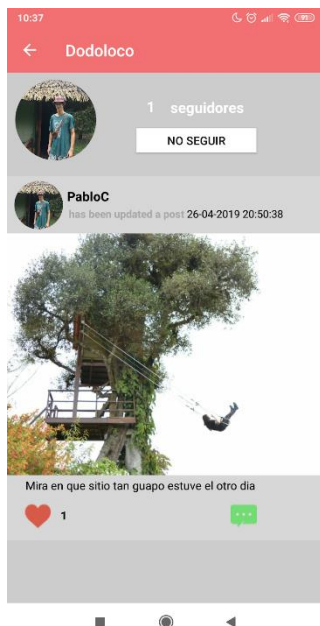
-**Erabiltzailea:** MainGUI-ra joateko eskaera egiten du bere erabiltzaile identifikazio zenbakia emanaz.

-**Sistema:** Eskaera onartu eta erabiltzailearen identifikazioa aztertu ondoren hau bideratzen du. Bertan bere erabiltzaile identifikazio zenbakira lagun bezala lotuak dituen beste erabiltzaileen identifikazio zenbakien bidez, hauen publikazioak erakutsiko ditu.

Fluxu printzipala: Lagun konkretuen publikazioak

-**Erabiltzailea:** Lagunen profilerako interfazera joateko eskaera egiten du (*ikus 5 Gertaera fluxua*)

-**Sistema:** Eskaera onartu eta lagunaren identifikadorea aztertu ondoren honen profilera bideratzen du eta bertan profilaren jabearen publikazioak erakusten ditu.



.. 4.13 irudia

Lagunen publikazioak ikusi gertaera fluxua

16. Gertaera fluxua: Publikazioetan iritzia eman

Fluxu printzipala

-**Erabiltzailea:** Publikazio baten aurrean iritzia ikusteko eskaera egiten du.

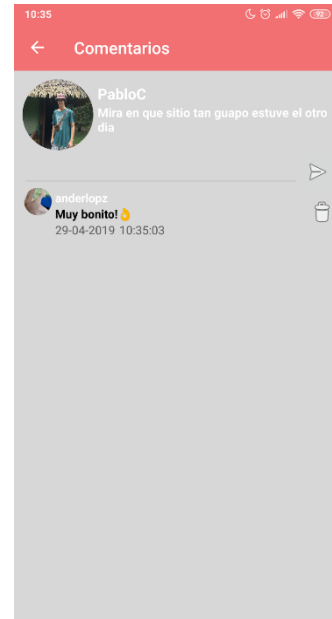
-**Sistema:** Eskaera onartu eta publikazioaren iruzkin lista erakusten du.

-**Erabiltzailea:** Iritzi bat idazteko hau idatzi eta bere erabiltzaile identifikazio zenbakiarekin batera iritzia bidaltzeko eskaera egiten du

-**Sistema:** Eskaera onartu eta iritzia datu basera igotzen du.

Fluxu alternatiboa: Iritzi hutsa

-**Sistema:** Iritzia hutsa dagoela abisua erakusten du.



.. 4.14 irudia Publikazioen iritzia gertaera fluxua

17. Gertaera fluxua: Publikazioetan iritzia ezabatu

Fluxu printzipala

-**Erabiltzailea:** Iritzia ezabatzeko eskaera egin.

-**Sistema:** Eskaera eskatutako erabiltzailearen identifikadorea aztertu eta iritzia sortzailearen identifikadorearekin konparatu ondoren eskaera onartu eta iritzia ezabatzen du.

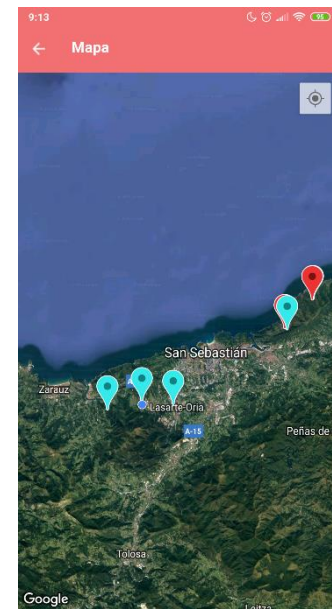
18. Gertaera fluxua: Mapa kontsultatu

Fluxu printzipala

-**Erabiltzailea:** Mapa kontsultatzeko eskaera egiten du bere erabiltzaile identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du honen identifikadorea aztertu ondoren.

Bertan erregistratuak dauden blokeak erakutsiko dira eta emandako erabiltzaile identifikadorearen bidez zein bloke eginak (*urdinez*) eta zeintzuk ez (*gorriz*) agertuko dira.



.. 4.15 irudia Mapa kontsultatu gertaera fluxua

19. Gertaera fluxua: Blokea sortu

Fluxu printzipala: Blokea sortu Mapa bidez

-**Erabiltzailea:** Bloke bat sortzeko mapa bidezko eskaera egiten du.

-**Sistema:** Eskaera onartu eta erabiltzailea mapa interfazera bideratzen du.

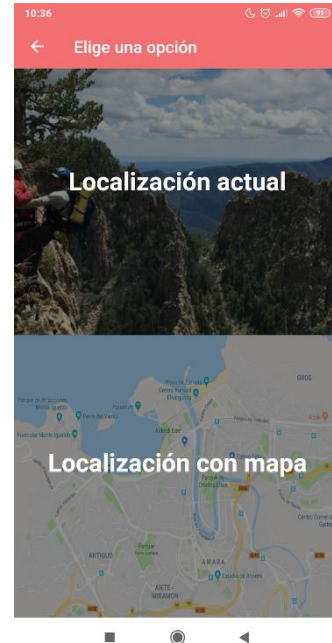
-**Erabiltzailea:** Mapa interfazeaz zona bat aukeratzen du honen latitude eta longitudeak emanez

-**Sistema:** Aukeratutako zonaren koordinatuen bidez lokalizazioa lortzen du.

Fluxu printzipala: Blokea sortu uneko kokapen bidez

-**Erabiltzailea:** Bloke bat sortzeko uneko kokapena bidezko eskaera egiten du.

-**Sistema:** Eskaera onartu eta GPS bidez uneko koordinatuak eta lokalizazioak lortzen ditu.



4.16 irudia Blokea sortu gertaera fluxua

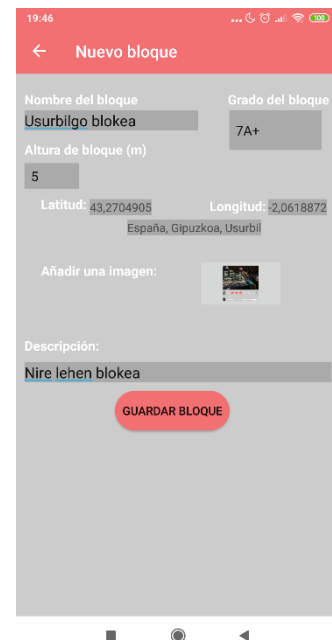
Fluxu printzipala

-**Erabiltzailea:** Erabiltzaileak blokea sortzeko beharrezko datuak sartzen ditu: Blokearen izena, gradu maila, altuera, latitudea, longitudea, lokalizazioa, irudi bat eta deskripzioa.

-**Sistema:** Eskaera onartu eta blokea datu basean erregistratzen (sartutako datuekin) du sortu duen erabiltzailearen identifikazioa eta blokeari jarritako identifikazioarekin batera.

Fluxu alternatiboa: Datu hutsak

-**Sistema:** Beharrezkoak diren datuak hutsak dauden abisua erakusten du.



4.17 irudia Blokea sortu gertaera fluxua

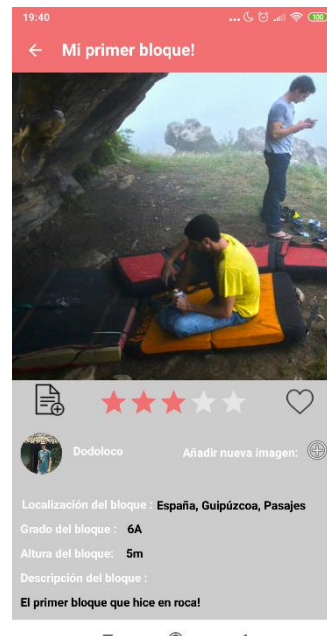
20. Gertaera fluxua: Blokearen informazioa ikusi

Fluxu printzipala

-**Erabiltzailea:** Bloke baten profilerara sartzeko eskera egiten du honen identifikazio zenbakia eta bere erabiltzaile identifikazio zenbakia emanaz.

-**Sistema:** Eskaera onartu eta erabiltzailearen identifikazioa blokearen sortzailearen identifikazioarekin alderatu ondoren, hau bideratzen du.

-**Sistema:** Blokeko profilean sistemak honen datu guztiak erakusten ditu.



.. 4.18 irudia Blokearen informazioa ikusi gertaera fluxua

21. Gertaera fluxua: Gustuko blokeak gehitu

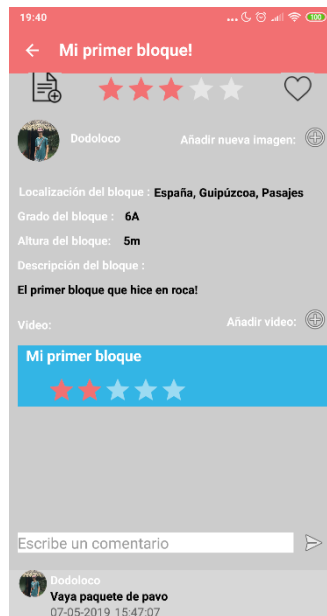
Fluxu printzipala

-**Erabiltzailea:** Bloke baten profilerara sartzeko eskera egiten du.

-**Sistema:** Eskaera onartu eta erabiltzailearen identifikazioa blokearen sortzailearen identifikazioarekin alderatu ondoren, hau bideratzen du.

-**Erabiltzailea:** Blokeak gustukoetara gehitzeko eskaera egiten du blokearen identifikazio zenbakia lortuz.

-**Sistema:** Eskaera onartu eta erabiltzailearen gustuko blokeetara gehitzen du blokearen datu basean.



.. 4.19 irudia Gustuko blokeak gehitu gertaera fluxua

22. Gertaera fluxua: Gustuko blokeetatik kendu

Fluxu printzipala

-**Erabiltzailea:** Blokeak gustuko blokeetatik kentzeko eskaera egiten du.

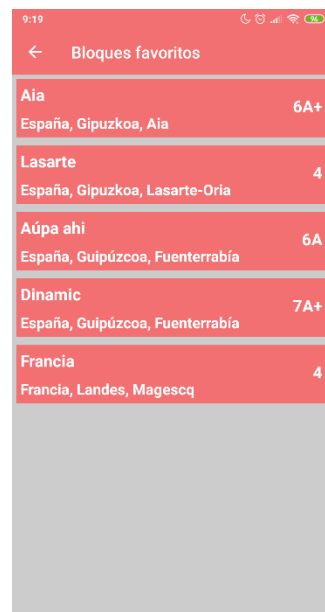
-**Sistema:** Eskaera onartu eta datu basean erabiltzailearen gustuko blokeak listatik blokeak kentzen du.

23.Gertaera fluxua: Gustuko blokeak bistaratu

Fluxu printzipala

-**Erabiltzailea:** Gustuko blokeen lista erakusteko eskaera egiten du bere erabiltzaile identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta erabiltzailea gustuko blokeen lista erakusten duen interfazera bideratzen du. Bertan, jasotako identifikadore zenbakia duen erabiltzailearen gustuko blokeen lista erakusten du.



. 4.20 irudia Gustuko
blokeak bistaratu gertaera fluxua

24.Gertaera fluxua: Blokea baloratu

Fluxu printzipala

-**Erabiltzailea:** Blokea baloratzeko eskaera egiten du bere erabiltzaile identifikadore zenbakia eta balorazio bat (*1etik 5era*) emanez.

-**Sistema:** Balorazioa lehendik eginda duen egiaztatzen du identifikazio zenbakiaren bidez eta ahala ez bada balorazioa jaso eta hauen media datu basean gordetzen du

25.Gertaera fluxua: Blokea egiteko listara gehitu

Fluxu printzipala

-**Erabiltzailea:** Blokea egiteko dauzkan blokeen listara gehitzeko eskaera egiten du bere erabiltzaile identifikazio zenbakia eta blokearen identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta blokea egiteko blokeen listara gehitzen du datu basean.

26.Gertaera fluxua: Blokea egin

Fluxu printzipala

-**Erabiltzailea:** Blokea burutzeko eskaera egiten du bere erabiltzaile eta blokearen identifikazio zenbakia emanez

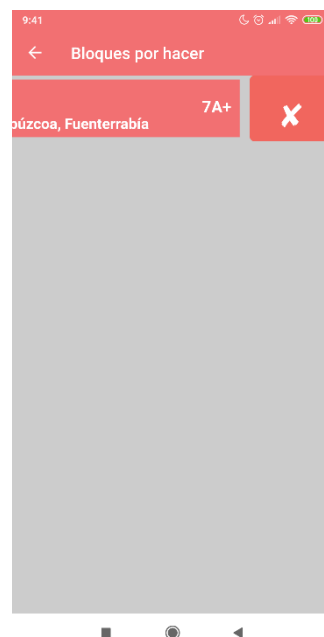
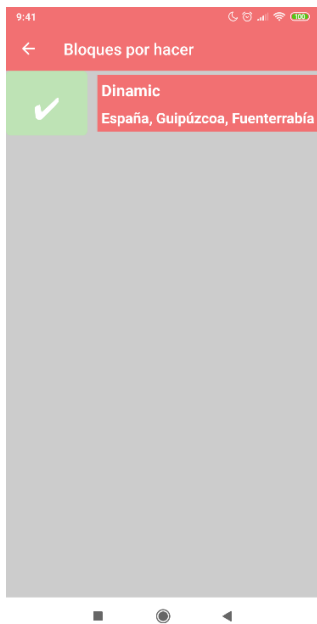
-**Sistema:** Eskaera onartu eta blokea egindako blokeetara gehitzen du datu basean.

27.Gertaera fluxua: Blokea ez egin

Fluxu printzipala

-**Erabiltzailea:** Blokea ez burutzeko eskaera egiten du bere erabiltzaile eta blokearen identifikazio zenbakia emanez

-**Sistema:** Eskaera onartu eta blokea egiteko blokeetatik ezabatzen du datu basean



. 4.21 irudia

Blokea burutu/ez burutu gertaera fluxua

28.Gertaera fluxua: Blokerak Argazki/Bideoa/Iritzia igo

Fluxu printzipala: Argazkia igo

-**Erabiltzailea:** Blokearen profilean argazki bat igotzeko eskaera egiten du bere erabiltzaile identifikadore zenbakia, irudia eta honen identifikazio zenbakia bat emanez

-**Sistema:** Eskaera onartu eta blokearen profilean argazki bat gehitzen du.

