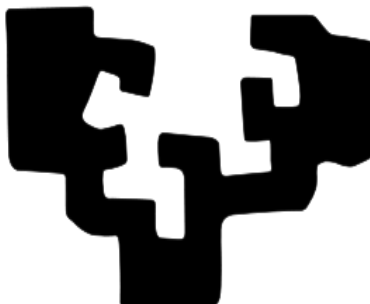


eman ta zabal zazu



Universidad Euskal Herriko
del País Vasco Unibertsitatea

Informatika fakultatea

TITULAZIOA: Informatikan ingeniaritza

ADIZKI BILAKETA ANALOGIAREN BIDEZ

Ikaslea:	Asier Kortajarena Guridi
Zuzendaria:	Kepa Mirena Sarasola Gabiola

Karrera amaierako proiektua, 2014ko Ekaina

Laburpena

Proiektu honen helburua adizkietan oinarritutako aplikazio bat egitea da Android sistema eragilea duten makinetarako. Bertan, erabiltzaileak nahi dituen adizkiak aurkitu ditzakete adizkien arteko analogia erabiliz eta gainera aldaketa motak kudeatuz, erabiltzaileak nahi dituen emaitza ezberdinak ikusi edota esaldi osoak bihurtu ditzakete besteak beste.

Aplikazio hau ahalik eta erabilgarriena izatea lortu nahi da eta nahiz eta hizkuntzalariak errazago erabili, edozein herritarrek erabiltzeko prestatua dago, esaterako euskara ikasi nahi duen ikasle batentzat tresna interesgarria izan daitekeelarik.

Proiektu hau, irakasle batek eskainitakoa izan da Asier Kortajarena Ingeniaritza informatikako ikasleak karrera amaierako proiektu bihurtu duelarik.

AURKIBIDEA

Laburpena	3
1. Sarrera	9
2. Proiektuaren definizioa	11
2.1 Definizioa	11
2.2 Motibazioa	13
2.3 Erabiltzaile motak	14
2.4 Hizkuntza-aplikazioen garrantzia	15
3. Teknologien azterketa	17
3.1 Aurrekariak	17
3.1.1 Antzeko aplikazioen azterketa	17
3.1.1.1 Aditzak jokatzeko (ADA)	17
3.1.1.2 Aditzak.com (Erref1)	18
3.1.1.3 Aditz laguntzailea (Erref2)	19
3.1.1.4 Aditzak (Mintzalagun.com) (Erref3)	19
3.1.1.5 Adizkitekia (Erref4)	20
3.1.2 Aurrekariaren konparaketa	21
3.1.3 Abantailak	22
3.2 Android	22
3.2.1 Androiden funtzionamendua	23
3.2.2 Garapenerako tresnak	26
3.2.2.1 Eclipse	26
3.2.2.2 Android Studio	27
3.2.3 Aukeraturako tresnak	28
4. Proiektuaren kudeaketa	29
4.1 Proiektuaren garapena eta esfortzuen hausnarketa	29
4.2 Esfortzuen hausnarketa	29
4.3 LDE eta azpi-atzen zerrenda	32
4.3.1 Lanaren Deskonposaketa Egitura diagrama (LDE)	32
4.3.2 Proiektuko ataza eta azpi-atzak	33
4.4 Plangintzako Gantt VS Gantt erreala	34
4.4.1 Plangintzako Gantt diagrama	34
4.4.2 Gantt diagrama erreala	34
4.5 Eginkizunen hausnarketa	35
4.5.1 Egindako eginkizunak	35

4.5.2 Egin gabeko eginkizunak	36
5. Garapen teknikoa.....	37
5.1 Eskakizunen bilketa	37
5.1.1 Erabilpen Kasuen Eredua (EKE)	37
5.1.1.1 Deskribapen zehatzak	37
5.1.1.2 Gertaera-fluxuak	40
5.2.1 Sekuentzia diagramak	43
5.3 Diseinua	51
5.3.1 Sistema sekuentzia-diagramak.....	51
5.3.2 Klase diagrama	59
5.4 Sistemaren arkitekura.....	59
5.5 Implementazioa.....	61
5.5.1 Implementazioan arazoak	61
5.6 Probak	66
5.6.1 Garatzailearen probak.....	66
5.6.2 Proba errealak	68
5.7 Ezarpen plana	70
6. Ondorioak.....	71
6.1 Proiektuaren balorazioa.....	71
6.2 Balorazio pertsonala	71
6.3 Etorkizunerako hobekuntzak.....	72
7. Bibliografia.....	73
7.1 Erreferentziak.....	73
7.2 Laguntza-gidak	73
8. Eranskinak	75
8.1 Eskuliburua.....	75
8.1.1 Modu arrunta	75
8.1.2 Aldaketa motak	78
8.1.3 Esaldiak	82
8.2 Kategorien itzulpena.....	84
8.3 Proiektuaren helburu dokumentua	85
8.3.1 Sarrera	85
8.3.2 Helburuak.....	86
8.3.3 Lanaren Deskonposaketa Egitura diagrama(LDE).....	87
8.3.4 Proiektuko ataza eta azpi-atazak	88
8.3.5 Planifikazioa.....	89

8.3.6 Eginkizunak	91
8.3.7 Garapen prozesua	92
8.3.8 Lan metodologia	93
8.3.9 Arriskuak	93

IRUDIEN AURKIBIDEA

1.irudia: Adizkien datu-basea	12
2.irudia: Oinarrizko programazioko praktika ADAn garatua	17
3.irudia: Aditzak.com aplikazioaren Android bertsioa	18
4.irudia: Aditz Laguntzailea aplikazioa	19
5.irudia: Aditz Laguntzailea aplikazioa	20
6.irudia: Adizkitegia aplikazioa	21
7.irudia: Android Smartphonen salerosketen hazkundera	22
8.irudia: Aplikazioaren manifestuaren adibidea	23
9.irudia: Aplikazioko diseinu baten adibidea	25
10.irudia: Aplikazioa Eclipse ingurunean garatzen	26
11.irudia: Aplikazioa Android Studio ingurunean diseinatzen	27
12.irudia: Esfortzuaren estimazioa eta egindako errealaren konparaketa	30
13.irudia: Esfortzuaren estimazio eta egindako errealaren konparaketa atazen arabera	30
14.irudia: KonparaketaActivity-ko parametroen erabilera	64
15.irudia: Androideko sailkapena pantaila tamainaren arabera	65
16.irudia: Dropboxeko apk fitxategia	70
17.irudia: Menu nagusia	75
18.irudia: Modu arrunta	76
19.irudia: Zutabe ezabaketa prozesua	77
20.irudia: Desanbiguatutako adibidea	77
21.irudia: Hizkuntza aldaketaren adibidea	78
22.irudia: E-maila anbiguoaren adibidea	78
23.irudia: E-maila arteko konparaketa	79
24.irudia: Aldaketa mota sortzen	79
25.irudia: Adizkia ez da existitzen	80
26.irudia: Aldaketa mota existitzen da	80
27.irudia: Aldaketa mota ondo sortua	81
28.irudia: Aldaketa mota aldatzeko dialogoa	81
29.irudia: Aldaketa mota ezabatua	82
30.irudia: Esaldia mikrofono bidez ahoskatzen	82
31.irudia: Esaldiaren e-mailaren adibidea	83

DIAGRAMEN AURKIBIDEA

1.diagrama: Androiden aktibitate- diagrama.....	24
2.diagrama: Androiden fragmentu-bilaketa.....	24
3.diagrama: Lanaren deskonposaketa egitura	32
4.diagrama: Plangintzako Gantt.....	34
5.diagrama: Gantt erreala	34
6.diagrama: Erabilpen Kasuen Eredua.....	37
7.diagrama: Adizkiak bilatu sekuentzia-diagrama	43
8.diagrama: Adizkia aukeratu sekuentzia-diagrama.....	44
9.diagrama: Adizkia ezabatu sekuentzia-diagrama	45
10.diagrama: Emaitzak konparatu sekuentzia-diagrama	46
11.diagrama: Aldaketa mota sortu sekuentzia-diagrama	47
12.diagrama: Aldaketa mota aldatu sekuentzia-diagrama.....	48
13.diagrama: Aldaketa mota ezabatu sekuentzia-diagrama	49
14.diagrama: Esaldia bihurtu sekuentzia-diagrama.....	50
15.diagrama: hitzaBihurtu() sistema sekuentzia-diagrama	51
16.diagrama: konparaketaBidali() sistema sekuentzia-diagrama.....	52
17.diagrama: aukeratu() sistema sekuentzia-diagrama.....	52
18.diagrama: konparaketaBidali() sistema sekuentzia-diagrama.....	53
19.diagrama: ezabatu() sistema sekuentzia-diagrama	53
20.diagrama: konparaketaBidali() sistema sekuentzia-diagrama.....	54
21.diagrama: sortuDialogoa() sistema sekuentzia-diagrama	55
22.diagrama: sortu() sistema sekuentzia-diagrama.....	55
23.diagrama: aldatuDialogoa()sistema sekuentzia-diagrama	56
24.diagrama: aldatu() sistema sekuentzia-diagrama.....	56
25.diagrama: aldatuDialogoa() sistema sekuentzia-diagrama	57
26.diagrama: ezabatu()sistema sekuentzia-diagrama	57
27.diagrama: bihurtu() sistema sekuentzia-diagrama	58
28.diagrama: Klase diagrama.....	59
29.diagrama: Sistemaren arkitektura orokorra	59
30.diagrama: Aktibitateen fluxu-diagrama	60
31.diagrama: Plangintzako Lanaren deskonposaketa egitura	87
32.diagrama: Plangintzako Gantt Diagrama.....	90
33 diagrama: Fase eskema	92

1. Sarrera

Proiektu honetan egiten den aplikazioa, Informatikako Ingeniaritzako karrera amaierako proiektua da. Proiektua egiteko proposamena Otsailaren amaieran jaso zuen ikasleak eta euskararen inguruko proiektua nahi zuenez, berehala onartu zuen proposamena. Proiektua Martxoaren hasieran hasi eta Ekaina amaiera arte iraun du , lau hilabeteko epe batean.