Fluxu printzipala: Bideoa igo

-**Erabiltzailea:** Blokearen profilean bideo bat igotzeko eskaera egiten du bere erabiltzaile identifikadore zenbakia, bideoa eta honen identifikazio zenbakia bat emanez

-**Sistema:** Eskaera onartu eta blokearen profilean bideo bat gehitzen du.

Fluxu printzipala: Iritzia igo

-**Erabiltzailea:** Blokearen profilean iritzi bat igotzeko eskaera egiten du bere erabiltzaile identifikadore zenbakia, iritzi eta honen identifikazio zenbakia bat emanez

-**Sistema:** Eskaera onartu eta blokearen profilean iritzi bat gehitzen du.

Fluxu alternatiboa: Datu hutsak

-**Sistema:** Beharrezkoak diren datuak hutsak dauden errorea erakusten du.

29.Gertaera fluxua: Bideoak baloratu

Fluxu printzipala

-**Erabiltzailea:** Blokeko bideoa baloratzeko eskaera egiten du bere erabiltzaile identifikadore zenbakia eta balorazio bat (*1etik 5era*) emanaz.

-**Sistema:** Balorazioa lehendik eginda duen egiaztatzen du eta ahala ez bada balorazioa jaso eta hauen media datu basean gordetzen du

30.Gertaera fluxua: Blokea izen bidez bilatu

Fluxu printzipala

-**Erabiltzailea:** Blokeak bilatzeko GUI-ra joateko eskaera egiten du

-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du.

-**Erabiltzailea:** Bloke baten izena sartu eta bilaketa eskaera egiten du.

-**Sistema:** Sartutako datuak datu basean begiratu eta bat egiten badute bloke lista bat erakusten du.

31.Gertaera fluxua: Blokea GEO lokalizazio bidez bilatu

Fluxu printzipala

-**Erabiltzailea:** Distantzia bat aukeratu eta bilaketa egiteko eskaera egiten du.

-**Sistema:** Datuak uneko lokalizazioarekin konparatu eta bloke lista bat erakusten du.



4.22 irudia Blokea bilatu gertaera fluxua

32.Gertaera fluxua: Erronkak manualki sortu

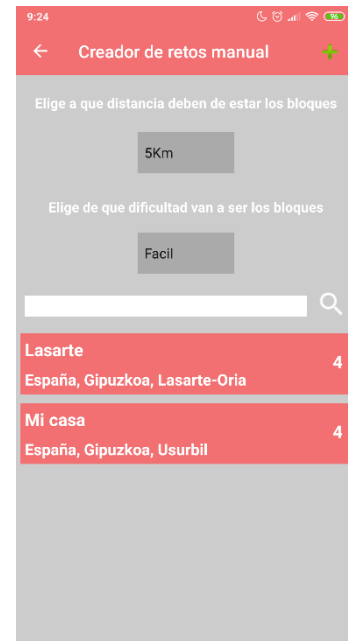
Fluxu printzipala

-**Erabiltzailea:** Erronkak sortzeko GUI-ra joateko eskaera egiten du eta bere erabiltzaile identifikazio zenbakia ematen dio.

-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du.

-**Erabiltzailea:** Distantzia, zailtasuna eta blokeak aukeratzeko datu bezala eta erronka sortzeko eskaera egiten du.

-**Sistema:** Eskaera onartu eta erronka berri bat sortzen du.



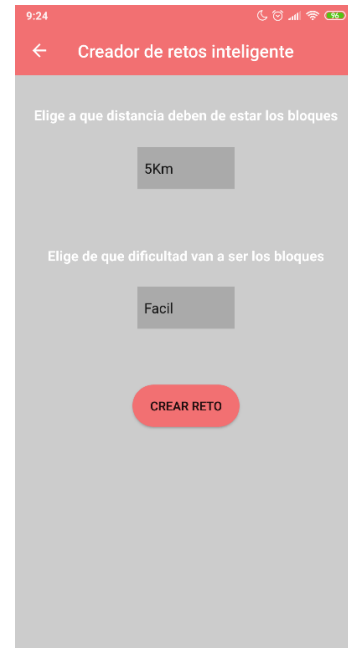
4.23 irudia Erronka manualki sortu gertaera fluxua

33. Gertaera fluxua: Erronkak modu adimentsuan sortu

Fluxu printzipala

-**Erabiltzailea:** Distantzia eta zailtasuna aukeratzen ditu datu bezala eta erronka sortzeko eskaera egiten du.

-**Sistema:** Eskaera onartu eta erronka berri bat sortzen du sartutako datuekin automatikoki blokeak aukeratuz.



. 4.24 irudia Erronka automatikoki sortu gertaera fluxua

34. Gertaera fluxua: Erronkan parte hartu

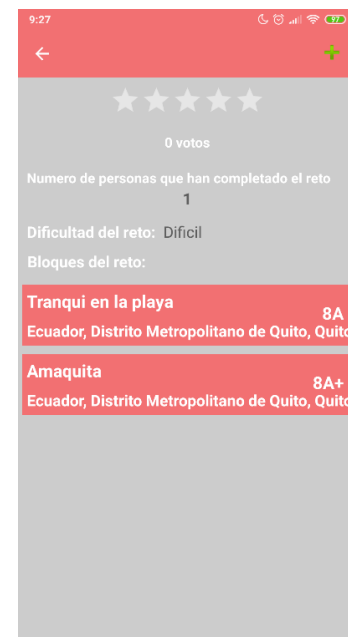
Fluxu printzipala

-**Erabiltzailea:** Erronka baten profileraren sartzeko eskaera egiten du bere erabiltzaile identifikazio zenbakia eta erronkaren identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du.

-**Erabiltzailea:** Erronkan parte hartzeko eskaera egiten du.

-**Sistema:** Eskaera onartu eta erabiltzailea erronkaren parte hartzaileen artean gehitzen du.



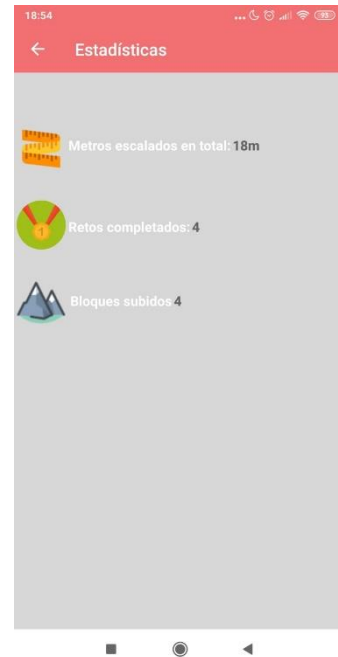
. 4.25 irudia Erronkan parte hartu gertaera fluxua

35. Gertaera fluxua: Estatistikak ikusi

Fluxu printzipala

-**Erabiltzailea:** Bakoitzaren profilean egonda, estatistikak ikusteko eskaera egiten du.

-**Sistema:** Eskaera onartu eta estatistikak erakusten dizkio.



.. 4.26 irudia Estatistikak ikusi gertaera fluxua

36. Gertaera fluxua: Erronkako bloke bat bete

Fluxu printzipala

-**Erabiltzailea:** Erronkaren profileko blokea betetzeko eskaera egiten du blokearen identifikazio zenbakia emanez.

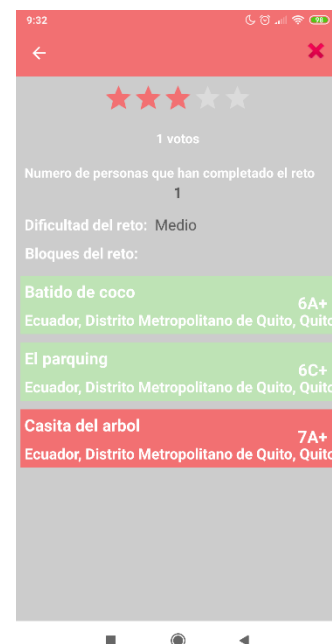
-**Sistema:** Eskaera onartu eta datu basean erronkaren bloke zehatz hori beteta duela gordetzen du. Gainera erabiltzaileak eginda dituen blokeetara ere gehitzen du bloke hau.

37. Gertaera fluxua: Erronka bete

Fluxu printzipala

-**Erabiltzailea:** Erronkaren profileko blokea betetzeko eskaera egiten du.

-**Sistema:** Eskaera onartu eta datu basean erronkaren bloke zehatz hori beteta duela gordetzen du. Ondore datu basean erronkak dituen bloke kopurua eta erabiltzaileak egin dituenak aztertu eta guztiak eginak badigu erronka beteta duela erakusten du.



.. 4.27 irudia Erronkako blokeak bete gertaera fluxua

38. Gertaera fluxua: Erronkak ezabatu

Fluxu printzipala

-**Erabiltzailea:** Erronka ezabatzeko eskaera egiten du honen erronkaren identifikazio zenbakia emanez.

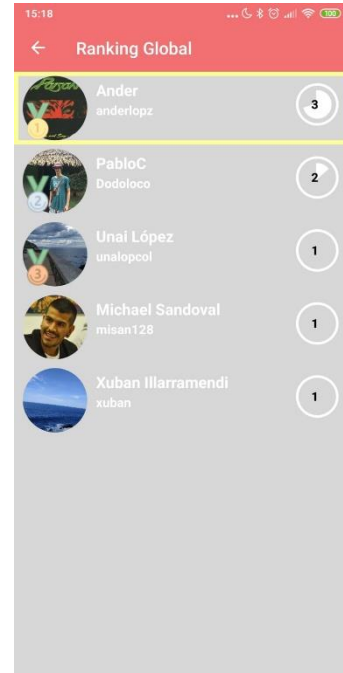
-**Sistema:** Eskaera onartu eta erabiltzailea emandako identifikazio zenbakia duen erronkatik kentzen du.

39.Gertaera fluxua: Erabiltzaile guztien Ranking-a ikusi

Fluxu printzipala

-**Erabiltzailea:** Ranking globala ikusteko eskaera egiten du

-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du. Datu basean dauden erabiltzaile guztien ranking bat erakusten dio hauen mailaren arabera antolatuz.



.. 4.28 irudia Rankinga ikusi gertaera fluxua

40.Gertaera fluxua: Konbertsazioa sortu

Fluxu printzipala

-**Erabiltzailea:** Mezuak ikusteko interfazera joateko eskaera egiten du bere erabiltzaile identifikazio zenbakia emanda.

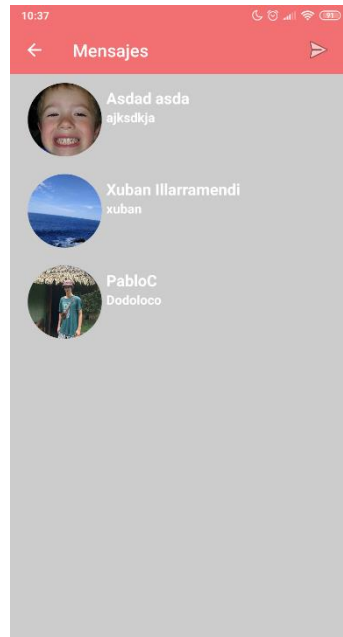
-**Sistema:** Eskaera onartu eta erabiltzailea bideratzen du.

-**Erabiltzailea:** Mezuak bidaltzeko erabiltzaile lista zabaltzeko eskaera egiten du.

-**Sistema:** Eskaera onartu eta erabiltzaile honen lagunen erabiltzaile lista erakusten du.

-**Erabiltzailea:** Erabiltzaile bat aukeratu eta konbertsazio bat hasteko eskaera egiten du.

-**Sistema:** Eskaera onartu eta bi erabiltzaileen artean konbertsazio bat sortzen du.



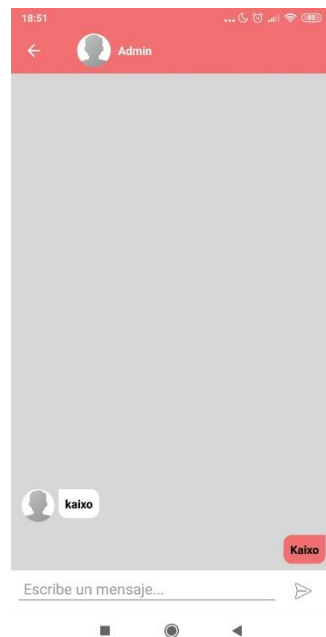
.. 4.29 irudia Konbertsazioa sortu gertaera fluxua

Fluxu printzipala: Administratzailearekin

Konbertsazio bat sortzeko metodo berdina jarraitzen duen arren, kasu honetan erabiltzailearen datuak aldatzeko atalean egin beharko da bertan zuzenean konbertsazio bat sortuko da goiko mezu ikonoari klik eginez.

-**Erabiltzailea:** Administratzailearekin hitz egiteko eskaera egiten du bere erabiltzaile identifikazioa emanez.

-**Sistema:** Eskaera onartu eta erabiltzailearen eta administratzailearen artean konbertsazio bat sortzen du.



.. 4.30 irudia Konbertsazioa sortu gertaera fluxua

41.Gertaera fluxua: Konbertsazioa ezabatu

Fluxu printzipala

-**Erabiltzailea:** Konbertsazioa ezabatzeko eskaera egiten du bere erabiltzaile identifikazioa eta konbertsazioa duen erabiltzailearen identifikazioa emanez

-**Sistema:** Eskaera onartu eta konbertsazioa ezabatzen du eskaera egin duen erabiltzaile horrentzako.

42.Gertaera fluxua: Mezua gehitu konbertsaziora

Fluxu printzipala

-**Erabiltzailea:** Mezu bat sartzen du datu bezala eta konbertsazioan mezu hori sartzeko eskaera egiten du

-**Sistema:** Eskaera onartu eta konbertsazioan mezua gehitzen du.

Fluxu alternatiboa: Datu hutsak

-**Sistema:** Beharrezkoak diren datuak hutsak dauden errorea erakusten du.



.. 4.31 irudia Mezuak gehitu gertaera fluxua

ADMINISTRATZAILEA

Administratzaile aktoreak erabiltzaile arrunt batek egin ditzakeen erabilpen kasu batzuk dituen eta hauen funtzionalitatea berdin betetzen diren ez dira berriz ere atal honetan errepikatuko (Konbertsazioak). Aipatzekoak dira hala ere zeintzuk diren erabilpen kasu hauek: LogIn, Erabiltzaileak bilatu, Profilak ikusi, Blokeen informazioa ikusi, Blokeak bilatu, Konbertsazioak sortu, Konbertsazioak ezabatu eta Mezuak konbertsaziora gehitu.

Administratzaile moduan logeatzean lagunak gehitzeko, blokeak gehitzeko eta beste horrelako aukerak eta ezaugarriak ez direnez garrantzitsuak, ez dira azalduko erabiltzaile honentzako.

Administratzaile kontua aplikazioaren gestiorako balioko du, hau da, erabiltzaile normal batek bere kontua gauza txarretarako erabiltzen badu edo beste erabiltzaileei gaizki esaka ibili diren erabiltzaileei aurre egiteko erabiliko da kontu hau.