Lana proposatu zuen irakasleak adizkien - edo aditz-formen - inguruko praktika bat eginda zuen oinarritzko programazioko ikasleekin eta horren ardatzera, aplikazio hori *Androidera* ematea baliagarria dela pentsatu zuen. Honela , praktika hau aurkeztu eta adizkien bilaketarako sistema bat ikusi da , modu sinple batean 3 adizki emanda eta analogia bidez adizkiak aurkitzen dituen emaitza gisa.

Praktika hau ordea , *Ada* programazio lengoaiaz egindako praktika bat da , interfaze gabekoa eta erabilgarriagoa izateko hobekuntza ugari egin daitezkeela ikusi da. Lehenik eta behin, praktika *Androidera* eramatea gaur egun hazkunde handia baitu sistema eragile honek. Honetaz gain, 3 adizki ematen ibili beharrean aldaketa motak sortuta aplikazioa askoz ere erabilgarriagoa izan daitekeela ikusi da eta gainera aldaketa motak leheneste aukera ematen bada, adizki bat jarrita (edo ahotsez esanda) soilik lor daiteke emaitza, hau da botoi bat sakatuz soilik emaitza esan eta entzun daiteke, erabilera sinple eta praktikoa bat lortuz . Azkenik, hobekuntza hauez gain, esaldi osoak bihurtzeko edota hizkuntzak dinamikoki aldatzeko aukera ere ematen da.

Aplikazio hau erabiltzaile guztiei zuzenduta dago baina hizkuntzalariei eta batez ere euskara ikasten ari den herritarrei benetan baliagarri gerta dakieke.

Laburbilduz, helburu nagusia Adan garatutako aplikazio bat oinarritzat hartuta, ahalik eta erabilgarritasun eta irisgarritasun gehien bermatuz , edonork adizkien bilaketak egiteko *Android* sistema eragileetarako aplikazio bat garatzea da.

2. Proiektuaren definizioa

2.1 Definizioa

Proiektu honen ardatza adizkiak izango dira. Adizkien artean konparatzeko aukera emango du garatuko den proiektu honek. Berau, Android ingurunerako sortuko da mugikorretan modu erabilgarri eta xume batean erabiltzaileek adizkiei buruz ikasi edo zalantzak argitu ditzaten.

Adizki bakoitzak bere ezaugarriak dituzte eta kasu honetan adizki bakoitzari 7 ezaugarri aplikatzen zaizkio, ondorengoak hain zuzen:

- Lema: Zein aditz trinkotik datorren
- Hitano: Hikaz adierazita dagoen ala ez , Toka eta Noka bereiziz (Aukerazkoa)
- Nor: Aditzaren nor pertsona (Aukerazkoa)
- Nork: Aditzaren Nork pertsona (Aukerazkoa)
- Nori: Aditzaren nori pertsona (Aukerazkoa)
- Modu-denbora: Modu eta denboraren ezaugarriak (Adib. Subjuntibozko lehenaldia)
- Aspektua: Zein aspektu duen.

Hau horrela sistemak aditz-forma horien ezaugarri gramatikalak erabiliz gero beraien artean konparaketa bat eginez beste adizki batzuk bilatzeko aukera ematen digu gramatika konplexuetara joan beharrik izan gabe. Hau egiteko, lehenik eta behin ahalik eta adizki gehien daukan datu-base bat behar dugu eta bertan adizki bakoitzeko informazioa gordetzen da. Proiektu honetan erabili den datu-basea testu-fitxategi batetik dator eta testu horrek lerro bakoitzeko adizkiaren izena eta aipatu berri diren ezaugarriak erakusten ditu. Guztira 17.000 euskal adizki baino gehiagoren ezaugarriak gordetzen dira bertan. Modu-denbora ezaugarria (5.zutabea), kategoria baten bidez dago adierazita eta horren itzulpenak eranskinean ikus daitezke (

8.2 Kategorien itzulpena).

genikean	ukan	TO	HURA	GUK		B3	PNT
genikek ukan	TO	HURA	GUK		B2	PNT	
geniken ukan	NO	HURA	GUK		B2	PNT	
genikenan	ukan	NO	HURA	GUK		B3	PNT
genin ukan		HURA	GUK	HIRI-NO	B4	PNT	
geninake	ukan		HURA	GUK	HIRI-NO	B2	PNT
geninakeen	ukan		HURA	GUK	HIRI-NO	B3	PNT
geninan ukan		HURA	GUK	HIRI-NO	B1	PNT	
geninan ukan	NO	HURA	GUK		B1	PNT	
genio ukan		HURA	GUK	HARI	B4	PNT	
genio esan		HURA	GUK		B4	PNT	
genio esan		HURA	GUK		B6	PNT	
genioan esan	TO	HURA	GUK		B1	PNT	
genioan ukan	TO	HURA	GUK	HARI	B1	PNT	
genioen esan		HURA	GUK		B1	PNT	
genioen esan		HURA	GUK		B5B	PNT	
genioezen	esan		HAIEK	GUK		B1	PNT

1.irudia: Adizkien datu-basea

Datu-base honetan ordea arazo larri bat aurkitzen da analogia aplikatu nahi badugu, izan ere aurreko irudian ikus daitekeen bezala, adizki asko anbiguoak dira , hots, izen berdina dute baina ezaugarri gramatikal ezberdinak eta horrek konparaketak egiterakoan emaitza posible asko eman dakizkiguke, agian behar ez direnak barne. Horregatik, garrantzitsua da analogia egin aurretik adizkiak desanbiguatzea nahi diren emaitzak lortzeko. Noski, desanbiguatze nahi bada ezaugarri bakoitza zer den ulertu behar da.

Hori dela eta, aplikazioari beste aukera bat gehitu zaio. Hitanoa itzultzeko aukera aldeztu aurretik ematea , izan ere adizki bat hitanora bihurtzean ez dago anbiguotasunik analogia egiterako orduan eta gainera, erabiltzaile batek hizkuntzalaritzari buruz ezer jakin gabe hitz edo esaldi bat zukatik hitanora pasa dezake, bai nokara (nesken hitanoa) edota tokara (mutilen hitanoa). Horretarako adizki lehenetsi bat hartzen da (‘ditut’) eta bigarrenagoak lehenengoaren ezaugarri gramatikal berdina izan behar ditu baina hitanoa gehituta (‘ditiat’ tokaz edo ‘ditinat’ nokaz). Azkenik bihurtu nahi den adizkiari lehenengo eta bigarrenaren arteko ezberdintasuna aplikatzen zaio.

Adibidez, demagun tokara pasa nahi dela adizkia eta horretarako prozesua honakoa da:

1. Lortu lehen bi aditz-formak eta beraien ezaugarri gramatikalak:
Kasu honetan, lehenengoa ‘ditut’ izango da eta bigarrena ‘ditiat’.

Adizkia	Lema	Hitanoa	Nor	Nori	Nork	Modu denbora
<i>Ditut</i>	ukan	-	Haiek	-	nik	Indikatibozko orainaldia
<i>Ditiat</i>	ukan	Toka	Haiek	-	nik	Indikatibozko orainaldia

Ezberdintasunak aplikatu ondoren argi dago hitanoaren ezaugarria aldatzen dela soilik, ‘ditiat’ adizkiak tokaz baitago eta ‘ditut’ aldiz ez

2. Hirugarren adizkiaren izena lortu: Adibidez, ‘dauka’.

Adizkia	Lema	Hitanoa	Nor	Nori	Nork	Modu denbora
<i>Dauka</i>	ukan	-	Hura	-	Hark	Indikatibozko orainaldia

3. Hirugarren adizkiari lehenengo bien arteko ezberdintasuna aplikatu: ‘dauka’ hitzari ‘toka’ ezaugarria aplikatzen zaio.

Adizkia	Lema	Hitanoa	Nor	Nori	Nork	Modu denbora
<i>Dauka</i>	ukan	Toka	Hura	-	Hark	Indikatibozko orainaldia

4. Emaizta lortzeko ,aldaketa aplikatu zaionean hirugarren adizkiak eskuratu dituen ezaugarriak hartzen dira, emaitza lortzeko.

Adizkia	Lema	Hitanoa	Nor	Nori	Nork	Modu denbora
???	ukan	Toka	Hura	-	Hark	Indikatibozko orainaldia

5. Azkenik, lortu berri den ezaugarri berri hori zein adizkik duen edo duten bilatu behar da datu-basean.

Adizkia	Lema	Hitanoa	Nor	Nori	Nork	Modu denbora
<i>zaukaat</i>	ukan	Toka	Hura	-	Hark	Indikatibozko orainaldia
<i>zeukaat</i>	ukan	Toka	Hura	-	Hark	Indikatibozko orainaldia

Kasu honetan, ezaugarri berdina duten bi adizki agertu dira, biak onartuta baitaude. ‘zaukaat’ eta ‘zeukaat’ hain zuzen ere.

Hau guztia honela izanik, aldaketa mota ezberdinak egin ditzakegu eta horregatik aldaketa mota propioekin analogiak egin nahi dituztenek ezaugarri gramatikal hauek menperatzea beharrezkoa dute. Aldiz, hitanoa erabiltzeko alde aurretik aukera ematen da edonork erabil dezan. Gainera, segurtasun arrazoiengatik ezin da hitanoen inguruko informazioa aldatu edo kendu.

2.2 Motibazioa

Proiektu hau egiteko motibazio nagusia, euskararen inguruko proiektu bat egitea izan da eta gainera proiektuan aurrera joan ahala erabilgarritasuna handiagoa izan daitekeela ikusi da aplikazio xume baina praktiko bat sortuz , euskarari ekarpen interesgarri bat egitearekin batera.

Aplikazio honek erabilera ugaria izan dezake. Badira aurrekarietan aipatuko diren beste aplikazio batzuk (3.1 Aurrekariak) aditzekin jolasteko aukera ematen dutenak baina horietan adizki bat aurkitzeko behar beharrezkoa da adizki horren ezaugarriak jakitea eta proiektu honek aldiz, lan hori saihesteko aukera ematen du . Aurreko aplikazioek 1.000 deskargatik gora dituztenez Androiden, nahikoa arrazoi izan da lana aurrera eramateko aplikazio osatuago bat eraikiz.

Aplikazio hau oso erabilgarria izan daiteke erabiltzaile batzuentzat. Esaterako, euskaltegian ikasten dabilen norbaitek , hitanoz ez badu ongi idatzi edo hitz egiten, aplikazio honek bere hitanoa hobetzen erakuts diezaioke. Modu berean aldaketa mota sinpleak sortuz, lehenaldira edo orainaldira pasatzerako garaian ere.

Modu bertsuan, herritarren batek agian badaki tokaz baina gero neskekin hitz egiterakoan nokaz ez ditu aditzak asmatzen eta honek edozein momentutan ahalbidetuko lioke mikrofonoaren bidez nahi duena esan eta nokara pasatzea neskak hikaz hitz egin nahi balu. Hori guztia egiteko bi botoi klikatu beharko lituzke soilik, bata mikrofonoa martxan jartzeko eta bestea bihurtzeko eskatzeko.

Guzti hau dela eta, aplikazioa sortzeko nahikoa arrazoi ikusi da eta bideragarritasuna duela dirudi. Antzeko beste aplikazioek 1.000 deskarga baino gehiago jaso dituzte Google Play-tik eta kopuru hori ez da batere txikia euskal teknologiaren inguruan dagoen baliabide urriak ikusita.

2.3 Erabiltzaile motak

Sarritan aipatu bezala, aplikazio hau edozein erabiltzaile motak erabil dezakete baina horren barruan sailkapen bat egin daiteke erabilera mota batzuk gomendatuz:

- Erabiltzaile arruntak: Euskararen gramatikari buruz ezer ez dakiten erabiltzaileak. Hauek aplikazioa hitanoa ikasteko erabil dezakete edota aurretik aipatu bezala gogoratzen ez diren adizkiren bat noka edo tokara pasatzeko edozein unetan.
- Euskara ikasten ari direnak: Hauek hitanoa erabiltzeaz gain aldaketa motekin “jolas” dezakete mota ezberdinekin zein emaitza aterako diren probatzeko eta horrela erabiltzailearen euskal gramatika hobetzen joateko. Nahaspila asko sortuz gero datuak ezabatzea nahikoa izango luke eta berriro hitanoa soilik agertuko litzaioke.

- Euskal gramatika menperatzen dutenak: Hizkuntzalariak edo hizkuntzalaritzarekin interesatuak izan ohi dira. Hauek aldaketa motak modu egokian kudeatzen jakin eta aplikazioaren tresna guztiak aprobetxatzeko gaitasuna izan beharko lukete, tresna serioago bat bilakatuz. Adibidez, euskara irakasle batek nahi duen aldaketa mota bat sortu eta bere ikasleei horiek ikasteko tresna bat eman dakioke.

2.4 Hizkuntza-aplikazioen garrantzia

Gaur egun hizkuntzen inguruan teknologia berri ugari sortu dira eta modu berean hizkuntzak erabiltzeko gero eta aplikazio gehiago daude nonahiko konputazioan eta batez ere hizkuntzetatik abiatuta itzultzaile ugari dira sarean eskuragarri.

Hizkuntza-aplikazioek berebiziko garrantzia dute gaur egun, “informazioaren garaian” bizi garenean, eta esaterako oso arraroa da bidaiatzera joan eta atzerriko hizkuntzen hiztegi bat ematea mugikorrean aplikazioen bat eman beharrean hizkuntzarekin egon daitezkeen arazoak saihesteko. Gainera gero eta lokalizazio gehiago egiten dira edozein motako aplikazioetan hizkuntza naturala eta kultura uztartuta aberastasun handiagoa lortzen delako eta hizkuntzalaritza aplikatuaren zati garrantzitsua bihurtu da hizkuntza-aplikazioa.

Honetaz gain, kasu honetan euskararen inguruko aplikazioa egin denez, euskarazko aplikazioen garrantzia ere azpimarratu beharra dago. Izan ere, gaur egun euskarazko teknologien edota aplikazioen eskaintza oso urria da eta horregatik, euskararen inguruko edozerk euskarari eta euskararen teknologiari bultzada ematen dio eta euskarak bultzada honen beharra dauka .

3. Teknologien azterketa

3.1 Aurrekariak

3.1.1 Antzeko aplikazioen azterketa

Hona hemen sarean eskuragarri dauden antzeko aplikazioen azterketa: 5 aplikazio aztertu dira guztira egingo den proiektuarekin antz gehien daukatenak.

3.1.1.1 Aditzak jokatzeko (ADA)

Aplikazio hau ADA programazio-lengoaia erabiliz garatuta dago eta proiektu honen oinarria izan da. 'Oinarrizko Programazioa' ikasgaian garatu dute ikasleek eta datu-base berdina erabiliz, proiektu honetan modu arruntean lortzen dena egiten du.

Erabiltzaileak 3 adizki pasatzen dizkio eta analogia egin ondoren emaitzak pantailaratzen dira komando-leiho batean.

```
Idatzi 1. aditz forma: ditut
Idatzi 2. aditz forma: ditiat
Idatzi 3. aditz forma: daukat

Emaitzak ondorengoak dira:
1. ditut 1: ukan 2: 3: HAIEK 4: NIK 5: 6: A1
   ditiat 1: ukan 2: TO 3: HAIEK 4: NIK 5: 6: A1
   daukat 1: eduki 2: 3: HURA 4: NIK 5: 6: A1
   (*) zaukaat 1: eduki 2: TO 3: HURA 4: NIK 5: 6: A1
   (*) zeukaat 1: eduki 2: TO 3: HURA 4: NIK 5: 6: A1

Idatzi 1. aditz forma:
```

2.irudia: Oinarrizko programazioko praktika ADAn garatua

Interfazerik gabeko proiektua da exekutagarria martxan jartzean komando-leihoaz azaleratzen baita. Gainera, aplikazio hau nola egin den ikusteko aukera izan da eta

zerrendak soilik erabiltzen direnez, iterazio eragiketa bakoitzean lor litekeen konplexutasun orden onena, $O(n)$ orden lineala da eta horrek denboraren kostua ekartzen du.

Abantailak: Analogia bidez egiten dira bilaketa

Desabantailak: Interfazerik gabekoa, ez da eleanitza eta ez dago esaldi osoak tratatzeko aukerarik.

3.1.1.2 Aditzak.com (Erref1)

Aplikazio hau ere adizkietan oinarritzen da eta bere egitekoa sinplea da. Erabiltzaileak modua, denbora, nork, nori, nor eta mota (nor-nori-nork) aukeratu ondoren baldintza horiek betetzen dituzten emaitzak ematen ditu. Hala ere, ez ditu ez aspektua eta ez hitanoa kontuan hartzen.



3.irudia: Aditzak.com aplikazioaren Android bertsioa

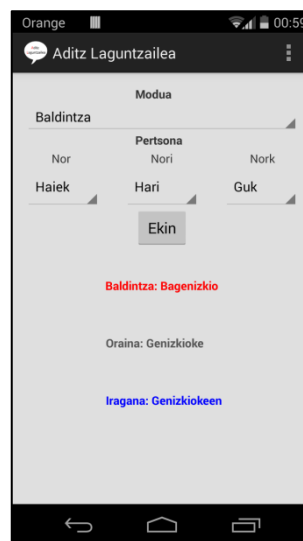
Abantaila bezala, aplikazioa plataforma anitza da eta web-ean ez ezik, Android eta IOS sistema eragiletarako garatuta dago. Dagoeneko, Play Store dendan 1.000 deskargatik gora dauzka.

Abantailak: Plataforma anitza da eta adizki bilaketa ezaugarrien arabera egiten du. Interfaze erabilgarria diseinu aurreratuarekin.