Adibide bezala erabiltzaile batek bloke bat gehitzen badu eta kexa asko jaso baditu administratzaileak hau berrikusi eta hartu beharreko erabakiak hartuko ditu; Blokea sortu duen erabiltzailearekin kontaktuan jarri, blokea ezabatuz, ...

Hurrengo gertaera fluxuetan administratzailearen erabilpenak ikusiko dira:

43.Gertaera fluxua: Erabiltzaileak ezabatu

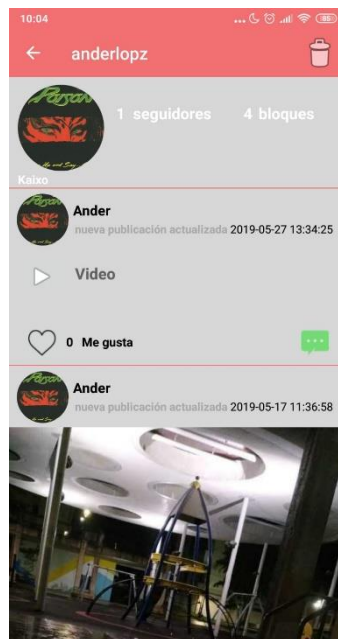
Fluxu printzipala

-**Erabiltzailea:** Profil bat ikusteko eskaera egiten du honek ikusi nahi duen profilaren identifikazio zenbakia emanez.

-**Sistema:** Eskaera onartu eta jasotako identifikazio zenbakia duen erabiltzailearen profilerara bidaltzen du.

-**Erabiltzailea:** Erabiltzailea ezabatzeko eskaera egiten du (jadanik ezabatu nahi den erabiltzailearen identifikazio zenbakia sistemaren eskuetan dago)

-**Sistema:** Eskaera onartu eta datu basean erabiltzailea ezabatzeaz gain, berak sortutako objektu guztiak ezabatzen ditu datu basean.



.. 4.32 irudia Erabiltzaileak ezabatu gertaera fluxua

Aplikazioaren domeinuaren eredia 10 klasez osatuta dago, bakoitza bere atributuekin eta loturekin beste klaseen artean:

- **Erabiltzaile:** Erabiltzaile klase *abstraktua* bat da, ondoren Administratzailea eta Erabiltzaile arrunta klaseak klase *abstraktu* honen herentziak dira, horrela klase honen atributu heredatuz.
- **Administratzailea:** Klase honek lehengoko klase *abstraktuaren* atributuak heredatzeaz gain ez du beste atributu berezirik. Konbertsazio klasearekin lotura bat edukiko du, beste Erabiltzaile arruntekin hainbat konbertsazio edukitzeko aukerarekin.
- **Erabiltzaile arrunta:** Klase honek ere Erabiltzaile klase *abstraktutik* atributu guztiak heredatzen ditu. Honez gain ez du beste atributu berezirik baina hainbat lotura ditu domeinuko beste hainbat klaserekin: Konbertsazio (beste erabiltzaileekin edo administratzailearekin mezuak bidaltzeko), Blokea (Blokeak sortzeko, egiteko eta gustuko blokeetara gehitzeko), Publikazioa (Publikazioak publikatzeko, gustukoetara gehitzeko eta bertan iritzia emateko) eta Erronka (Blokeekin erronkak sortu eta egiteko).
- **Konbertsazio:** Konbertsazio klasea sortu zen data du atributu bezala. Gainera konbertsazio klasea hainbat mezuz osatua dago, nahiz eta hasiera batean konbertsazio bat sortzean ez den beharrezkoa mezurik edukitzea (0..*)
- **Mezu:** Mezu klaseak mezua eta mezua bidalitako data gortzen ditu. Mezu honek konbertsazio bat edukiko du (mezua konbertsazio baten parte da), eta beste aldetik bi erabiltzaile, mezuaren jatorria eta helburua.
- **Publikazioa:** Publikazioa klase *abstraktu* bat da. Bi atributu ditu, publikazioaren igotze data eta honen ordua. Klase honek Erabiltzaile arrunta klasearekin hainbat lotura ditu, honen sortzailea zein den jakiteko, zein duen gustuko eta eduki ditzakeen iritzia gordetzeko. Klase *abstraktu* honetatik beste hiru klasek jasoko dute herentzia: Argazkia, Bideoa eta Testua.
- **Argazkia:** Argazkia klaseak Publikazioa klase *abstraktuko* atributuak heredatzeaz gain bere atributu propioa du, argazkia
- **Bideoa:** Bideoa klaseak Publikazioa klase *abstraktuko* atributuak heredatzeaz gain bere atributu propioa du, bideoa
- **Testua:** Testua klaseak Publikazioa klase *abstraktuko* atributuak heredatzeaz gain bere atributu propioa du, testua
- **Iritzia:** Iritzia klasea, klase asoziatua da Publikazio bakoitzak hainbat Erabiltzaile arruntan iritzia gordeko du, baita ere ze orduan eta zein egunetan izan zen idatzia iritzi bakoitza.

- **Blokea:** Blokea klaseak hainbat atributu eta lotura ditu beste klaseekin. Atributu aldetik gerora bloke batek edukiko dituenak izango dira: Izena, zailtasuna, balorazioa, deskribapena, lokalizazioa, latitudea, longitudea eta altuera. Honez gain esan bezala, beste klaseekin hainbat lotura ditu, adibidez: Erabiltzaile arrunta klasearekin gustuko, sortutako eta egindako blokeak lortuko ditugu, gainera egindako blokeak gerora Estatistika klasean atributu batzuk kalkulatzeko balioko digu. BlokeArgazki eta BlokeBideo klaseetan blokearen argazki zein bideoak atributuak edukiko ditugu, gainera bideoen kasuan hauek baloratzeko atributua ere edukiko dugu bideoak baloratzeko. Azkenik azkeneko lotura Erronka klasearekin da, bertan blokeak gordetzeko.
- **BlokeArgazki:** Klase honetan blokearekin zerikusia duten argazkiak eta bere titulua izango dira honen atributuak.
- **BlokeBideo:** Klase honetan blokearekin zerikusia duten bideoak eta bere titulua izango dira klase honen atributuak. Gainera bideoen kasuan aipatu bezala baloratzeko atributua ere edukiko dute.
- **Erronkak:** Erronkak klaseak hiru atributu ditu: izena, balorazioa eta zailtasuna. Beste klaseekin loturak aipatzean Erabiltzaile arruntarekin dugu lehenengoa, bertan sortutako erronkak eta parte hartutako erronkak atributuak gehituko dira. Beste aldetik Erronka klasearekin lotura ere izango du, bertan erronkan egongo diren blokeak gordeko dira. Azkenik kasu honetan ere Estatistika klasearekin ere lotura bat dauka atributu batzuk kalkulatzeko.
- **Erronka:** Klase honek Erronkak klaseak behar dituen blokeak edukiko ditu, azken finean Erronkak bakoitzak bloke zehatz batzuk edukiko ditu.
- **Estatistika:** Estatistika klasea, asoziatutako klase bat da. Bi asoziazio ditu, Blokea eta Erabiltzaile arrunten arteko loturarekin, erabiltzaileak egin dituen bloke kopurua eta metro kopurua kalkulatzeko eta beste aldetik Erronkak eta Erabiltzaile arruntaren artekoa, erabiltzaileak egin dituen erronka kopurua kalkulatzeko. Bestalde data atributua edukiko du estatistika oiek noiz hartuak izan diren jakiteko.

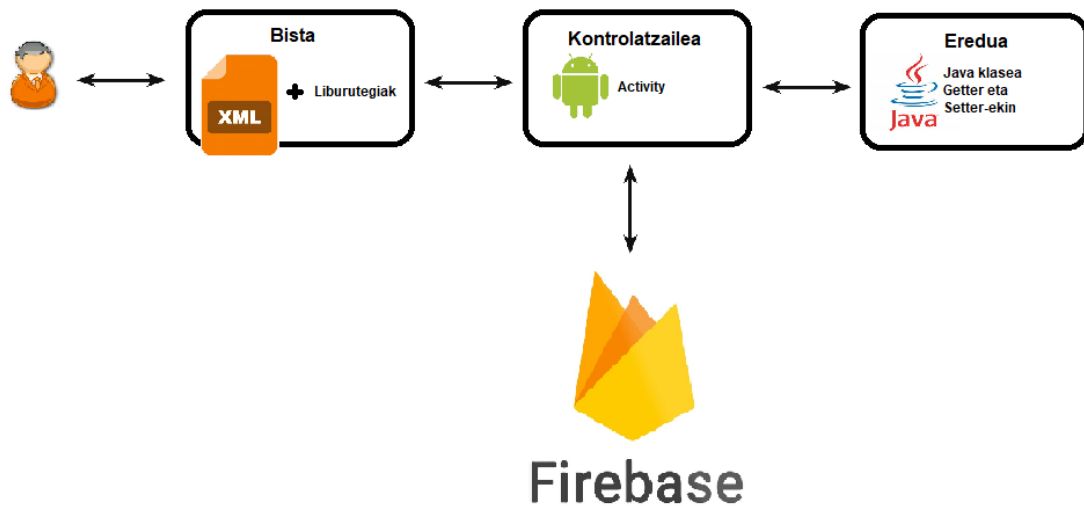
5

Diseinua

Memoriako atal honetan aplikazioaren diseinuari buruz hitz egingo dugu. Zehatzago izanda, aplikazioa garatzeko erabili den arkitekturari buruz (MVC “*Model View Controller*”), datuak gordetzeko erabili den datu basearen egiturari buruz eta erabilpen kasu garrantzitsuenen sekuentzia diagramak azalduko dira.

5.1. Arkitektura

Kapitulu honen sarreran aipatu bezala, garatutako mugikorrerako aplikazio honek MVC arkitektura jarraitzen du. Honen arabera erabiltzailea aplikazioarekin komunikatu ahal izateko MVC-ko Bista atala erabiliko dugu. Bista atalean elementuak lortu ahal izateko Kontrolatzaile atala erabili beharko dugu eta honek datu hauek adibidez datu base batetik lortzeko Eredu atala erabiliko du. Hurrengo irudian ikus daiteke nola dagoen egituratua arkitektura hau gure aplikazioan:



. 5.1 irudia Proiektuaren arkitektura

Behin irudia ikusita banan bana azalduko ditugu arkitektura hau egituratzen dituen hiru atalak:

5.1.1 Eredua

Eredu (*Model*) atalak gerora Kontrolatzailea *Firestore* plataforman kokatua dagoen datu basea atzitzeko erabiliko den lotura da. Datu baseak gordetako objektu bakoitzaren atributuak lortzeko balioko digu arkitektura honetako maila honek. Hau java klase bat izango da eta aplikazioarekin lortu nahi ditugun objektu bakoitzarentzat java klase bat edukiko dugu. Adibidez gure kasuan "Publikazioak", "Blokak", "Erronkak" eta "Erabiltzaileak" jasotzeko erabiliko dugu hauen atributuak lortuz klase hauen bidez.

Aurrerago memoriako *Implementazio* atalean azalduko da MVC arkitekturako maila bakoitza zein modutan inplementatu diren adibide batzuen bidez.

5.1.2 Bista

Bista (*View*) mailaren bidez aplikazioaren interfazea ikusi ahal izango dugu. Dagoeneko estatikoak diren datuak adierazteaz gain, Kontrolatzaileak Ereduaren laguntzaz lortutako datuak era dinamiko batean ere adieraziko ditu, adibidez publikazioak agertzen diren

bistetan, bertan publikazio berri bat eguneratzen bada bista hau dinamiko aldatuko da elementu berri hau erakutsiz.

Gure aplikazioaren kasuan bista hauen inplementazioa XML bidez egin da. *Android Studio*-ren bidez Bistak edo bertan *Layout* hitzarekin erabiltzen diren aplikazioko interfazeak garatzeko. Plataforma honek ekartzen dituen liburutegiez gain, beste garatzaileek sortutako liburutegiak ere erabili dira proiektu honetan. Adibide modura erabiltzaileen profileko irudia ikusteko *Android Studio*-n erabiltzen den *ImageView* elementua erabili beharrean *CircleImageView* elementu berria erabili da *GitHub* web orritik garatzaile batek egindako liburutegi batetik atzitua.

5.1.3 Kontrolatzailea

Kontrolatzaile (*Controller*) maila da ikusiko dugun azkeneko maila. Maila honek egiten du lan gehien bera baita *Firebase* plataformarekin komunikatzen dena eta beraz datu base eta beste zerbitzuekin lan egiten duen maila. Erabiltzaileak Bista geruza erabiliz aplikazioaren bide eskaerak egiten dituenean, adibidez erabiltzaile guztien lista erakustea, maila hau izango da hau era dinamiko batean Bista geruzan erakustez.

Lehen esan bezala maila honen bidez datu basearekin eragiketa guztiak egiten ditugu; datuak sortu, datuak lortu, datuak aldatu, datuak ezabatu, multimedia fitxategian igotze eta ezabatzea, erabiltzaileen autentifikazioa, Hasiera batean Kontrolatzaileak Eredu mailaren laguntza soilik datu basetik datuak lortu nahi dituenean soilik erabiliko ditu, gainontzeko eragiketa guztiak egiteko Kontrolatzaileak berak egingo ditu. Kontrolatzailea eta *Firebase*-en arteko komunikazio eta gauzatzen diren eragiketen inplementazioak esan bezala ez dira memoriako atal honetan azalduko.

Arkitektura honen hiru geruzen inplementazioa bezeroan bertan egin da, azken finean *Firebase* erabiltzearen abantailetakoa bat zerbitzari aldea inplementatzeko beharra kentzen du.

5.2. Datuen egitura

Proiektu honetan erabilitako datu basea *Firebase* zerbitzarietan kokatua dago. Datu baseaz gain *Firebase*-ek multimedia fitxategiak gordetzeko hodei bidezko fitxategi sistema bat eta erabiltzaileen kautotzea egiteko zerbitzua ere eskaintzen ditu. Beraz aplikazioak behar dituen datu guztiak *Firebase* plataforman edukiko ditugu gordeta.

Esan bezala *Firebase*-ek ezaugarri asko ditu baina proiektu honetan erabili direnak hiru nagusienak izan dira: *Firebase Authentication*[6], *Realtime Database*[7] eta *Firebase Storage*[8].

Kasu honetan *Realtime Database* tresnari buruz hitz egingo dugu. Hasteko esan beharra dago datu base hau NoSQL[9] motako dela eta beraz datu base ez erlazional baten aurrean gaudela. Honez gain JSON[10] formatua jarraitzen duela ere aipatu behar da. Bertan aplikazioak behar duen informazioa jasoko da, gure kasuan erabiltzaileei, blokeei, erronkei, publikazioei eta beste hainbat objektuei buruzko informazioa.

Aipatu berri diren objektuek multimedia fitxategi bat edo gehiago atzitu a baldin badute hauek *Firestore Storage* tresnaren bidez gordeko dira eta objektuen atributuetan hauek gordeta dauden helbide (*path*) edukiko dute gordeta ondoren erreferentziatu ahal izateko.