Desabantailak: Hitanoa eta aspektua ez dira kontuan hartzen, bilaketa bat egiteko 3 pantailatik eta 4 botoi sakatu behar dira eta ez da eleanitza. Ezin da analogia bidez bilatu eta esaldiak tratatu.

3.1.1.3 Aditz laguntzailea (Erref2)

Aplikazio hau ere Androiderako egina dago eta aurrekoaren ia berdina da. Modua, nor, nori eta nork formak hartu eta botoi bati sakatuta emaitzak agertzen orainaldi, lehenaldi eta alegiazko denboretan. Honek ere dagoeneko 1.000 deskargatik gora dauzka Play Store-an.



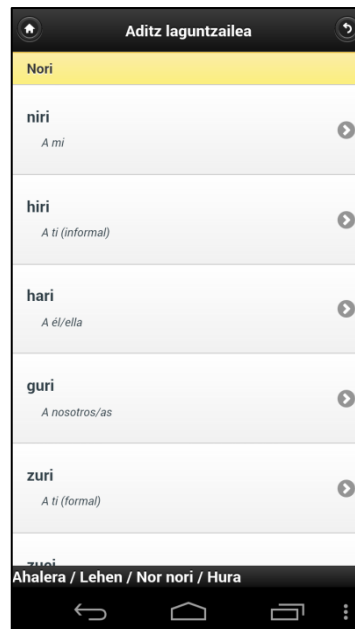
4.irudia: Aditz Laguntzailea aplikazioa

Abantailak: Adizki bilaketa ezaugarrien arabera.

Desabantailak: Hitanoa eta aspektua ez dira kontuan hartzen eta interfaze zeharo sinplea. Ez du adizkien arteko analogiarik egiten eta ez da eleanitza. Ezin da analogia bidez bilatu eta esaldiak tratatu.

3.1.1.4 Aditzak (Mintzalagun.com) (Erref3)

Aurreko bien antzekoa da hau ere, izan ere Modua, denbora, nor, nori eta nork formak emanda adizkia bilatzen du. Guztia bi hizkuntzetan ikus daiteke; gaztelania edo euskara. Dagoeneko 1000 deskargatik gora ditu Play Store-an.



5.irudia: Aditz Laguntzailea aplikazioa

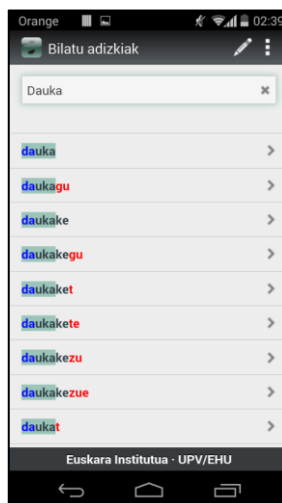
Abantailak: Adizki bilaketa ezaugarrien arabera, eleanitza da eta adibideak ematen ditu.

Desabantailak: Hitanoa eta aspektua ez dira kontuan hartzen eta gutxienez 5 pantailatan zehar ibili behar da emaitza lortzeko. Ezin da analogia bidez bilatu eta esaldiak tratatu.

3.1.1.5 Adizkitegia (Erref4)

Euskara institutuak EHUrekin elkarlanean sortua, ezaugarrien arabeko bilaketaz gain, aditz bilaketa trinko bat ere eskaintzen du. Ezaugarrien arabeko bilaketa egiteko lehenengo aditz laguntzailea aukeratu eta gero nor, nori, nork, modua eta denbora aukeratu behar dira nahi den adizkia lortzeko. Interfaze eroso dauka eta koloreen bidez adizkien ezberdintasunak ikusteaz gain eta nahikoa azkar egiten dira bilaketak.

Aplikazio honek 500 deskarga baino gehiago ditu Play Storean.



6.irudia: Adizkitegia aplikazioa

Abantailak: Adizki bilaketa ezaugarrien arabera eta adizki bilaketa trinkoa. Interfaze erabilgarria eta eroso dauka.

Desabantailak: Hitanoa eta aspektua ez dira kontuan hartzen eta gutxienez 5 pantailatan zehar ibili behar da emaitza lortzeko. Ezin da analogia bidez bilatu eta esaldiak tratatu. Gainera ez da eleanitza eta pantaila batean ez dago atzera beste pantailara joaterik hasierara bakarrik itzultzen uzten duelarik.

3.1.2 Aurrekarien konparaketa

Konparaketa-aula honetan adierazten da aurrekarien arteko ezberdintasun nagusienak proiektu honetako aplikazioarekin alderatuz

Aplikazioa	Analogia bidezko bilaketa	Ezaugarri bidezko bilaketa	Esaldien tratamendua	Hizkuntza anitza	Plataforma
Aditzak jokatzeko (ADA)	✓	✗	✗	✗	ADA (PC)
Aditzak.com	✗	✓	✗	✗	Android, Web eta IOS
Aditz laguntzailea	✗	✓	✗	✗	Android
Mintzalagun.com	✗	✓	✗	✓	Android
Adizkitegia	✗	✓	✗	✗	Android
Aditzak jokatzeko (Android)	✓	✗	✓	✓	Android

3.1.3 Abantailak

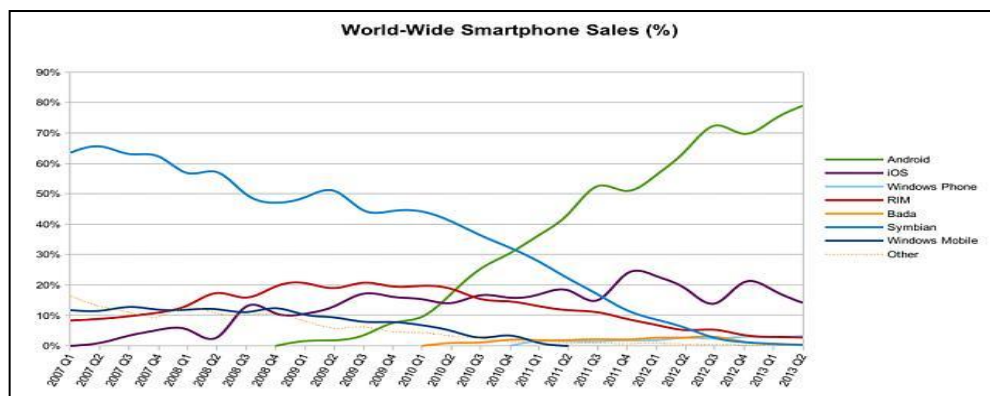
Tresna aurrekariak aurkitu ondoren argi geratu den lehen gauza , ezaugarri bidezko bilaketa egiten duten tresna ugari daudela izan da eta sortu nahi den aplikazioak erabilera berriak izan behar dituen, aukera berriak uztartzea lehenesten da. Hori dela eta, momentuz ezaugarri bidezko bilaketak ezin dira egin aplikazio honetan.

Hori alde batera utziz, aurrekari hauei abantailak ateratzeko aukerak analogiaren bidez bilaketak egin eta horrela aldaketa motak kudeatzeko aukera izateaz gain, esaldi osoak tratatzeko modua eta interfaze eleanitza izatea erabaki da tresna berriak sortuz besteen aldean erabilera ezberdin eta osoago bat izango duena ahalik eta erabiltzaile gehien eskuratzeko

3.2 Android

Aplikazio hau garatzerako orduan lehenengo ideia nonahiko konputaziorako zerbait egitea izan da , izan ere gaur egun edozein konputagailu nonahi erabil daitezke izan dezaketen tamaina txikiagatik eta ez dago ezer erabilgarriagorik zerbait mugikorra sortzea baino. Hori dela eta, mugikorretarako aplikazio bat egitea pentsatu eta zuzenean Android sistema eragilerako egitea pentsatu da.

Sistema eragile honen hazkundea esponentziala izan da eta European mugikorraren %70eko monopolioa eskuratu du. Hau ikusita, normaltzat jo daiteke milaka garatzailek Android sistema eragilerako aplikazioak sortu nahi izatea, eta horrela, *Google Play Store* dendan dauden aplikazio kopuruaren hazkundea ere izugarria izan da, egun 1.000.000 aplikaziotik gora daudelarik eta hazkundea etengabea da.



7.irudia: Android Smartphonen salerosketen hazkundea

Hau guztia ikusirik, egin beharreko aplikazioa Android sistema eragilerara bideratuta egon behar dela ikusi da.

3.2.1 Androiden funtzionamendua

Androiden aplikazio bat garatzerakoan elementu asko izan behar dira kontuan baina hemen aplikazioan behar izan diren elementu aipagarrienak azalduko dira:

- **AndroidManifest.xml:** Aplikazioaren burmuina da. Bertan adierazten da zenbat aktibitate dauden, bakoitzaren ezaugarri eta murriztapenak (esaterako zein den aktibitate abiarazlea), zein aktibitateren gurasoa den, zein orientazio onartzen den, aktibitate bakoitzeko tituluak, SDK bertsio minimo eta konpilaziokoa etab.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.aditzekin_jokatzen"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-feature
        android:name="android.hardware.microphone"
        android:required="false" />

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity android:name="com.example.aditzekin_jokatzen.KargatuActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait"
        >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        </activity>
        <activity
            android:name="com.example.aditzekin_jokatzen.MenuActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait"
        >
    </application>
</manifest>
```

8.irudia: Aplikazioaren manifestuaren adibidea

Aplikazioak fitxategi honen arabera gidatzen du alde batera edo bestera , beraz oso garrantzitsua da hemen errorerik ez egotea.

-
- ```
graph TD
 A([Activity launched]) --> B[onCreate()]
 B --> C[onStart()]
 C --> D[onResume()]
 D --> E([Activity running])
 E --> F[onPause()]
 F --> G[onStop()]
 G --> H[onDestroy()]
 H --> I([Activity shut down])
 F -- "User returns to the activity" --> D
 G -- "User navigates to the activity" --> C
 G -- "User navigates to the activity" --> J[onRestart()]
 J --> C
 E -- "App process killed" --> K([App process killed])
 K -- "User navigates to the activity" --> B
 F -- "Apps with higher priority need memory" --> L([App process killed])
 L -- "User navigates to the activity" --> B
 F -- "The activity is no longer visible" --> G
 F -- "Another activity comes into the foreground" --> G
 G -- "The activity is finishing or being destroyed by the system" --> H
```
- The flowchart illustrates the lifecycle of an Android Activity. It begins with 'Activity launched', leading to 'onCreate()', 'onStart()', and 'onResume()' methods. The activity then enters the 'Activity running' state. From here, it can be paused ('onPause()') or stopped ('onStop()'). 'onPause()' can lead back to 'onResume()' if the user returns to the activity. 'onStop()' can lead back to 'onStart()' if the user navigates to the activity, or to 'onRestart()' which then leads to 'onStart()'. 'onStop()' can also lead to 'App process killed' if the app process is killed or if apps with higher priority need memory. 'onStop()' can also lead to 'onDestroy()' if the activity is no longer visible, another activity comes into the foreground, or the activity is finishing or being destroyed by the system. 'onDestroy()' leads to 'Activity shut down'.

) metodora doa eta bertan normalean