Firestore Storage-k bi helbide mota ditu objektuak erreferentziatzen dutena, ondore MVC arkitekturan Bistan erakutsi ahal izateko eta bestea multimedia fitxategia gordeta dagoen helbidea. Bertan aplikazioaren publikazioak, blokeek edo bestelako objektuek eduki ditzaketen bideo zein irudiak gordeko dira.





Datu basearen egitura erakutsiko da hurrengo irudian (*ikus 5.2 irudia*), bertan datu basearen egitura nagusia ikusiko da baina handiegia denez Blokeen zatia ikusiko da modu sakonago batean bere atributu guztiekin:



Proiektu honen kasuan 5.2 irudian ikus daitekeen bezala Blokeak (*Boulders*) objektuak bakoitza bere id zenbakiekin banatzen dira. Ondoren bere barnean hainbat atributu ditu nahiz eta denak ez egon. NoSQL motako datu base bat denez, objektuek edukiko duten egitura ez da hasieratik definitu behar eta beraz atributu guztiak ez dira hasieratik agertuko.









Adibide modura *BoulderImages* agertzen den bezala, non bertan bloke zehatz horrek dituen irudiak gordeta dauden helbideak biltegitratzen ditu, beste atributu bat gehiago faltako litzake, hau da, *BoulderVideos*, hitzak dioen bezala blokeen bideoak gordetzeko balioko duena. Bloke bat sortzerakoan irudiak igo daitezke soilik, beraz bideoen atributua oraindik ez da sortua izango, hau erabiltzaile batek bideo bat igotzean gertatuko da.


Aurrerago aipatu bezala multimediazko fitxategiak *Firebase Storage* zerbitzuaren bidez gordetzen dira. Zerbitzu edo tresna honek Windows-eko fitxategi sistemaren antza handia dauka hurrengo irudian ikus daitekeen antzera:

<input type="checkbox"/>		BoulderImage/	-	Carpeta	-
<input type="checkbox"/>		PostImages/	-	Carpeta	-
<input type="checkbox"/>		PostVideos/	-	Carpeta	-
<input type="checkbox"/>		Profile Images/	-	Carpeta	-

. 5.3 irudia *Firebase Storage* karpeten egitura

Kasu honetan ere ez denez bideorik igo *BoulderVideos* karpeta falta da eta esan bezala aplikazioaren bide bideo bat igo ondoren karpeta hau sortuko da. Hauetako karpeta bakoitzean objektu bakoitzak edukiko dituen fitxategi multimediaki daude gordeta. Lehen esan bezala fitxategi bakoitzak bi helbide edukiko ditu hurrengo irudian (*ikus 5.4 irudia*) ikus daitekeen modura. Gainera 5.2 irudian ikusi den bezala, *BoulderImages* atributuak dituen objektuek beste bi atributu dituzte gordeta. Hauek bi helbide motak dira:

<input type="checkbox"/>		1234327052019205536.jpg	21,8...	image/jpeg	27 may. 2...
<input type="checkbox"/>		1817052019162650.jpg	35,0...	image/jpeg	17 may. 2...
<input type="checkbox"/>		4317052019194527.jpg	38,1...	image/jpeg	18 may. 2...
<input type="checkbox"/>		4417052019194643.jpg	27,1...	image/jpeg	18 may. 2...
<input type="checkbox"/>		4517052019194704.jpg	33,9...	image/jpeg	18 may. 2...
<input type="checkbox"/>		4617052019194942.jpg	48,3 ...	image/jpeg	18 may. 2...
<input type="checkbox"/>		4817052019195606.jpg	32,4...	image/jpeg	18 may. 2...
<input type="checkbox"/>		4917052019202126.jpg	40,7...	image/jpeg	18 may. 2...



Nombre
[4317052019194527.jpg](#)

Tamaño
39.093 bytes

Tipo
image/jpeg

Fecha de creación
18 may. 2019 2:45:11

Fecha y hora de actualización
18 may. 2019 2:45:11

Ubicación del archivo

Ubicación de almacenamiento
gs://aprieta-84585.appspot.com/BoulderImage/4317052019194527.jpg

URL de descarga 1 revocar
<https://firebasesto...97bb-2bdb652a06e0>

. 5.4 irudia *Firebase Storage* fitxategien egitura

5.4 irudian ikus daiteke fitxategiak nola gordetzen diren eta gainera bakoitzari esleitzen zaizkion bi helbide motak, lehenengoa fitxategia gordetzeko eta bigarrena aplikazioan erakusten dena, azken finean kontrolatzaileak bistari pasatzen diona.

5.3. Kautoketarako datuak

Firebase-ek *Firebase Authentication* izeneko zerbitzuaren bidez, garatutako aplikazioarekin erabiltzaileak sortu, hauekin saioak ireki eta ixteko aukera emango digu. Tresna honen bidez aplikazioak dituen erabiltzaile guztiak gordeko dira bakoitzari identifikazio zenbaki bat emanez. Hurrengo irudian ikus daiteke nola gordetzen diren erabiltzaile hauek:

Identificador	Proveedores	Creado	Accediste a tu cuenta	UID de usuario ↑
ander2481996@gmail.com	✉	8 abr. 2019	1 jun. 2019	8wXFsw1pBRYXBVtChqY1Kqi14dy1
pablo.cuenca.petit@gmail.co...	✉	27 abr. 2019	17 may. 2019	ArqZEg9bx5dt7Eo1h0WnUVZ0ZUf2
xuban@gmail.com	✉	10 abr. 2019	8 may. 2019	FP1YJniJBpfU5aSZISAerL0DWZI2
misan128@hotmail.com	✉	18 may. 2019	25 may. 2019	aMycPQE0ogjLUd7Ci1X6CwbkaP2
chiquitodesanesteban@hot...	✉	17 may. 2019	17 may. 2019	qh00gfaWjOXJb7h48AFGds41nDK2

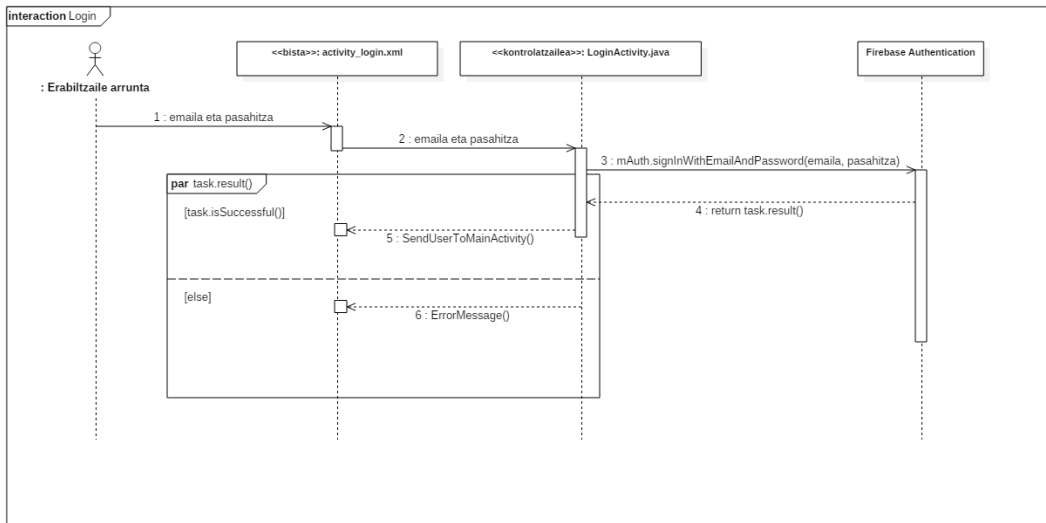
.. 5.5 irudia *Firebase erabiltzaileen kautotze egitura*

Irudian ikusten den moduan erabiltzaileen pasahitza ez da agertzen. Behin erabiltzaile bat sortzen dugunean pasahitza hash bidez enkriptatzen dira eta gainera sortzen den hash karaktere katea ez da ikusgai egongo ezta administratzailearentzat.

5.4. Sekuentzia diagramak

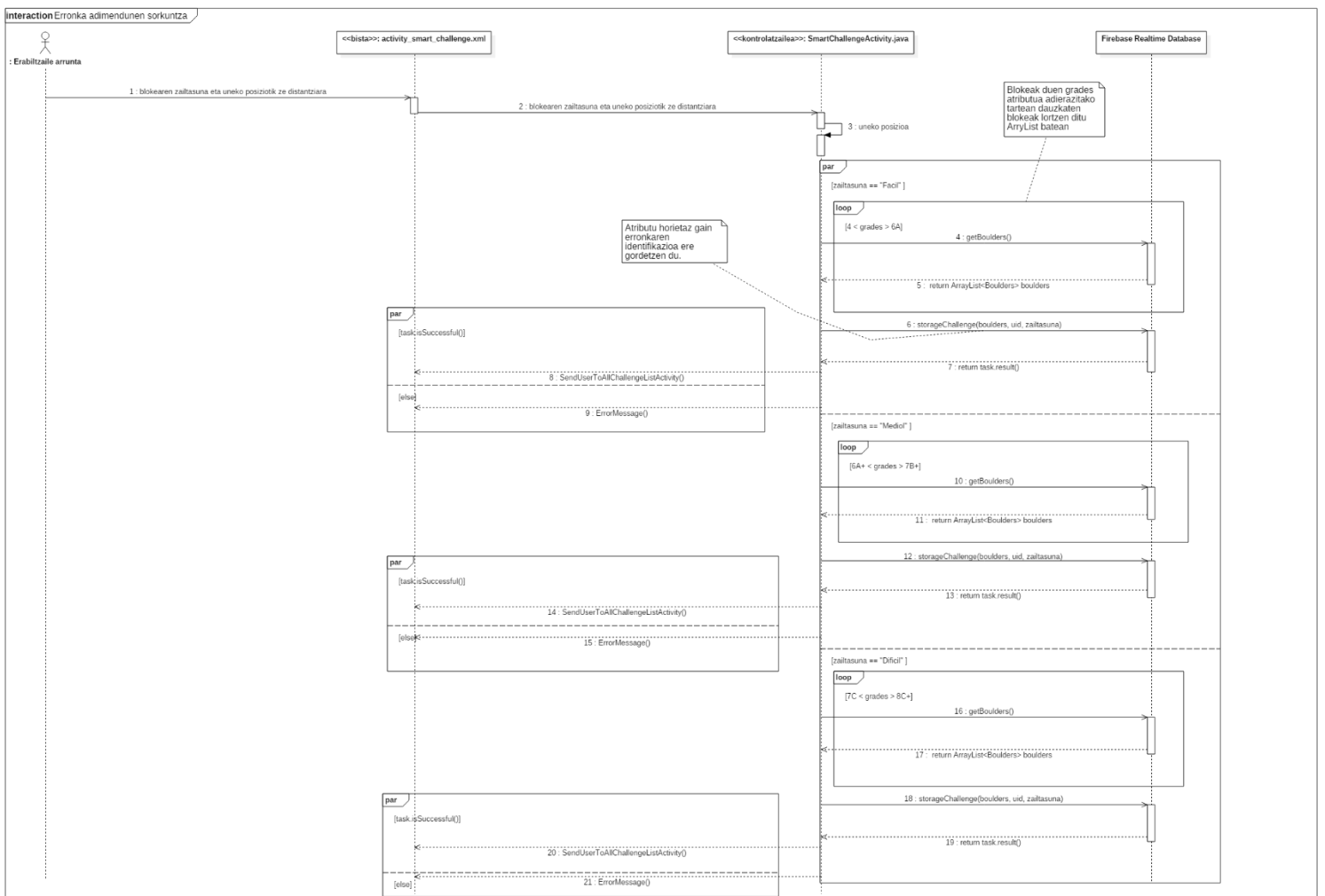
Erabilpen kasu bakoitzak bere sekuentzia diagrama eduki beharko luke, baina erabilpen kasu asko ditugunez, garrantzitsuenak egin ditugu. Hurrengo irudietan ikusi ahalko dira LogIn, Erronkak automatikoki sortu eta Blokeak sortu erabilpen kasuen sekuentzia diagramak, azken finean proiektu honetako erabilpen kasu garrantzitsuenetakoak baitira:

LogIn



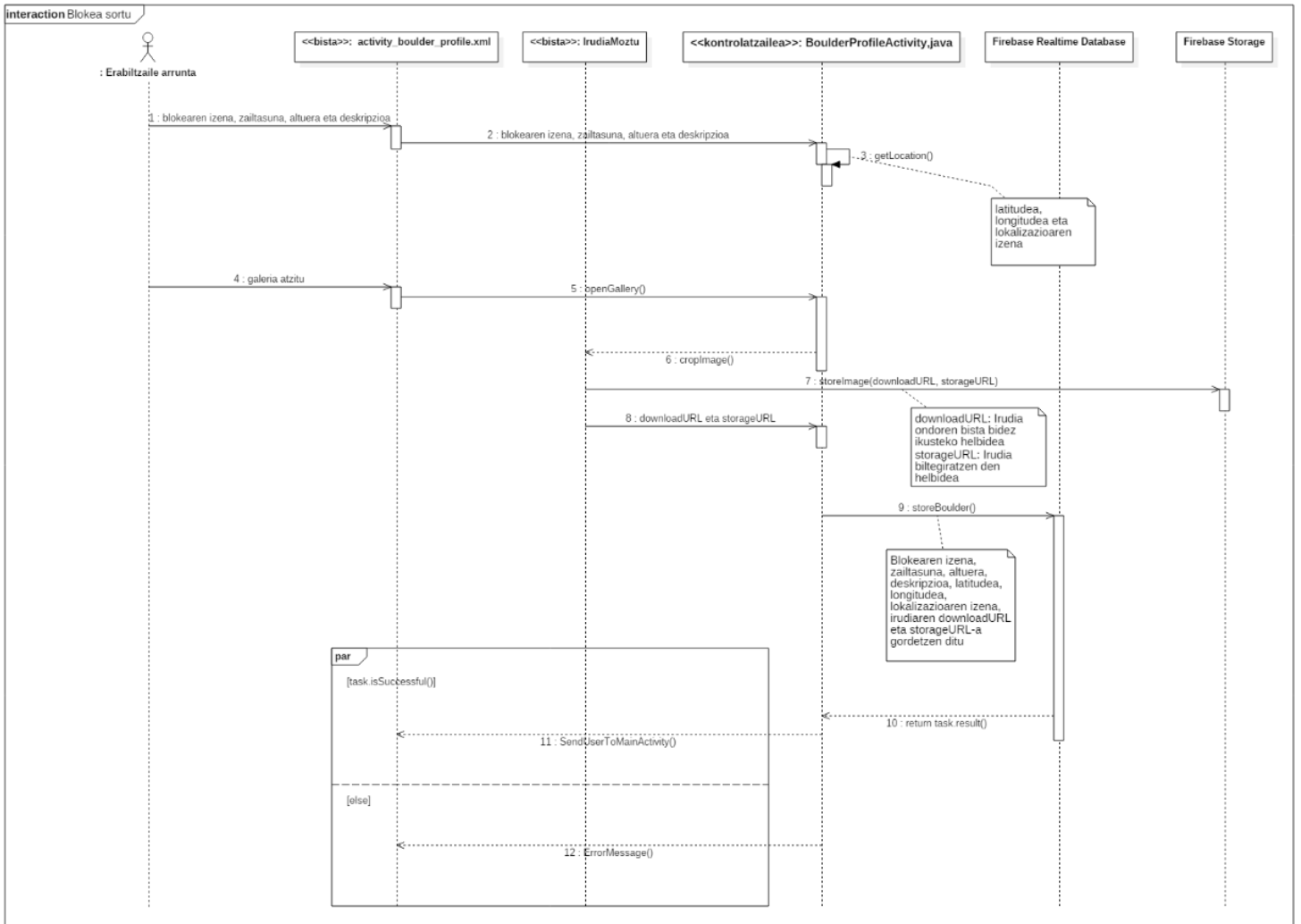
. 5.6 irudia Login sekuentzia diagrama

Erronkak automatikoki sortu



. 5.7 irudia Erronkak modu adimentsuan sortu sekuentzia diagrama

Blokea sortu



. 5.8 irudia Blokea sortu sekuentzia diagrama

6

Erabilitako teknologiak

Kapitulu honetan proiektuko produktua garatzeko erabili diren teknologiei buruz hitz egingo da, hauek produktuan zer eragin izan duten eta gainera azalduko da hauen funtzionalitate nagusia. Gainera teknologia hauen aukeraketa nola izan den eta hauen zergatia ere azaltzen saiatuko gara.

6.1. Produktuaren garapen ingurunea

Proiektuko produktuaren garapenarekin hasi baino lehen, memoria zehar aipatu den moduan, lehenagotik produktua hasita zegoen *Ionic* izeneko tresna bat erabiliz. Tresna hau *JavaScript* lengoaian idatzia dagoen arren, proiektuarekin hasi baino lehen *Android Studio*-rekin alderatu zen. Nahiz eta *Ionic*-en bidez bi sistema eragileentzako aplikazioa sortu daitekeen Interneten bilatu zitekeen informazioa ez zein ain zabala *Android Studio*-rekin alderatuta. Gainera denbora aldetik hobeto ibiltzeko eta hasieran bezeroarekin ezarritako funtzionalitateak guztiz betetzeko ideia hobea iruditu zitzaigun hasiera batean behintzat *Android*-erako soilik garatzea produktua.

Bestalde, datu-baseari dagokionez, unibertsitatean erlazional motakoak soilik ikusi ditugunez, ideia ona iruditu zitzaigun ez erlazional bat erabiltzea eta horrela gauza berriak ikastea. Gainera datu-base erlazional batean taulak sortu eta dagoeneko duten estruktura edo eskema jarraitu behar da. Gerora taula hauetan aldaketa bat egin nahi badugu, lehenik eta behin eskema eta taula biak aldatu beharko ditugu. Aldiz ez erlazionaletan produktua garatzen goazen heinean datu-basean aldaketarik egin behar bada ez da horrelako arazorik aurkeztuko, malguagoa baita aspektu honetan.

Produktuak domeinu batean kokatua dagoen datu-base batekin konexioa mantenduz funtzionatuko du. Hau da, Interneteko konexio bidezkoak dira bertan programatuak egongo diren funtzio guztiak, beraz beharrezko zen datu-base hau zerbitzari ez lokal batean kokatua egotea.

Aukera asko aztertu ondoren, *Firebase* izeneko aplikazioak garatzeko plataforma aurkitu genuen. Honek datu-base ez erlazionalak erabiltzen dituenek ideia ona iruditu zitzaigun plataforma hau erabiltzea eta honez gain jadanik Interneten kokatua zegoen.

6.2. Teknologia eta Lengoaien azalpenak

Kapitulu honetako atal honetan erabili diren teknologia eta lengoaiak zer diren eta gure proiektuaren barruan ze eragin izan duten azalduko dugu.

6.2.1 Android Studio

Android Studio *Android* aplikazioen garapenerako ingurune integratu bat da. *IntelliJ*-ak sortua eta Google berak garatuta dago Java lengoaiari erabiliz batik batik. Hasiera batean *Eclipse* bera erabiltzen zen *Android* erako aplikazioak sortzeko, gaur egun ordea hau *Android Studio*-ren lana da.

Tresna honekin Java-rekin objektuei orientatuko programazioa gauzatzeaz gain, aplikaziorako balioko digun interfaze grafikoa sortzeko XML markaketa lengoaiari ere

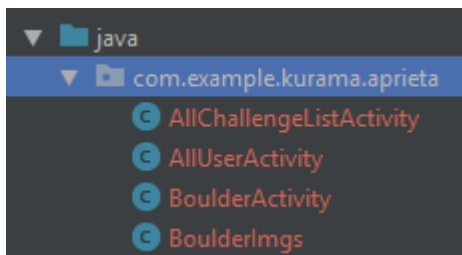
erabiltzen du. Interfaze hauek garatzeko garaian hutsetik hasi edo tresna berak emandako txantilo batzuk erabili ditzakegu.

Proiektu honetako aplikazioaren estruktura nagusia eratzeko balio izan du software honek. Azken finean android-eko aplikazio bat garatzea izan denez proiektuaren helburu nagusia, eta honen MVC arkitektura jarraitzeko asmotan tresna egokiena zela iruditu zaigu. Gainera gure aplikazioko datu guztiak gordetzeko eta erabiltzaileen kautotzea *Firebase* plataformaren bidez egin behar dugunez *Android Studio* erabiltzeak abantaila handiak ekartzen dizkigu biak modu oso erraz batean elkar daitezkeelako (Biak Google-enak dira).

6.2.2 Java

Java teknologia programazio lengoaia eta makina birtualetan banatu daiteke, baina gure kasuan proiektu honetarako Java programazio lengoaia modura hitz egingo dugu. Zehatzago izanda, asmo orokorretara eta objektuei zuzendutako lengoaia bat da.

Proiektu honetako lengoaia nagusia izan da. Gure aplikazioaren arkitekturaren Eredu eta Kontrolatzaileak sortzeko garaian lengoaia hau erabili behar izan da, hauek java klaseak baitira, hurrengo irudian ikus daitekeen antzera, (Activity hitza duten klaseak Kontrolatzaileak dira eta besteak Ereduak):



.. 6. 1 irudia Java klaseak

6.2.3 XML

XML, *eXtensible Markup Language*, markaketa lengoaia bat da, zehatzago esanda meta-lengoaia bat da eta beste markaketa lengoaiak definitzea ahalbidetzen du. W3C garatua da eta datuak modu ulergarri batean gordetzeko erabiltzen da.

Gure kasuan *Android Studio*-ko interfazeen diseinuak garatzeko erabili dugu. *Android*-eko diseinuko elementu bakoitza XML etiketetan elementu modura gordetzen dugu, horrela adibidez botoi bat nahi badugu gure diseinuan XML etiketa baten bidez adieraziko dugu.

Etiketa hauetan botoi edo beste edozein elementuek (bista) zein nolako ezaugarriak adierazteko *Android Studio* berak definituak dituen atributu batzuk erabili behar dira, nahiz eta gerora liburutegi berriak instalatzen baditugu atributu gehigarri batzuk ere eduki ditzakegu.

Adibide moduan diseinuan botoi bat gehitzeko jarri beharreko XML lerroak adieraziko ditugu hurrengo irudian:

```
<Button
  android:id="@+id/login"
  android:layout_width="118dp"
  android:layout_height="wrap_content"
  android:layout_below="@+id/login_password"
  android:layout_alignParentStart="true"
  android:layout_marginStart="24dp"
  android:layout_marginLeft="20dp"
  android:layout_marginTop="60dp"
  android:background="@drawable/button"
  android:padding="15dp"
  android:text="@string/login_log_button" />
```

. 6. 1 irudia XML kodea

6.2.4 Firebase

Firebase Google-en eskuetan dagoen mugikorreko zein web-eko aplikazioak garatzeko plataforma bat da eta hodeian kokatua dago Google Cloud Platform erabiliz. Plataforma honen bidez oso erraz gauzatzen da *Android* ekin edo beste tresnekin sortzen ari garen proiektuak sinkronizatzea.

Firebase-ek lau ezaugarri garrantzitsu ditu eta ondoren azalduko dira hauek:

1. Proiektuko datuak modu erraz batean **sinkronizatzea** konexiorik administratu gabe.
2. **Plataforma anitzak erabiltzeko tresnen erabilpena** erabiltzen du. Horrela bai web plataformak eta mugikorretako aplikazioak sortzeko aukera ematen du. Mugikorreko aplikazioen gainean iOS eta *Android* sistema eragileak onartzen ditu.
3. **Google-ren infraestruktura** erabiltzen duenez edozein motako aplikazio garatu daiteke, maila handiko zein txikikoak.
4. **Zerbitzari gabeko** garapena eskaintzen du, honek ez du esan nahi ez dagoenik zerbitzaririk baizik eta *Firebase* osoa Google-eko zerbitzarian dago muntatu eta beraz gu garatzaileok ez dugu horretaz arduratu behar.

Firebase-ek ezaugarri horietaz gain beste hainbat zerbitzu eskaintzen ditu baina hemen proiektu honetarako erabili diren hiru garrantzitsuenak azalduko ditugu:

- ***Firebase Auth***, erabiltzaileak kautotutzeko zerbitzu bat da eta bezeroaren aldeko kodea erabiliz gauzatzen du hau. *Firebase Auth* ek eskaintzen dituen backend, SDK eta beste liburutegiei esker egin daiteke hau. Gainera bi motako erabiltzaileak gorde daitezke:
 - *Firebase* bertan gordetako erabiltzaileak, beraien emaila eta pasahitzarekin

- Google, Github, Twitter, Facebook eta beste hainbat zerbitzuetako kontuekin sortutako erabiltzaileak

Zerbitzu honek autentifikazio-sistemak sortzea errazten du, erabiltzaileen sarrera eta segurtasuna hobetuz. Horri esker, garatzaileak ez du kautotze metodo klasikoak garatzeaz kezkatu behar eta kasu honetan horrela izan da.

- **Realtime Database**, *Firebase*-en kokatuta dagoen denbora errealeko datu-base bat da, eta JSON zuhaitz forman sortua dago. Zerbitzu honek garatzaileentzako API bat dauka, garatutako aplikazioak *Firebase*-eko hodeiarekin modu erraz batean sinkronizatzeko.
- **Firebase Storage**, fitxategi karga eta deskarga seguruak eskaintzen ditu *Firebase* erabiliz sortu diren aplikazioetarako, sarearen kalitatea kontuan hartu gabe. Garatzaileak irudiak, audioak, bideoak edo beste erabiltzaile batzuen edukia gordetzeko erabil dezakete. Gainera *Firebase Storage* *Google Cloud Storage* bidez babestua dago.

6.2.5 JSON

JSON, *JavaScript Object Notation* datuen elkartruckerako formatu arineko testua da. Notazio honek sei motako datuak gordetzeko ahalmena dauka:

- Zenbakiak; bai positibo zein negatiboak eta hamartarren kasuan puntu batez separatuak
- String edo karaktere kateak
- Boolearrak
- Balio hutsak edo null
- Arrayak
- Objektuak, adibidez: {"izena": "Ander", "abizena": "Lopez"}

Adibide zehatzago baten bila hurrengo irudian proiektuko datu-basearen zati txiki bat ikusiko da JSON en egitura zertxobait hobeto ulertze arren nahiz eta aurrerago datu-basearen azalpen sakonago bat egingo da:

```

"Boulders" : {
  "-016614551654332707-7848568975925446" : {
    "BoulderImages" : {
      "-Lf6g6PVlyZF0QKczKt9" : {
        "images" : "https://firebasestorage.googleapis.com/v0/b/aprieta-84585.appspot.com/o/BoulderImage3",
        "profilestorage" : "gs://aprieta-84585.appspot.com/BoulderImage/1817052019162650.jpg"
      }
    },
    "boulderkey" : "-016614551654332707-7848568975925446",
    "description" : "Aguanta!",
    "grade" : "6B",
    "height" : "1",
    "latitude" : "-0.16614551654332707",
    "location" : "Ecuador, Distrito Metropolitano de Quito, Quito",
    "longitude" : "-78.48568975925446",
    "name" : "Dinamic",
    "numberOfVots" : "0",
    "rate" : "0",
    "userid" : "ArqZEg9bx5dt7Eo1h0WnUVZOZUf2",
  },
}

```

. 6.2 irudia

Datu-basea

6.2.6 StarUML

StarUML UML-n oinarraitua dagoen software modelaketa gauzatzeko tresna bat da. Proiektu honetan memoriako *"Eskakizun bilketa"* izeneko kapituluari agertzen diren *Domeinu eredia* eta *Erabilpen kasuak* eta *"Diseinua"* izeneko kapituluko *Sekuentzia diagramak* egiteko erabili da.

7

Implementazioa

Kapitulu honetan proiektuko aplikazioa garatzerakoan egin diren inplementazio garrantzitsuenei buruz hitz egingo da. Batez ere MVC arkitekturako maila bakoitzaren inplementazioa ikusiko dugu. Kontrolatzailearen kasuan *Firebase* plataformarekin egin diren eragiketak nola inplementatu diren azalduko da, azken finean aplikazioa eta datu base edo zerbitzariarekin egindako garapena ikuste arren.

7.1. Bista

Garatutako aplikazioak dituen bistak esan bezala XML bidez sortzen dira. XML fitxategi hauetan *Android Studio*-k jadanik definituak dituen defektuzko liburutegien bidez XML-ko elementuak erabili daitezke aplikazioko bistak sortzeko.

Bista hauetan garrantzitsuena beti lehenengo elementua izango da, honek definituko baitdu bistaren egitura nagusia, honi *Layout* deritzo. Aplikazio honetarako gehien erabili den layout mota *RelativeLayout* izan da. Nahiz eta beste hainbat layout mota izan: *LinearLayout*(*vertical* eta *horizontal*), *ConstraintLayout*,... malgutasun handiena ematen duena *RelativeLayout* da.