```
graph TD; Activity[Activity] -- hosts --> FragmentStack[Fragment]; Activity --> FM[Fragment Manager]; FM -- "1. Activity or Fragment looking for a fragment" --> Find[Find Fragment]; Find -- "2. Searches fragments" --> FragmentStack; FragmentStack -- "3. Found requested fragment" --> Find; Find -- "4. The found fragment" --> Activity; FragmentStack --> FM; FM --> Activity;
```

The diagram illustrates the workflow of the Fragment Manager. It features four main components: Activity, Fragment Manager, Find Fragment, and Fragment. The process follows these steps:

- 1. Activity or Fragment looking for a fragment**: A dashed green arrow points from the Activity or Fragment to the Find Fragment component.
- 2. Searches fragments**: A dashed green arrow points from the Find Fragment component to the Fragment component.
- 3. Found requested fragment**: A dashed green arrow points from the Fragment component back to the Find Fragment component.
- 4. The found fragment**: A dashed green arrow points from the Find Fragment component back to the Activity component.

Additional solid blue arrows show the following relationships:

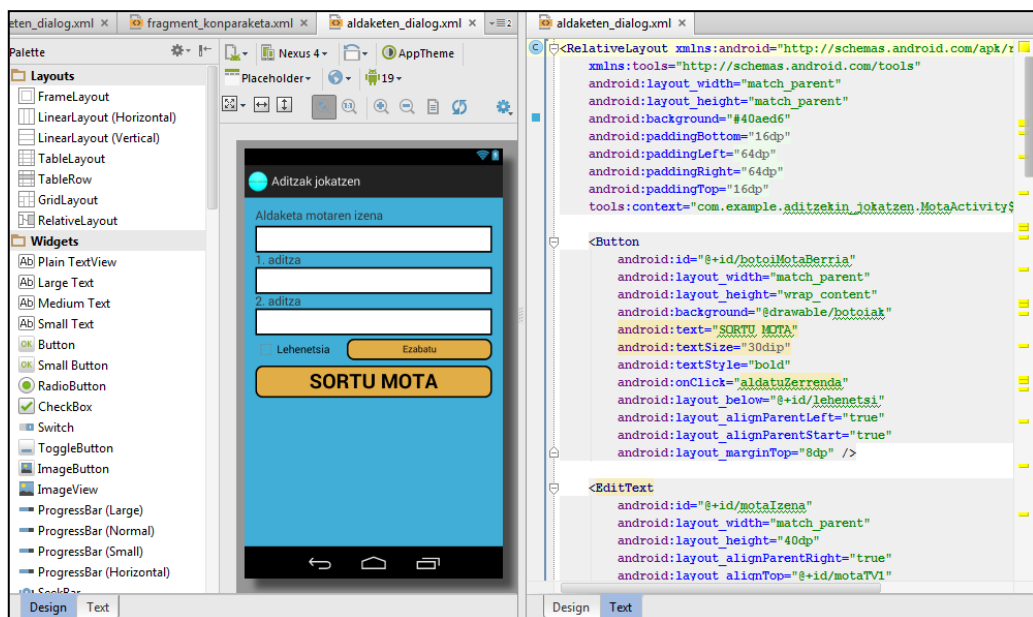
- Activity hosts the Fragment component.
- Activity is connected to the Fragment Manager.
- The Fragment Manager is connected to the Activity.
- The Fragment component is connected to the Fragment Manager.



Ondoren onStart() eta onResume() metodoak aurkitzen ditu eta aplikazio honen garapenean azkeneko metodo horretan egin dira hasieratzeak. Amaieran nahi izanez gero suntsitu egin daiteke finalize() metodoaren bidez eta orduan onDestroy() metodora jotzen du balin baldin badauka.

Aktibitate bakoitza sortu eta desagertu arteko prozesua honela egikaritzen da eta aktibitate berri bat irekitzen den bakoitzean beti prozesu bera egingo du hasieratik.

- R (Resource): Androiden datuen erroa da, baliabideak biltegiratzeko. Zuhaitz bezala funtzionatzen du eta aplikazioko elementu guztiak honen barruan daude. Adibidez, elementuak R/id nodoaren barnean gordetzen dira edota diseinuak 'R/layout' nodoan ('R/ layout-land' diseinu horizontalerako) edota irudiak 'R/drawable' izeneko nodoan. Hauen izenak ez dira aldatzea komeni modu lehenetsian horrela sortzen baita Androideko proiektua eta manifestuan aldaketarik egin ezean nodo hauek bilatuko ditu.
- Diseinuak (Layout): R/Layout nodoan kokatua, interfazeen diseinuak egon ohi dira hemen. Diseinu bakoitza .xml fitxategi bat da etab aktibitate bati edo gehiagori lotuta egon daiteke.



9.irudia: Aplikazioko diseinu baten adibidea

Diseinu hauek nahi izanez gero testuinguru bati lotu daitezke, zehazki nahi den testuinguruan soilik erabili ahal izateko

Hala ere esan bezala Android oso mundu zabala da eta erabili direnez gain elementu interesgarri askoz gehiago daude Androideko garapenarekin lotuta.

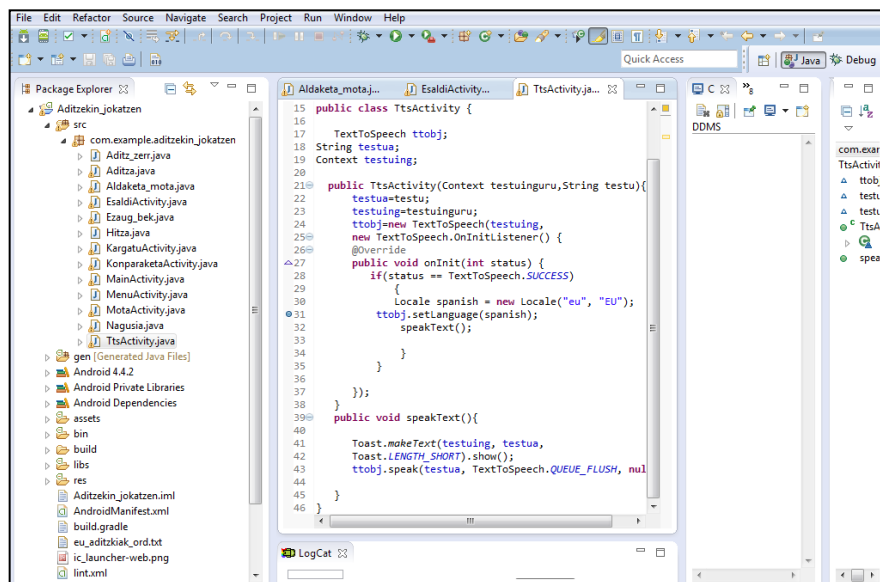
### 3.2.2 Garapenerako tresnak

Android aplikazio bat egiterako orduan aldeztu aurretik tresnak zehaztea komenigarria da. 2013 urtera arte, Android garatzaileek *Eclipse* ingurunea (*IDE*) erabiltzen zuten baina orduetik Google enpresak *Android Studio* izeneko programa bat garatu du, androideko aplikazioak egiteko espezifikoki sortua.

#### 3.2.2.1 Eclipse

Ingurune honetan mota askotako garapena egin daitezke, 10 programazio lengoai baino gehiagorekin. Kodea idazteko, optimizatze edota kode automatikoak sortzeko erraztasunak ematen ditu eta oso erabilgarria da programa sinple edota konplexuak sortzeko. Programa azkarra da exekuzioak martxan jartzean eta arazketak egiteko oso modu interesgarria eskaintzen du.

Ingurune hau Androidekin lotzeko, beharrezkoa da ADT (Android Developer Tools) plugina instalatzea bertan eta modu horretan diseinu eta kodearekin jokatzeko aukera ematen du interfaze interesgarri batean.



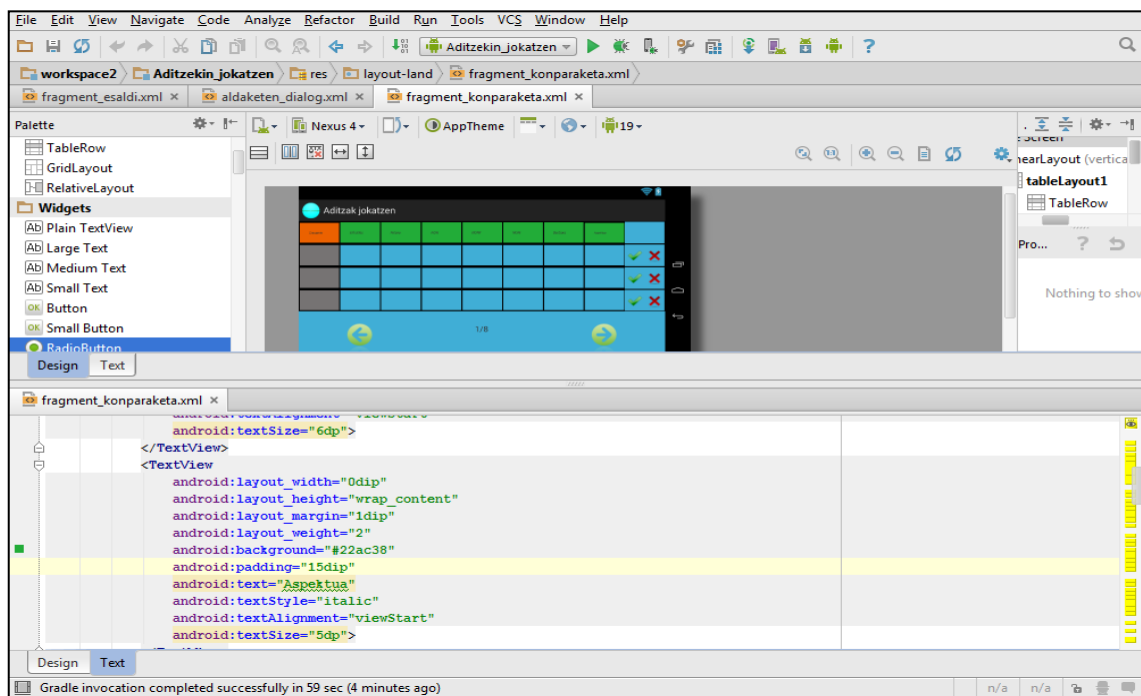
10.irudia: Aplikazioa Eclipse ingurunean garatzen

ADT pluginak ordea, garatzeko aukera ona ematen badu ere, diseinatzeak ez da hain ona, eta esaterako ezin dira elementu bat baino gehiagori aldaketa bera aldi berean aplikatu, izen berdina jarri *layout* ezberdinetan edota diseinuaren interfazea eta xml orria aldi berean ikusi leihua zatituta.

Laburbilduz, kodea garatzeko eroso da baina interfazeak diseinatzeko aukera hobeak badaude.

### 3.2.2.2 Android Studio

2013. urtean atera zuen Android Studioren lehen bertsio egonkorra Google-ek eta zehazki Android aplikazioak garatzeko sortu zen, Android beraiek sortua baita. Programa hau, Eclipserekin alderatuta oso ezberdina da garatzeko orduan eta ohitu arte nahiko deserosoa izan daiteke Eclipserekin ohitu den garatzaile batentzat. Aldiz, diseinuak egiteko tresna oso interesgarria dauka, interfazean atributuak modu argiago baten erakutsiz eta elementu bat baino gehiagori aldaketak automatikoki aplikatuz adibidez. Gainera, leihoak zatitu daitezkeenez, fitxategi bera modu bat baino gehiagotan ikus daiteke.



11.irudia: Aplikazioa Android Studio ingurunean diseinatzeko

Beste alde txar bat eraginkortasunean dago, izan ere proiektuak kargatzen denbora asko behar du eta programa beste zerbaitetan lanean ari bada, nekez egin dezake beste eragiketarik. Programa oro har motela da Eclipserekin alderatuta.

Laburbilduz, diseinuak egiteko oso interesgarria da baina Javarekin lan egitea neketsua egin dakioke beste ingurune bat erabiltzen ohituta denari.

### 3.2.3 Aukeratutako tresnak

Proiektua hasieran Eclipsen garatu eta diseinatzen hasi zen ADT 22.0 bertsioarekin, garatzailea ingurune horretara ohituta baitago eta Javarako ingurune oso egokia delako.