Beste ezaugarri garrantzitsu bat elementua aurkitzen diren atributuak dira. Bistako elementu bakoitzak bere atributu propioak edukitzeaz gain, denek komunean dituzten atributu batzuk ere edukiko ditugu. Hauen artean, elementua identifikatzeko *id* bat eta hauen tamaina adierazteko erabiltzen diren *layout_width* eta *layout_height*. Gure kasuan bista definitzen duen lehenengo elementuaren atributu hauek *match_parent* izango dira, hau da, mugikorraren pantaila osoa hartuko duela (bistak) adierazten du kasu honetan. Bistako elementu batek atributu hauetan balio hau baldin badu, dagoen layout tamainaren tamaina bera lortuko du horregatik normalean gainontzeko elementuetan *dp*(*density pixel*) izeneko unitatea erabiltzen da atributu hauek definitzeko. Hurrengo irudian ikusi ahalko dugu bista baten lehenengo elementua eta hauen atributuak:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ProfileActivity"
    android:background="@color/colorAccent">
```

.. 7.1 irudia

RelativeLayout inplementazioa

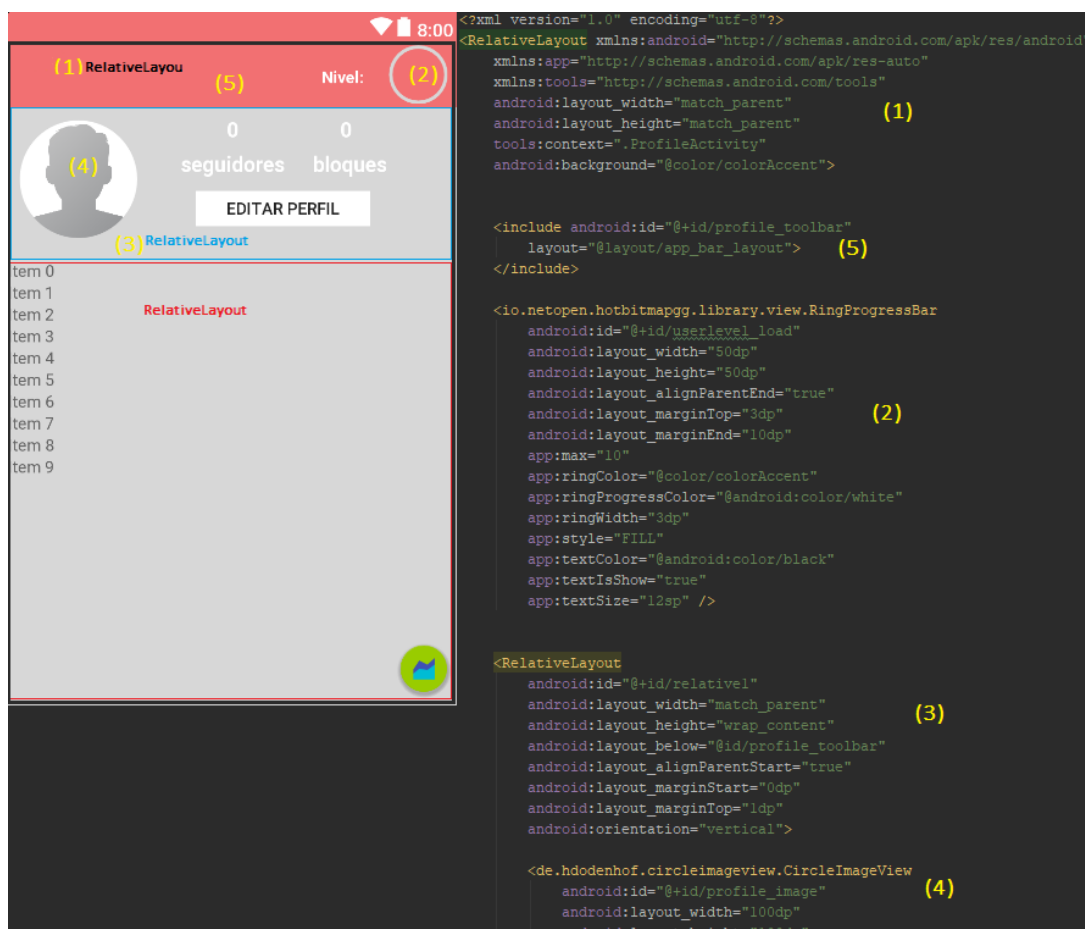
Bista bat osatzen duten elementuen artean esan bezala lehenengo elementua layout bat izango da, baina honen barnean doazen gainontzeko elementuen artean beste layout bat ere aurkitu dezakegu, hau bista modu ordenatuago eta txukunago batean antolatzeko erabiltzen da, bista barruan sekzio desberdinak sortuz.

Gainontzeko elementuak gehienak *Android Studiok* daukan defektuzko liburutegietatik etortzeaz gain, memoriako beste kapitulu batean aipatu bezala, Interneteko beste plataformetan aurkitzen diren liburutegiak ere erabili daitezke lehenik eta behin proiektuan inplementatzen baditugu. Horretarako *Android Studio*-ren dependentzietan liburutegi berri honen inplementazioa adierazi behar dugu. Adibide bezala aplikazioko erabiltzaile baten profileko bi elementuri buruz hitz egingo dugu: Erabiltzailearen profileko argazkia zirkulu batean jartzea eta bere maila (7.3 irudian "Nivel" deritzona) adierazten duen zirkulua. Hurrengo irudiak ikus daiteke nola sartu liburutegi hauek *Android Studio*-ren dependentzietan:

```
dependencies {
...
implementation 'de.hdodenhof:circleimageview:2.2.0'
implementation 'com.github.HotBitmapGG:RingProgressBar:V1.2.2'
...
}
```

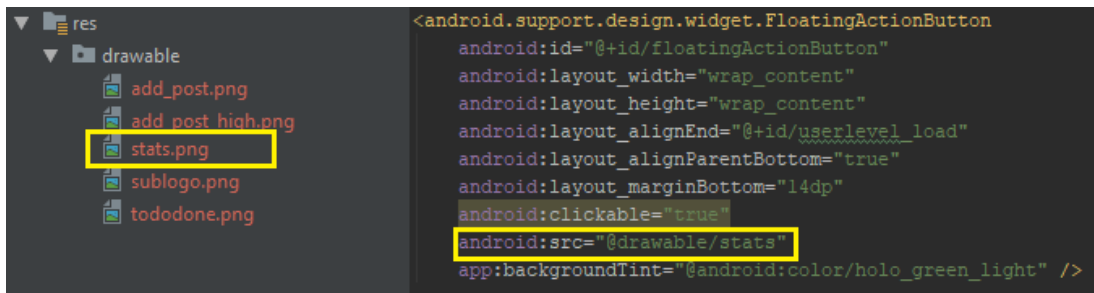
. 7.2 irudia *Liburutegian inplementazioa proiektuan*

Behin liburutegiak gehituta ditugula, elementu berri hauek *Android Studio*-k defektuz ekartzen dituen elementuak bezala erabili daitezke, nahiz eta bakoitzak bere atributu propioak ere edukiko dituen. Hurrengo irudian erabiltzaile baten profileko bista bat ikusiko dugu. Bista honetako elementu bakoitza inplementatutako kodearekin lotuz:



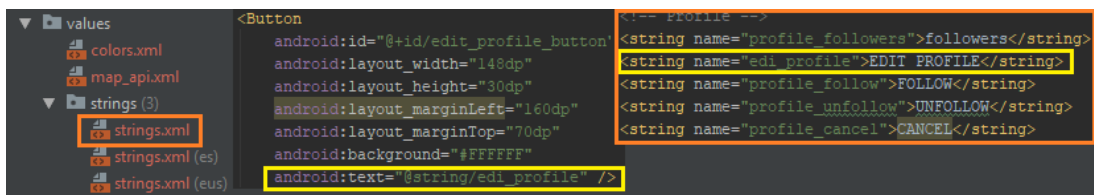
. 7.3 irudia *Bista baten inplementazioa*

Honetaz gain bistetan azaltzen diren ikonoak aurretik irudia karpeta batean gorde eta ondoren elementuaren atributuen bitartez ekarri behar dugu bistara, hurrengo irudian agertzen den moduan:



. 7.4 irudia Bistako elementuetan irudia ezarri

Bistekin amaitzeko, aplikazioak testuzko elementu asko edukiko dituzenez eta hizkuntza bakarreko aplikazio bat ez denez, bertan agertzen diren testuzko esaldi guztiak aplikazioak edukiko dituen hizkuntzetan idatzi beharko ditugu. Horretarako *values* -> *string* izeneko karpeta bat sortu behar da eta bertan hizkuntza bakoitzerako strings.xml fitxategi bat. Bistako elementu batek testua badu, idatzi beharreko testua bertan jarri beharrean, hizkuntza bakoitzeko strings.xml fitxategietan izen bereko <string> etiketetan idatziko da, baina strings.xml fitxategi bakoitzean dagokion hizkuntzan, ondoren testua nahi dugun lekuan <string> etiketa honi erreferentzia egingo da. Horrela mugikorrean defektuz dugu hizkuntzan egongo da aplikazioa. Argiago ulertzeko hurrengo irudian ikusi ahalko dugu nola egiten den hau:



. 7.5 irudia Bistako elementuko testuen hizkuntza

7.2. Kontrolatzailea

Kontrolatzailea da MVC arkitekturako implementazio gehien jaso duen atala, azken finean aipatu bezala bista eta ereduekin lotuta egoteaz gain, *Firebase*-rekin egin beharreko eragiketa eta mugimendu guztiak egiten dituen arduradun nagusia da. Hau esanda maila honen implementazioa bi zatitan banatuko dugu. Lehenik eta behin kontrolatzailea bera nola dagoen implementatua eta dagokien bistekin egiten den elkarrekintzen implementazioa eta beste aldetik *Firebase* zerbitzuarekin egiten dituen eragiketen implementazioa.

7.2.1 Kontrolatzailearen implementazio nagusia

Proiektu honetako aplikazioaren kontrolatzaileaz hitz egiten dugunean *Android Studio*-k sortzen dituen Activity-*ei* buruz ari gara hizketan. Activity hauek java klaseak dira azken finean, baina klase hauek lehen aipatutako XML bistei lotuak daude. Derrigorrezkoa da

Activity klase bat bista bati lotua egotea eta beraz ondoren bista honek dituen elementuen instantziak lortzea horrela ondoren hauek egin behar dituzten akzioak kontrolatu ahal izateko.

Hasteko Activity klase bakoitzak *AppCompatActivity* klasea heredatu behar du bistako elementuen instantziak lortzeko eta baita ere metodo garrantzitsu batzuk edukitzeko. Metodo hauek garrantzitsuenak *onCreate(Bundle savedInstanceState)* edo *onStart()* dira. Hauetatik lehenengoa (*onCreate(...)*) derrigorrezkoa izango da, horrela Activity batean sartzen garenean metodo honen barnean egiten diren instantziak, elementuen akzioak eta metodo deiak gauzatu dira. Gainera metodo honen barnean adierazi behar da Activity-aren bista zein den, horrela bista aplikazioan kargatu ahal izateko, irudian ikusi daitekeen moduan:

```
public class ProfileActivity extends AppCompatActivity {
    private CircleImageView profileImage;
    private Button settingButton;
    private RingProgressBar userlevel;
    private FloatingActionButton statsButton;
    ...
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //BISTARI DEITZEKO
        setContentView(R.layout.activity_profile);

        //BISTAKO ELEMENTUEN INSTATZIA BATZUK
        profileImage = (CircleImageView)findViewById(R.id.profile_image);
        settingButton = (Button) findViewById(R.id.edit_profile_button);
        userlevel = findViewById(R.id.userlevel_load);
        statsButton = findViewById(R.id.floatingActionButton);
        ...

        //BISTAKO ELEMENTUEN AKZIO BAT
        settingButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                SendUserToSettingsActivity();
            }
        });

        //METODO BATI DEIA
        DisplayUserPosts();
    }
    ...
}
```

.. 7.6 irudia Kontrolatzailearen hasierako inplementazioa

Activity batetik bestera pasatzeko metodoa da 7.6 irudian ikusten den *SendUserToSettingsActivity()*; bertan ze Activity-tik zein Activity-ra joan nahi dugun adierazi beharko dugu, eta behin hau eginda, momentuan gauden Activity-a amaitzea nahi badugu *finish()*; metodoa erabili beharko dugu. Metodo honen bidez hurrengo Activity-ra pasatzean

aurrekoa guztiz amaituko da eta beraz bertara iristeko egin beharreko pausuak berriz egin beharko dira. Aldiz metodo hau ez badugu erabiltzen eta Activity berritik atzera egiten badugu hasieran geunden Activity-ra itzuliko gara. Hurrengo kode zatian ikus daiteke metodo hau:

```
private void SendUserToSettingsActivity() {
    Intent settingIntent = new Intent(ProfileActivity.this, SettingsActivity.class);
    startActivity(settingIntent);
    finish();
}
```

. 7.7 irudia Bistatik bistara mugitzeko metodoa

7.2.2 Kontrolatzailea eta Firebase

Kontrolatzailearen azkeneko puntu honetan, honen eta *Firebase*-ek eskaintzen dituen zerbitzuen arteko interakzioen implementazioei buruz hitz egingo dugu. Memoriako beste ataletan aipatu bezala *Firebase*-en hiru zerbitzu garrantzitsuenak erabili dira; *Firebase Authentication*, *Firebase Realtime Database* eta *Firebase Storage*. Banan-bana *Android Studio*-n gure aplikaziorako *Firebase*-en hiru fazeta hauek nola implementatu diren ikusiko da orain. Baina lehenik eta behin bista atalean egin bezala kasu honetan ere *Firebase*-en zerbitzuak erabili behar baditugu bakoitzak dituen liburutegiak dependentzietan sartu behar ditugu, horrela *Firebase*-ek eskaintzen dituen API-ak erabili ahal izango ditugu:

```
implementation 'com.google.firebase:firebase-database:15.0.0'
implementation 'com.google.firebase:firebase-auth:15.0.0'
implementation 'com.google.firebase:firebase-storage:15.0.0'
```

. 7.8 irudia *Firebase* zerbitzuen liburutegiak

7.2.2.1 *Firebase Authentication*

Firebase Authentication zerbitzua aplikazioaren erabiltzailea kautotzeko erabiliko dugu. Bere funtzionalitate nagusiak erabiltzaileak sortzea eta hauen saioak ireki eta ixtea den arren, aplikazioan momentu oro zein erabiltzaile dagoen konektatua eta bere identifikazio lortzeko ere erabiltzen da.

Modu oso erraz batean implementatzen da izan ere *Firebase*-ek eskaintzen duen API-a oso intuitiboa eta erabil erraza egiten da.

Lehenik eta behin erabiltzaile baten sorrera ikusiko dugu. Hurrengo irudian agertzen den moduan, nahikoa da email bat eta pasahitza erabiltzea erabiltzaile bat sortzeko, baina lehenik eta behin *FirebaseAuth* objektuaren instantzia bat sortu behar dugu:

```
private FirebaseAuth mAuth;

mAuth = FirebaseAuth.getInstance();

mAuth.createUserWithEmailAndPassword(email, password)
```

. 7.9 irudia Erabiltzaile baten sorrera

Gerora esan bezala *FirebaseAuth*-en instantzia honekin momentu oro saioa irekia duen erabiltzailearen identifikazioa lortuko dugu *mAuth.getCurrentUser().getUid()* erabiliz.

Saio bat irekitzeko ere oso sinplea da, azken finean lehen egindako instantziarekin nahiko dugu *mAuth.signInWithEmailAndPassword(email, password)* metodoa erabiltzearekin eta berriz saioa ixteko, *mAuth.signOut()* .

Argiago ikuste arren, *mAuth* instantzia lortzen dugunean eta saioa ireki gabe baldin badago *mAuth* objektu instantziatuak null emango du. Aldiz saioa irekitzen dugunean hau informazioz betetzen da eta horietako bat erabiltzailearen identifikazio zenbakia da. Saioa ixtean berriz ere informazio guztia ezabatu egiten da objektutik. Honekin erabiltzaile batek saioa irekita duen edo ez jakin dezakegu momentu oro.

7.2.2.2 Firebase Realtime Database

Firebase Realtime Database erabiliko dugu proiektuko datu base bezala. Bertan gordeko dira aplikazioak behar dituen datu guztiak eta beraz edozein datu baserekin bezala CRUD eragiketak egin beharko dira. Atal honetan eragiketa hauen inplementazioei buruz hitz egingo dugu. Baina lehenik eta behin datu basearen erreferentzia egingo duen objektu bat instantziatu behar dugu. Kasu honetan datu baseko erabiltzaileen atalari erreferentziatuko dion objektu baten instantzia erakutsiko da:

```
private DatabaseReference UserRef;  
UserRef = FirebaseDatabase.getInstance().getReference().child("Users");
```

.. 7.10 irudia Datu basearekiko erreferentzia

Instantzian agertzen den "Users" testuak datu basean objektu horrek duen izena adierazten du.

- **Datu basean datuak sortu**

Datu basean datu berriak sortzeko lehenik eta behin komenigarria da *HashMap* bat sortzea sartu behar ditugun datuak atributu asko baldin baditu. Erabiltzaile bat sortzerakoan adibidez izena, profileko irudia, herrialdea, emaila,... atributuak dituzenez, lehenik eta behin hauek *HashMap* batean gorde eta ondoren datu basera bidaltzea izango da ideia nagusia. Hurrengo irudian ikusten da hau nola egin:

```
HashMap userMap = new HashMap();  
userMap.put("username", username);  
userMap.put("fullname", fullname);  
userMap.put("country", country);  
userMap.put("status", "");  
UsersRef.child(currentUserID).updateChildren(userMap)
```

.. 7.11 irudia Datu basearekiko erreferentzia

Adibide honekin hurrengo datuak sortuko dira gure datu basean:



.. 7.12 irudia Datu basea sortutako datu berriekin

- **Datu basetik datuak irakurri**

Datu basetik datuak irakurtzerakoan bi kasurekin aurkitzen gara. Lehenengo kasua datu basetik aldiro datu bat soilik irakurtzea da. Hau egin nahi badugu nahikoa da hasierako atalean adierazitako instatziak baliatzea. Instanzia honekin datu basean entzuten jarriko gara *Listener* metodo baten bidez eta horrela entzuten ari garen datuak eta hauek dituzten atributuak atzitu ahal izango ditugu, hurrengo irudian azaltzen den moduan:

```
UserRef.child(currentUserID).addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot)
    {
        if(dataSnapshot.exists())
        {
            if(dataSnapshot.hasChild("username")) {
                username = dataSnapshot.child("username").getValue().toString();
            }
        }
    }
});
```

.. 7.13 irudia Datu basetik datu bat irakurri

Azken finean bide bat jarraitzea bezala hartu daiteke ideia bat egiteko, Users → Uneko erabiltzailean jartzen dugu gure erreferentzia eta bertan lortu nahi ditugun datuak dituen atributuen izena jarriz lortuko dugu nahi dugun datua.

Aldiz aldiro datu asko lortu nahi baditugu MVC arkitekturako Eredu mailaren laguntza beharko dugu. Hau egiteko *Query* objektu bat sortu behar dugu aurreko datu basearen erreferentzia egiten duen instanzia erabiliz.

```
Query searchPeople = UserRef.orderByChild("username").startAt(searchUser).endAt(searchUser + "\uf8ff");
```

.. 7.14 irudia Query bat sortu

Query bat sortuta, erabiltzaile bakoitzetik pasatzen da “username” atributuak “searchUser” string-ak duen balioa bilatzen.

Behin erabiltzaileak aurkituta banan-banan lortzen du hauen datuak Eredu-en laguntzaz. Hurrengo irudian ikus daiteke hau:

```
FirestoreRecyclerOptions<FindFriends> options=new
FirestoreRecyclerOptions.Builder<FindFriends>().setQuery(searchPeople,FindFriends
.class).build();

FirestoreRecyclerAdapter<FindFriends, FindFriendsViewHolder>
firebaseRecyclerAdapter = new FirestoreRecyclerAdapter<FindFriends,
```

```
FindFriendsViewHolder>(options) {
    @Override
    protected void onBindViewHolder(@NonNull final FindFriendsViewHolder holder,
int position, @NonNull FindFriends model) {
        final String ProfileKey = getRef(position).getKey();
        holder.username.setText(model.getUsername());
        holder.fullname.setText(model.getFullname());
    }
}
```

.. 7.15 irudia Erabiltzaileen datuak lortzen

FirebaseRecyclerView azken finean datu basetik lortutako datuak lista batean jartzen ditu. Kasu honetan erabiltzaile lista bat sortutako da eta horretarako lehen esandako baldintzak betetzen dituen erabiltzaileak lortu beharko ditugu. Hau egiteko 7.15 irudian ikusten den moduan *setQuery* zatian Query-a eta Eredua izango den java klasea pasatzen diogu. Eredua klasea gero aztertuko da hobeto, baina azken finean erabiltzaileagandik lortu nahi ditugun datuen getterrak edukiko ditu.

Ondore azaltzen den *onBindViewHolder* metodoaren bidez bistako elementuetan jarriko ditugu ereduak lortu ditugun datuak.

- **Datu baseko datuak eguneratu**

Hasierako instantziarekin egingo da hau ere. Instantzia honekin eguneratu nahi dugun daturaino joango gara, adibidez uneko erabiltzailearen erabiltzaile izena “anderlopz” izenetik “ander” izenera aldatu nahi badugu nahikoa da aldatu nahi den atributuraino bidea egin beheko irudian agertzen den moduan, eta behin posizio horretan balio berria ezarri.

```
UserRef.child(currentUserID).child("username").setValue("ander");
```

.. 7.16 irudia Datu baseko datuen eguneraketa

- **Datu baseko datuak ezabatu**

Azkeneko kasu honetan ere ezabatu nahi dugun datuaren bidea egin behar dugu. Aurreko kasuaren antzera erabiltzaile baten erabiltzaile izena ezabatu nahi badugu atributu honetara iritsi eta hau ezabatuko dugu.

```
UserRef.child(currentUserID).child("username").removeValue();
```

.. 7.17 irudia Datu baseko datuen ezabaketa

7.2.2.3 Firebase Storage

Azkenik *Firebase Storage* azalduko dugu. Hau aplikazioak erabiliko dituen fitxategi multimedialak gordetzeko erabiliko dugu eta beraz hiru eragiketa mota edukiko ditu: fitxategiak igo, lortu eta ezabatu. Baina lehenik eta behin beste *Firebase* zerbitzuekin egiten den antzera *Firebase Storage* instantzia bat lortu behar dugu:

```
private StorageReference postImageReference;
private FirebaseStorage storage;

postImageReference = FirebaseStorage.getInstance().getReference();
storage = FirebaseStorage.getInstance();
```

.. 7.18 irudia *Firebase Storage* instantzia

Ikusten den moduan eta memorian diseinu atalean aipatu bezala fitxategi multimediak igotzerakoan bi helbide mota adierazi behar dira, lehenengoa (*postImageReference*) aplikazioaren bistetan, kasu honetan, irudiak bistartzeko balioko du eta bestea (*storage*) aldiz fitxategia biltegitatu beharreko helbidea izango da.

postImageReference-k *Firestore Storage* biltegitartzeko sistemaren errora apuntatzen du. Beraz fitxategia gorde ahal izateko erroan karpeta berri bat edo dagoeneko izen hau duen karpeta batean jarri beharko dugu, gure kasuan "PostImages" izena edukiko du. Behin karpeta adierazita bere izena erabakitzen da eta azkenik fitxategia igotzen da *putFile(fitxategia)* metodoaren bidez.

Behin fitxategia igo dugula honen biltegitatze bidea lortuko dugu, hau ondoren baliagarria izango baita fitxategia ezabatu nahi dugunean. Hona hemen pauso hauek erakusten dituen irudi bat:

```
StorageReference filePath1 =
postImageReference.child("PostImages").child(ImageUri.getLastPathSegment()+postRandomName
+".jpg");

filePath1.putFile(ImageUri)
storage.getReferenceFromUrl(task.getResult().getStorage().toString
```

.. 7.19 irudia *Firestore Storage fitxategiak igo*

Fitxategi hauek bistaratu nahi baditugu gure bistetan aurkitzen diren elementuak kontrolatzailetik kargatuko ditugu. Irudien kasuan *ImageView* motako elementu bat behar dugu. Behin elementu hau kontrolatzailean instantziatua dugunean *Firestore Storage*tik lehenengo helbidea erabiliz irudia, lehen aipatutako elementuan kargatuko dugu liburutegi baten bidez, hurrengo irudian agertzen den moduan. Suposatuz *image* atributuan *Firestore Storage*-en gordetako irudiaren helbidea dugula:

```
private ImageView PostImage;

PostImage = (ImageView) findViewById(R.id.postEditImage);

Picasso.get().load(image).into(PostImage);
```

.. 7.20 irudia *Firestore Storage fitxategiak lortu*

Picasso motako objektuak *Firestore Storage*-ko helbidetik irudia lortu eta *PostImage* izeneko *ImageView* bistako elementuan kargatzen du.

Azkenik fitxategi bat ezabatzeko bigarren helbidea erabiliko dugu "storage". Hau egiteko nahikoa da fitxategia igo dugun momentuan gorde dugun "storage" helbidea atzitzea datu basetik, ondoren honen helbidetik erreferentzia bat lortzea eta hau *Firestore Storage* erreferentzia bati esleitzea. Ondoren *delete()* metodoa erabiltzen da erreferentzia honen gainean fitxategia ezabatzeko:

```
private FirebaseFirestore storage;
private StorageReference postStoreRef;

storage = FirebaseFirestore.getInstance();
```

```
postStoreRef =
storage.getReferenceFromUrl(dataSnapshot.child("storedpath").getValue().toString());

postStoreRef.delete();
```

. 7.21 irudia *Firestore Storage fitxategiak ezabatu*

7.3. Eredua

Azkenik MVC azken maila ikusiko dugu. Aplikazio honetan ereduak java klase bat besterik ez da izango eta lehen aipatu bezala datu basetik datu asko aldira lortzeko erabiliko da, hau da, kontrolatzaileak datuak hauek lista batean (*RecyclerView*) batean erakutsi behar dituenean.

Java klase honetan datu basetik irakurri nahi diren atributuen izenak edukiko dituen getterrak eta setterrak edukiko ditugu, horrela ondoren kontrolatzailetik atributu baten datua datu basetik lortu nahi badu nahikoa du ereduak getterra erabiltzea. Hurrengo kode zatian ikusiko dugu nolakoa den horrelako klase bat:

```
public class FindUsers {

    public String profileimage, fullname, username;

    public FindUsers(){}

    public FindUsers(String profileimage, String fullname, String username) {
        this.profileimage = profileimage;
        this.fullname = fullname;
        this.username = username;
    }

    public String getProfileimage() {
        return profileimage;
    }

    public void setProfileimage(String profileimage) {
        this.profileimage = profileimage;
    }

    public String getFullname() {
        return fullname;
    }

    public void setFullname(String fullname) {
        this.fullname = fullname;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