Denbora aurrera joan ahala ordea, Eclipse ingurunea diseinatzeko oso eskasa dela ikusi eta Android Studio probatu zen baina bertan Javarekin lan egitea askoz motelagoa dela ikusirik, bien arteko fusio bat egitea erabaki da. Alde batetik, Javaren kodea eta garapena Eclipse ingurunean egin eta bestetik diseinuak Android Studiorekin egitea eta Maiatzetik aurrera hala egin da denbora aurreztu eta eraginkortasuna hobetzearen itxaropenarekin.

Garapenerako tresnaz gain, beste batzuk ere erabili dira proiektuan zehar dokumentazioa Microsoft Word 2007n egin da, Gantt diagramak egiteko Gantt Project programa erabili da eta diagramak egiteko Dia Diagram Editor eta kudeaketako grafikoak egiteko Microsoft Excel 2007.

Programa probatu eta arazketak egiteko berriz zuzeneko konexio bat egin da USB bidez Motorola Moto G mugikor batekin, Android 4.4.2 bertsioa duena.

Azkenik, proiektu guztia Windows 7 Home Edition konputagailu batean garatu da.

## 4. Proiektuaren kudeaketa

---

### 4.1 Proiektuaren garapena eta esfortzuen hausnarketa

Proiektuaren kudeaketa nahiko egokia izan dela esan daiteke, izan ere nahiz eta epe batzuetan zertxobait atzean geratu, oro har epeen barruan murgildu da lana eta ataza batzuk espero baino azkarrago ere amaitu dira.

Epeetan ibiltzearen arrazoi nagusia proiektuaren helburu guztiak hasieran ez finkatzea izan da. Proiektu zuzendariak eskatutako eginkizun minimoak (ADAKo praktika Androidera ematea eta anbiguotasuna kontrolatzea) ataza sinpleak ematen zuten eta nahiz eta hasieran hobekuntza posibleak zehaztu, zein hobekuntza zehatz egingo ziren ez zen adierazi eta hobekuntza mailaren arabera sailkapen bat egin.

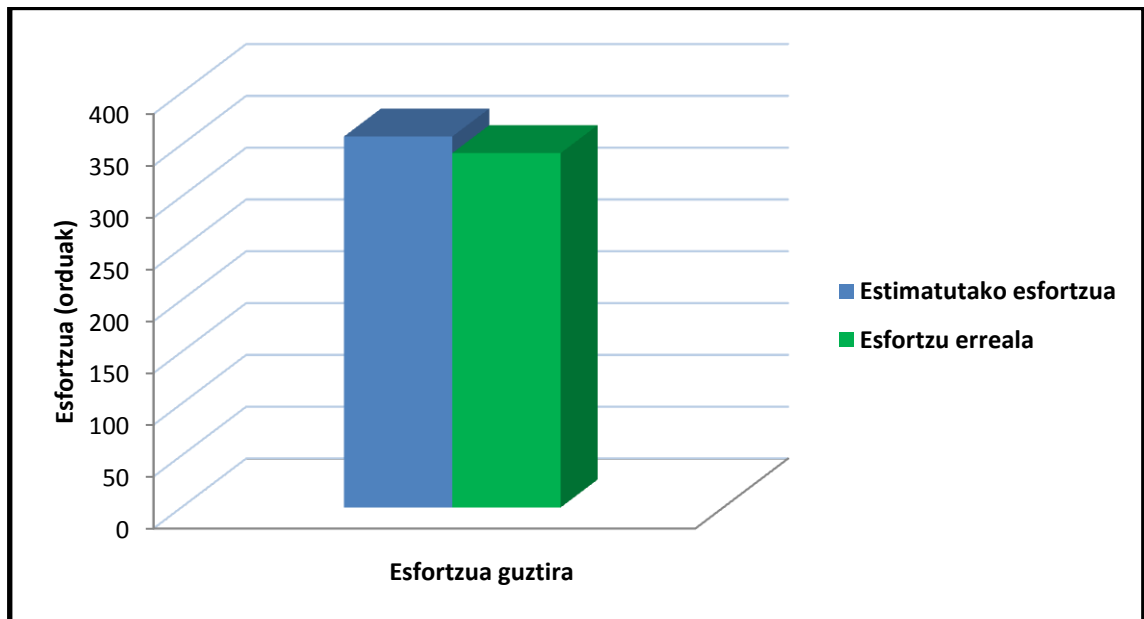
Erabaki hau proiektua amaitzeko dagoen epearengatik (2014ko Uztailak 1) hartu zen, eta asmoa denboraz ondo ibiliz gero hobekuntza mailari jarraituz, eginkizun hobeak egiten joatea izan zen eta horrela egin da, hobekuntza posible ugari inplementatu direlarik.

Egin den esfortzua orokorrean, uste baino zerbait txikiagoa izan da, alde batetik erabilpen kasuak eta erabiliko den arkitektura nahikoa azkar finkatu direlako eta analisi eta diseinuan ez delako aldaketa handirik egon proiektuaren garapenean zehar. Bestetik, arrazoi pertsonalengatik, garatzaileak 7 egunez jarraian ezin izan dut proiektua garatu eta horrek egun horietan espero zen esfortzua egin ezin izatea ahalbidetu du. Azkenik, proiektuaren amaiera apur bat aurreratu da dokumentazioan irakasleari memoriaren bertsio oso bat azkarrago emateko eta zuzenketak egin beharko balitz zuzentzeko aukera izateko epea ia 10 egunez aurreratuz. Hala ere hortik epea iritsi arte proiektuaren inplementazioa txukuntzeko eta ahal balitz hobekuntza txikiren bat egiteko asmoa dago.

Laburbilduz, proiektuaren kudeaketa egokia izan da, batez ere epe baten muga jarrita zegoelako proiektu amaierarako eta hori guztiz errespetatu da.

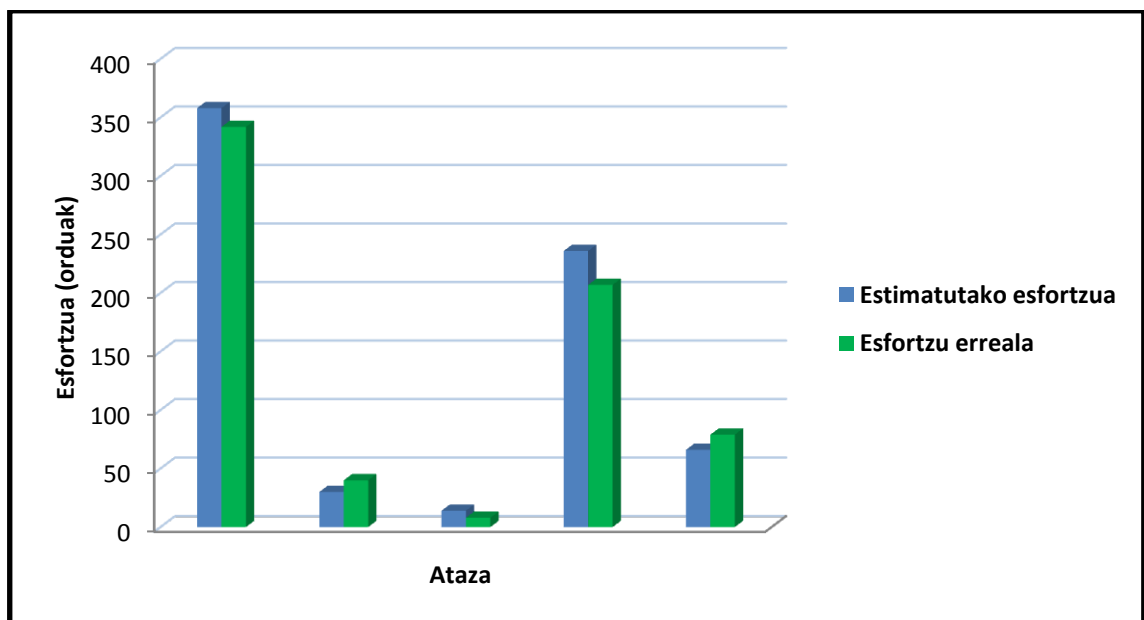
### 4.2 Esfortzuen hausnarketa

Proiektuaren hasieran, egingo zen esfortzuaren estimazio bat egin zen eta hau 357 ordukoa izan zen. Proiektu amaieran estimatutako esfortzu horretatik gertu ibili da esfortzu erreala, hau 242 ordukoa izan delarik. Honako grafiko honetan ikus daiteke konparaketa:



12.irudia: Esfortzuaren estimazioa eta egindako errealaren konparaketa

Honako grafiko honen bidez berriz, ataza orokorretan egin den esfortzu erreala, proiektu hasieran estimatutakoarekin konparatuta irudikatzen da:



13.irudia: Esfortzuaren estimazio eta egindako errealaren konparaketa atazen arabera

Azken grafiko honetan ikus daitekeenez, ataza orokorreko desfaseak ez dira handiak izan . Kudeaketa egiterako garaian espero baino esfortzu gutxiago behar izan dira. Formatzerakoan berriz, Androidek Javarekin dituen berezitasunak uste baino gehiago direnez denbora gehiago eskatu du formatzerakoan eta inplementatzeko garaian ere

Androiden inguruan formatzera itzuli behar izan da aldi batzuetan. Planifikazioa ere estimatutakoa baino esfortzu gutxiagorekin egin da eta dokumentatzen berriz hasieran pentsatzen zena baino 13 ordu gehiago egin behar izan dira.

Datu hauek guztiak eta gehiago taula honetan ikus daitezke ataza eta azpi-ataza bakoitzeko esfortzuak adieraziz

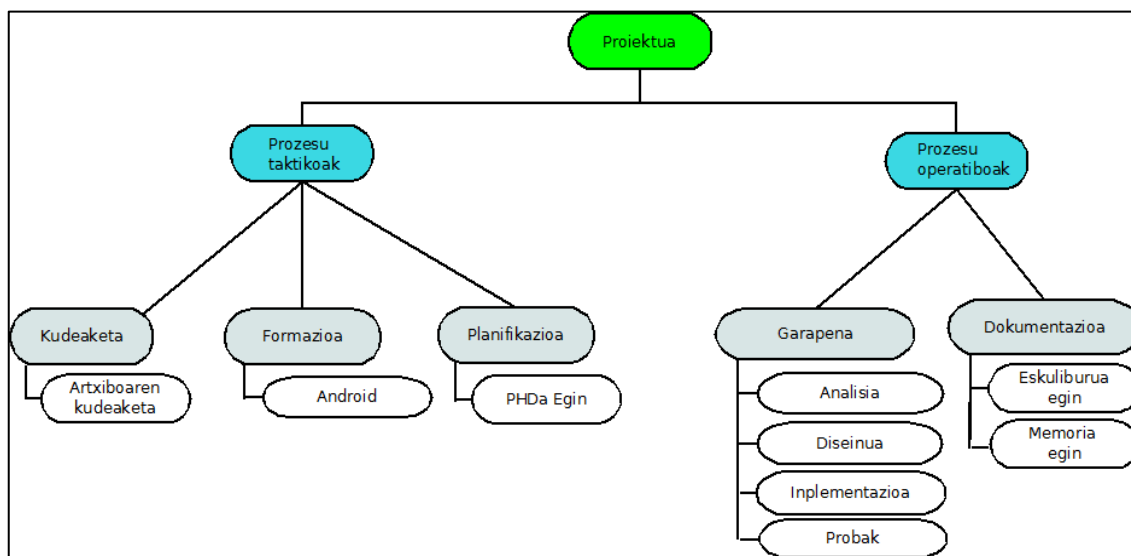
| Atazak                                   | Estimatutako esfortzua | Esfortzu erreala |
|------------------------------------------|------------------------|------------------|
| <b>K:Kudeaketa</b>                       | <b>12:00</b>           | <b>8:00</b>      |
| <b>K2: Artxiboaren kudeaketa</b>         | 12:00                  | 8:00             |
| <b>F:Formazioa</b>                       | <b>30:00</b>           | <b>40:00</b>     |
| <b>F1: Androiden formakuntza</b>         | 30:00                  | 40:00            |
| <b>P:Planifikazioa</b>                   | <b>14:00</b>           | <b>8:00</b>      |
| <b>P1: PHD-a egin</b>                    | 14:00                  | 8:00             |
| <b>G:Garapena</b>                        | <b>236:00</b>          | <b>207:00</b>    |
| <b>G1: Analisia</b>                      | 25:00                  | 15:00            |
| <b>G11: Analisi Funtzionala</b>          | 25:00                  | 15:00            |
| <b>G2: Diseinua</b>                      | 58:00                  | 40:00            |
| <b>G21: Aurkezpen maila diseinatu</b>    | 25:00                  | 20:00            |
| <b>G22: Datu maila diseinatu</b>         | 25:00                  | 15:00            |
| <b>G23: Probak diseinatu</b>             | 8:00                   | 05:00            |
| <b>G3: Implementazioa</b>                | 110:00                 | 125:00           |
| <b>G31: Arkitektura definitu</b>         | 15:00                  | 10:00            |
| <b>G32: Aurkezpen maila implementatu</b> | 25:00                  | 35:00            |
| <b>G33: Datu maila implementatu</b>      | 40:00                  | 60:00            |