}
```

. 7.22 irudia *Ereduen inplementazioa Erabiltzaileen datuak lortzeko*

Klase honetako metodoei deiak egiteko memoriako atal honetan ikus dezakezu, zehatzago esanda kapitulu honetako Kontrolatzaile atalaren *Datu basetik datuak irakurri* azpi-atalean.

8

Ondorioak

Memoriako kapitulu honetan egindako proiektutik ondorio bat ateratzea izango da helburua. Ondorio hauek bi zatitan banatuko ditugu. Lehenik eta behin proiektuaren garapenetik ateratako ondorioak eta bestetik hobetu daitezkeen aspektuak eta gerora egin daitezkeen funtzioei buruz hitz egingo da.

8.1. Proiektuaren garapena

Memoriako *Kudeaketa* atalean azaldu bezala, proiektuaren helburu guztiak bete dira eta gainera hasieran ezarritako funtzioak baino gehiago sortu dira azkenean, adibidez erabiltzaileen ranking-a eta hauek maila eta esperientzia progresua gordetzea gehigarri moduan jarri genituen denbora baldin bazegoen.

Bestetik aplikazioak edukiko zuen datu basean, *Firebase*-en zehazki, arazo txiki batekin topo egin genuen proiektuaren hasieran. Ezabatu genuen erabiltzaile bat ez zen ondo ezabatu eta beraz erabiltzaile bat aplikaziora lehenengo aldiz sartzen zen bakoitzean hau apurту egiten zen eta aplikazioa itxi egiten zuen. Aukera bat zen gordeta geneukan datu basearen kopia erabiltzea baina hori egin beharrean *Firebase*-eko zerbitzu teknikoarekin harremanetan jarri ginen. Bertan arazo arraro baten aurren geundela esan eta lagundu zezaketen esteka batzuk pasa zizkigun.

Azkenean aurretik egindako segurtasun kopia erabili behar izan genuen eta ez zen arazo handia izan. Datu basearen segurtasun kopia gabe sekulako arazoak sortuko lituzke, are eta gehiago jadanik aplikazioa merkatuan bazegoen eta hau erabiltzaile desberdinen artean datuz betea egongo bazen. Horregatik garrantzitsua da bai garatzen goazen heinean zein gerora beti segurtasun kopia bat edukitzea.

Proiektua garatzeko erabilitako teknologiak zein lengoaiak gehienak aurretik ikusita geneuzkan, batik bat unibertsitatean bertan ikasita. Batez ere lengoaiak; XML, Java eta JSON ikusiak geneuzkan. Bestalde *Android Studio*-ren eta *Firebase* plataformaren erabilerak proiektu honetarako erabili ditugu lehenengo aldiz eta honekin batera, NoSQL edo datu base ez erlazionalen erabilera.

Lehendik MVC arkitektura sortutako programa bat eginda geneukan unibertsitatean bertan eta gainera Java lengoaiaren ezagutzak aplikatuta ez da oso zaila izan proiektuko nondik norakoak gauzatea. Denbora aldetik gehien eskatu duena Java lengoia *Android Studio*-n aplikatzea eta gerora hau nola dabilen ikastea izan da.

XML lengoaiaren erabilera izan da ezaugarri erakargarriena. Lengoaia hau datu base modura datuak gordetzeko erabili genuen orain arte, *Android Studio*-n ordea honen interfaze grafikoak garatzeko erabili da. Esan beharra dago bi aukera eskaintzen zituenez (interfazea XML-tik zuzenean edo eskuz antolatzea) oso eroso izan dela bertatik interfaze grafikoak garatzea.

Gainera oso garrantzitsua izan da aplikazioaren garapenean zehar probak egiten joatea eta hauek burutzeko hainbat pertsonen laguntza eduki dugu. Eskalada buruzko aspektuak probatzeko Pablo berak egin ditu proba desberdinak eta aldiz aspektu sozial eta akatsen bila ibiltzeko, aplikazioa nire lagunen artean banatu da.

Amaitzeko esan beharra dago oso interesgarria izan dela proiektu hau egitea eta etorkizunera begira, mundu errealeko enpresek eskatzen dituzten eskakizunak nolakoak diren eta erabiltzaile batzuen eskakizunak asebetetzeko sortu beharreko produktu bat garatzeak dakarren lana eta egin beharreko analisi eta azterketak oso aberatsak izan dira.

8.2. Hobetu daitezkeen aspektuak eta etorkizuna

Nahiz eta helburu bezala jarritako funtzionalitate guztiak bete ditugun, batez ere estatistikak eta erabiltzaileen maila kalkulatzeko sistemak oraindik oso gordinik daude. Alde batetik etorkizunean erabiltzaile bakoitzaren estatistikak aberatsagoak izango dira. proiektuaren garapenean hiru estatistika sortu ziren; *erabiltzaileak egindako metro kopurua*, *egindako bloke kopurua* eta *egindako erronken kopurua*. Gerora aspektu gehiago edukitzea espero dugu.

Beste aldetik garatutako produktuan erabiltzaileek maila eta esperientzia bat edukiko dute. Maila hobetzeko maila bakoitzak behar duen esperientzia kopurua lortu behar dira. Hau egiteko modu bakarra dago produktu honetan, erronkak betetzea. Erronkaren zailtasunaren arabera esperientzia puntu desberdinak lortuko dira. Etorkizunean hau aldatzea espero da, erronkak betetzeaz gain aplikazioan beste akzioak eginez ere puntuak lortzea, adibidez publikazio bat igotzean, blokeak sortzean edo balorazioak egitean.

Aipatu berri ditugun funtzioak sortuak daude eta hobetzea izango dira hauen helburua. Bestetik beste funtzionalitate batzuk ere pentsatuak ditugu: *mapa deskargatu ahal izatea*, *blokeak igotzeko krokisak igotzea hauen profiletan*, ...

Azkenik aplikazioa merkaturatu baino lehen Pabloren ezagunen artean banatuko da. Hauek bera bezala eskalatzailerak dira eta hilabete batzuk aplikazioa probatzen ibili ondoren zein funtzionalitate botatzen duten faltan eskatuko zaie. Hauen feedback-a jaso ondoren bideragarriak diren funtzionalitateak aztertuko eta garatuko dira aplikazioa aberatsagoa izateko.

9

Bibliografia

Azken kapitulu honetan proiektuan zehar erabili diren erreferentzia bibliografiko garrantzitsuenak (erabilitako tresnak eta API-ak) lista batean ordenatuak erakutsiko dira.

- [1] **Búlder** (adaptación del inglés boulder), **escalada en bloque** o **bouldering**, es una modalidad de escalada: <https://es.wikipedia.org/wiki/B%C3%BAlder>
- [2] **Android Studio** es el entorno de desarrollo integrado oficial para la plataforma Android <https://developer.android.com/studio>
- [3] **Android Studio API** <https://developer.android.com/reference>
- [4] **Firebase** es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por James Tamplin y Andrew Lee en 2012 y adquirida por Google en 2014. <https://firebase.google.com/?hl=es-419>
- [5] **Firebase API** <https://firebase.google.com/docs/reference/android/packages?hl=es>
- [6] **Firebase Authentication** <https://firebase.google.com/docs/auth?hl=es-419>
- [7] **Firebase Realtime Database** <https://firebase.google.com/docs/database>
- [8] **Firebase Storage** <https://developers.google.com/android/reference/com/google/firebase/storage/FirebaseStorage>
- [9] **NoSQL** <https://es.wikipedia.org/wiki/NoSQL>
<https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
- [10] **JSON** <https://firebase.google.com/docs/database/web/structure-data?hl=es-419>

A

Eranskina: Bileren aktak

Eranskin honetan proiektuaren garapenean zehar burutu diren bileren aktak erakutsiko dira. Bi bilera egongo dira, bat zuzendariarekin egindakoak eta besteak Pablarekin egindakoak. Akta hauetan hurrengo ezaugarriak ikusiko dira: *Data, Ordua, Bertaratuak, Lekua, Gai-zerrenda eta bileratik ateratako ondorioak*

1. BILERA AKTA

Data: 2018/11/28

Ordua: 16:00

Bertaratuak: Jon Iturrioz (zuzendaria) eta Ander Lopez (ikaslea)

Lekua: Informatika Fakultatea, Donostia

Gai-zerrenda:

- Egin beharreko GrAL-aren balorazioa.

Bileran hitz egindakoa eta ateratako ondorioak:

- Bileran egin aurretik zuzendariari Pablok bidalitako *“Plan de viabilidad”* edo *Bideragarritasun plana* bidali zaio honek proiektuaren ideia nagusia zein den jakin dezan.
- Bileran bertan bidalitako dokumentua aztertzen dugu bion artean, interesgarriak diren puntuak aztertuz. Honetaz gain proiektua izango den aplikazioaren helburu nagusiak eta ideia printzipala azaltzen dizkiot nire hitzetan zuzendariari.
- Zuzendariak ontzat ematen du erakutsitako proiektua eta zuzendari bezala esleituko da GrAL honetarako.

2. BILERA AKTA

Data: 2018/12/20

Ordua: 16:00

Bertaratuak: Pablo Cuenca (Proiektuko nagusia) eta Ander Lopez (ikaslea)

Lekua: Doctor Livingstone kafetegia, Donostia

Gai-zerrenda:

- Proiektuaren bideragarritasunari buruz hitz egitea.

Bileran hitz egindakoa eta ateratako ondorioak:

- Zuzendariarekin hitz egin ondoren aplikazioak edukiko dituen funtzionalitate garrantzitsuenei buruz hitz egin da. Hauen artean zeintzuk diren beharrezkoak eta zeintzuk diren bideragarriak aztertu dira. Azken finean Pablok erakusten zuen Bideragarritasun planak GrAL batek eskatzen duen lan kopurua baino handiagoa zen eta beraz hau mugatu beharra zegoen.
- Aplikazioak edukiko dituen funtzionalitateak adostu ondoren, Android Studio-n egitea proposatzen da Ionic plataforma ez zen ezagutzen eta honek denbora gehiago eskatuko zuelako, baina ez zen bilera honetan erabaki zein plataforma erabili aplikazioa sortzeko.

3. BILERA AKTA

Data: 2019/01/28

Ordua: 16:00

Bertaratuak: Jon Iturrioz (zuzendaria) eta Ander Lopez (ikaslea)

Lekua: Informatika Fakultatea, Donostia

Gai-zerrenda:

- Proiektuaren laburpena eta eskakizun bilketen adibideak

Bileran hitz egindakoa eta ateratako ondorioak:

- Proiektuaren azalpen sakonago bat bidali zaio zuzendariari bilera hasi aurretik, bertan aplikazioak edukiko dituen aktoreei buruz eta hasiera batean bakoitzak edukiko dituen funtzioei buruz hitz egingo da. Laburpena ontzat onartu da.
- Aplikazioaren erabilpen kasu batzuk jarri dira grafikoki eta domeinu eredu bat ere sortu da ez oso osatua. Zuzendariak aztertu egin ditu eta hau hobetzeko eta osatzeko ideiak eman ditu.
- Hurrengo bilerarako merkatuan dauden eta aplikazio honekin erlazionatuak dauden aplikazioak aztertzeko eskakizuna egin du zuzendariak.

4. BILERA AKTA

Data: 2019/03/2

Ordua: 15:00

Bertaratuak: Pablo Cuenca (Proiektuko nagusia) eta Ander Lopez (ikaslea)

Lekua: Skype bidez.

Gai-zerrenda:

- Proiektuaren jarraipena eta diseinuak

Bileran hitz egindakoa eta ateratako ondorioak:

- Bilera Skype bidez gauzatzen da Pablo Ekuador-en aurkitzen baitda. Bilera honetan aplikazioarekin noraino iritsi garen hitz egin dugu, zein funtzionalitate egin diren eta hauek nola hobetu. Oraingoz aplikazioaren alde soziala soilik inplementatu da, oraindik ez da eskaladari buruzko funtzionalitaterik egin.
- Eskaladako funtzionalitateekin hasteko Pablok aurretik eginda zituen interfazeen diseinu batzuk eman dizkit hauek nola interfazea eduki beharko duten ideia nagusi bat edukitzeko.

5. BILERA AKTA

Data: 2019/04/15

Ordua: 15:00

Bertaratuak: Jon Iturrioz (zuzendaria) eta Ander Lopez (ikaslea)

Lekua: Informatika Fakultatea, Donostia

Gai-zerrenda:

- Orain arte egindakoa aztertu eta eskaladako funtzionalitateei buruz hitz egin

Bileran hitz egindakoa eta ateratako ondorioak:

- Aplikazioa zein puntutan dagoen aztertu da eta proiektua amaitzeko geratzen den denboraren arabera zein funtzionalitateei lehentasuna eman erabaki da. Batez ere denbora aldetik justu ibiliko garelakoan hobe izango da funtzionalitate garrantzitsuenak eginda edukitzea.
- Memoriarekin hasiko gara, nahiz eta memoriak edukiko dituen atal batzuk hasita eduki (Eskakizun bilketak eta plangintza) hauek memoria izango den dokumentuan ezarriko dira eta jarraitu beharreko esturaturarekin hasiko gara memoria osatzen.
- Hurrengo bilerarako memorian egin diren atalak erakutsiko dira eta produktuak egin diren aurrerakuntzak erakutsiko dira.

6. BILERA AKTA

Data: 2019/05/16

Ordua: 15:30

Bertaratuak: Jon Iturrioz (zuzendaria) eta Ander Lopez (ikaslea)

Lekua: Informatika Fakultatea, Donostia

Gai-zerrenda:

- Memoriaren zuzenketa eta aplikazioaren garapenaren jarraipena

Bileran hitz egindakoa eta ateratako ondorioak:

- Memorian egindako aurrerapenak aztertu dira eta hobekuntza batzuk proposatu dira batez ere eskakizun bilketari dagokion informazioan. Zein beste atal gehitu memorian eta zeri buruz hitz egin ere erabaki da.
- Aplikazioari dagokionez aurrerapen handiak egin dira. Dagoeneko Administrazioaren funtzionalitateak eta Erabiltzaileari dagokion erronken atala soilik geratzen dira inplementatzeko. Geratzen den denboraren arabera hasiera batean ezarritako funtzionalitate guztiak bete ahal izango ditugu.
- Kasu honetan ere hurrengo bilerarako aplikazioa amaitua egotea espero da eta memoriako atal gehienak eginda egotea ere espero da.

7. BILERA AKTA

Data: 2019/05/27

Ordua: 15:00

Bertaratuak: Pablo Cuenca (Proiektuko nagusia) eta Ander Lopez (ikaslea)

Lekua: Skype bidez.

Gai-zerrenda:

- Aplikazioen erakusketa eta probak

Bileran hitz egindakoa eta ateratako ondorioak:

- Bileran hau baino lehen aplikazioa %95 ean amaitua dago (Administratzaile funtzionalitateak izango lirateke falta den %5a) eta Pablori eman zaio bera eta beren lagunen artean banatu eta probatzeko. Bileran adostua geratu da hasierako produktu bezala oso daudela implementatu diren funtzionalitatean eta erabiltzaileak gustuko izan dutela.
- Oraingoz falta diren Administratzailearen funtzionalitateak amaituko direla adostu da eta denboraz ondo bagoaz beste funtzionalitate gehigarri batzuk ere garatuko direla adostu da.

8. BILERA AKTA

Data: 2019/05/28

Ordua: 15:00

Bertaratuak: Jon Iturrioz (zuzendaria) eta Ander Lopez (ikaslea)

Lekua: Informatika Fakultatea, Donostia

Gai-zerrenda:

- Memorian jarraipena eta aplikazioaren amaiera

Bileran hitz egindakoa eta ateratako ondorioak:

- Memoria amaitzeko Inplementazioa, Diseinua eta Merkatuaren Analisia zuzenketak egitea falta da. Hau egin ondoren posta elektronikoz bidaliko zaio zuzendariari eta bertatik erantzungo dio memoria entregatzeko eta defenditzeko prest dagoen. Beraz ez dira bilera gehiago egingo.
- Aplikazioa amaitua dagoela erakutsi zaio eta egindakoarekin gustura geratu gara biok.

B

Eranskina: Bi proba garrantzitsuenak

Azken eranskin honetan proiektuaren garapenean zehar egin diren bi proba garrantzitsuenak erakutsiko dira eta arazoren bat gertatzen bazen, probekin nola konpondu diren ere azalduko dira.

1-. Proba: Bloke bat gehitzea eremu ezezagun batean

Bloke bat gordetzerakoan, gerora blokearen profilak eduki eta erakutsiko dituen atributuak eduki behar ditu. Google Maps-en API-a erabilia bloke bat gordetzerakoan honen lokalizazioa ere lortzen du. Aldiz eremu ezezagun batean baldin badago blokea Google-ren metodo honek *null* itzuliko digu eta beraz ez du blokearen *location* atributua sortuko. Honek blokearen profila atzitzerakoan errore bat sortuko du ez baitu atributu hori aurkitzen.

Hau konpontzeko implementatzeko garaian "Leku ezezagun" modura gordeko da *location* atributua. Nahiz eta ezezagun modura gorde, blokearen *latitude* eta *longitude* atributuak gordetzen direnez beti jakingo da blokea munduko zein zonatan dagoen.

```
if(!geocoder.getFromLocation(lat, lng, 1).isEmpty()) {
    List<Address> addresses = geocoder.getFromLocation(lat, lng, 1);
    Address obj = addresses.get(0);
    String add;
    add = obj.getCountryName() + ", " + obj.getSubAdminArea() + ", " + obj.getLocality();
    add = add.replace(", null", "");
    System.out.println(add);
    return add;
}else{
    location = "Sitio desconocido";
    return location;
}
```

2-. Proba: Bloke bat ezabatzerakoan erronkaren batean azken blokea bada erronka ere ezabatuko da.

Hasiera batean bloke bat ezabatzean erronka batean baldin bazegoen erregistratua, blokea erronkatik ere ezabatzen zen.

Orain arte guztiz ondo dagoela ematen du, baina erronka bat amaitzeko erabiltzaile batek egin dituen erronkako blokeak erronkak dituen bloke kopuruarekin konparatzen da eta biak berdinak badira erronka eginda bezala geratuko zaio erabiltzaile honi.

Hemen dator arazoa, blokeak ezabatzean erronka 0 blokerekin geratzen da eta beraz erabiltzaile batek 0 bloke eginda baditu erronkaren barruan, hau burutua bezala geratuko zaio. Hau ez gertatzeko behin blokea ezabatuta zenbat erronkari zenbat bloke geratzen zaizkion begiratzen da eta hau 0 ren berdina bada erronka ere ezabatu egingo da.

```
BoulderRef.removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()) {
            if (ds.child("Boulders").getChildrenCount() == 0) {
                ChallengeRef child(ds.getKey()).removeValue();
            }
        }
    }
});
```