| <b>G34: Androiden integratu</b>          | 30:00                  | 20:00            |
| <b>G4: Probak</b>                        | 43:00                  | 27:00            |
| <b>G41: Arazketak egin</b>               | 25:00                  | 15:00            |
| <b>G42: Probak egin</b>                  | 10:00                  | 8:00             |

|                             |               |               |
|-----------------------------|---------------|---------------|
| <b>G43: Martxan jarri</b>   | 8:00          | 4:00          |
| <b>D:Dokumentazioa</b>      | <b>66:00</b>  | <b>79:00</b>  |
| <b>D2: Eskuliburua egin</b> | 6:00          | 4:00          |
| <b>D3: Memoria egin</b>     | 60:00         | 75:00         |
| <b>GUZTIRA</b>              | <b>358:00</b> | <b>342:00</b> |

## 4.3 LDE eta azpi-atazen zerrenda

### 4.3.1 Lanaren Deskonposaketa Egitura diagrama (LDE)

Proiektu honetan gauzatu den laneko deskonposaketaren egitura diagrama honetan irudikatzen da:



3.diagrama: Lanaren deskonposaketa egitura



### 4.3.2 Proiektuko ataza eta azpi-atazak

Honakoak izan dira proiektuaren garapenean zehar egin diren ataza eta beraien azpi-atazak.

#### 4.3.2.1 Ataza taktikoak

K: Kudeaketarekin lotutakoak

K1: Artxiboaren kudeaketa

F: Formazioarekin lotutakoak

F1: Androiden formakuntza

P: Planifikazioarekin lotutakoak

P1: Proiektuaren Helburu Dokumentua egin

#### 4.3.2.2 Ataza operatiboak

G: Garapenarekin lotutakoak

G1: Analisia

G11: Analisi funtzionala

G2: Diseinua

G21: Aurkezpen maila diseinatu

G22: Datu maila diseinatu

G22: Probak diseinatu

G3: Implementazioa

G31: Arkitektura definitu

G32: Aurkezpen maila implementatu

G33: Datu maila implementatu

G34: Androiden integratu

G4: Probak

G41: Arazketak egin

G42: Probak egin

G43: Martxan jarri

D: Dokumentazioarekin lotutakoak

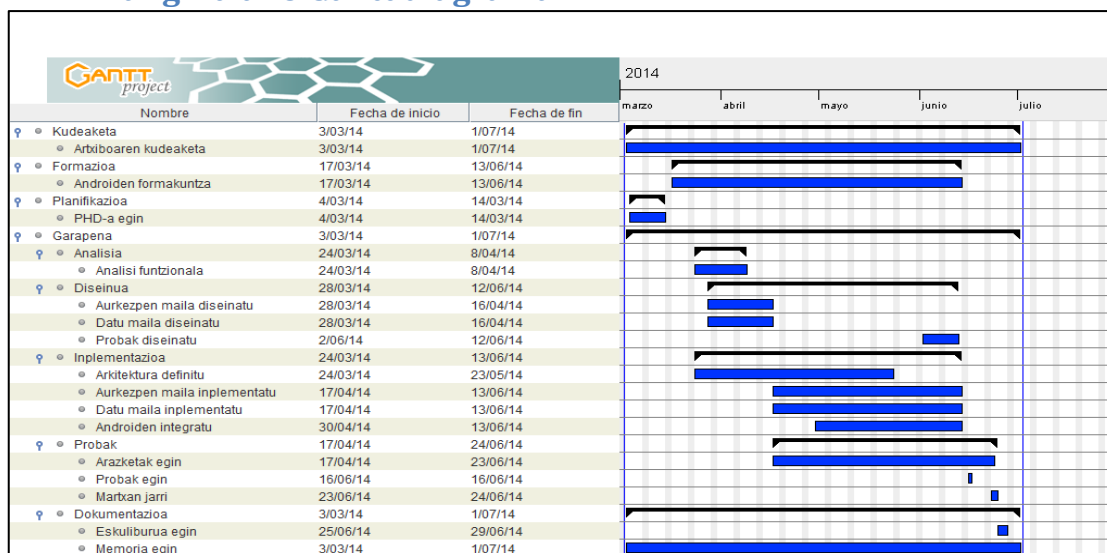
D1: Eskuliburua egin

D2: Memoria

## 4.4 Plangintzako Gantt VS Gantt erreala

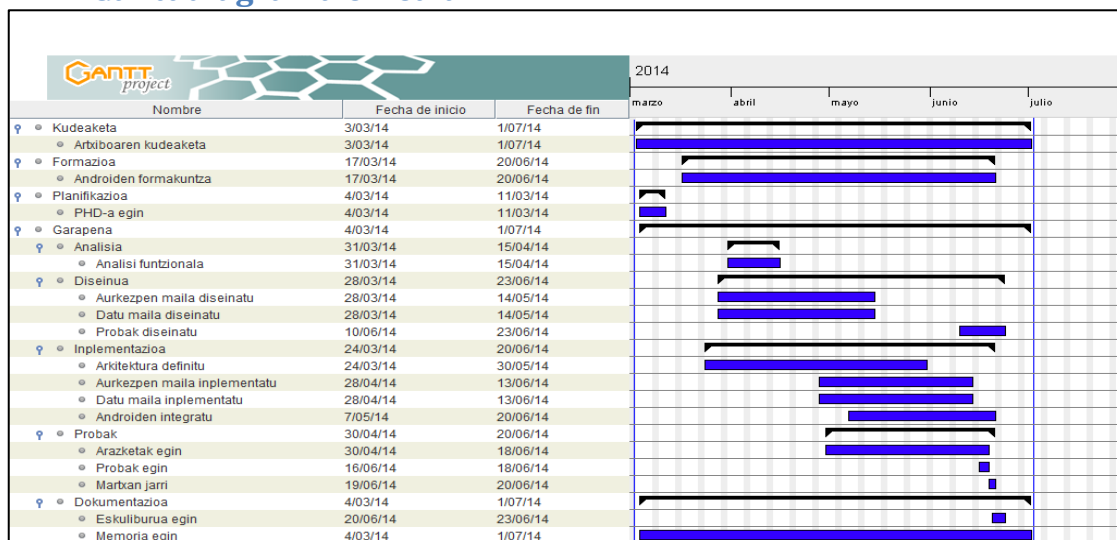
Proiektuaren hasieran estimatutako Gantt diagrama eta proiektuaren amaierako Gantt diagrama erreala aurkezten dira hemen:

### 4.4.1 Plangintzako Gantt diagrama



4. diagrama: Plangintzako Gantt

### 4.4.2 Gantt diagrama erreala



5. diagrama: Gantt erreala

Epeen bilakaera analizatzen bada, ikus daiteke gehiengoan espero zen epean joan dela gutxi gora behera baina analisi eta diseinua beranduago amaitu dira, batez ere implementatzearekin batera zehaztu direlako bereziki eta nahiz eta espero bezala hasi, uste baino beranduago amaitu da. Hala ere, honek ez du esan nahi esfortzu gehiago egin denik, epeetan beranduago ibili dela baizik.

## 4.5 Eginkizunen hausnarketa

Proiektua planifikatzerako garaian, proiektuan zehar egin beharreko eginkizun minimoak eta hobekuntza posibleak azaldu ziren. Ahalik eta hobekuntza gehien egiten saiatu da baina guztiak ezin izan dira egin, baliabide faltagatik eta batez ere, aurredefinitutako epean ezin izan delako egin.

### 4.5.1 Egindako eginkizunak

Proiektuan zehar amaitu diren eginkizunak taula honetan adierazten dira hobekuntza maila altuenetik baxuenera ordenatuta:

| <b>Eginkizuna</b>                                 | <b>Azalpena</b>                                                                                                                        | <b>Hobekuntza-maila</b> |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| <b><i>Adako praktika Androidera eraman</i></b>    | Adan ‘Oinarrizko programazioa’ ikasgaiaren eginiko praktika Androiden inplementatzea, hau da 3 adizki jarrita adizki emaitza itzultzea | Egin beharreko minimoa  |
| <b><i>Adizkien anbiguotasuna tratatu</i></b>      | Sartzen den 3 adizkietakoren bat anbigua bada, hori ere tratatu eta emaitzan sartu                                                     | Egin beharreko minimoa  |
| <b><i>Esaldi osoak bihurtzea</i></b>              | Adizki bakarra pasa ordeez esaldi osoa pasa eta bihurtzea                                                                              | Hobekuntza baxua        |
| <b><i>Emaitzak entzutea</i></b>                   | Adizkiak ahotsez esateko aukera eta lortzen diren emaitzak bozgorailu bidez entzuteko aukera izatea                                    | Hobekuntza baxua        |
| <b><i>Interfaze eleanitza</i></b>                 | Interfazea eleanitza izatea eta gutxienez euskara, gaztelania eta ingeleserako interfazea sortzea                                      | Hobekuntza ertaina      |
| <b><i>Aldaketa motak sortzea</i></b>              | 3 adizki jartzen ibili beharrean, erabiltzaileak bere aldaketa mota propioak jartzeko aukera izatea lehenengo bi aldaketa motak emanez | Hobekuntza ertaina      |
| <b><i>Emaitzak konparatzeko aukera izatea</i></b> | Ateratzen den emaitza anbigua bada, beraien arteko konparaketa ikustea taula bidez                                                     | Hobekuntza ertaina      |
| <b><i>Interfaze horizontala</i></b>               | Erabiltzaileak nahi badu, interfazea modu horizontalean ikusteko aukera ematea gailua iraultzen denean                                 | Hobekuntza ertaina      |
| <b><i>Adizkiak desanbiguatzea</i></b>             | Erabiltzailea adizki anbiguo bat idatzi orduko jakinarazi eta desanbiguatzeko aukera ematea                                            | Hobekuntza altua        |
| <b><i>Aldaketa motak kudeatzea</i></b>            | Aldaketa motak sortzeaz gain, berauek ezabatu, aldatu edota lehenesteko aukera                                                         | Hobekuntza altua        |

|                                        |                                                                   |                  |
|----------------------------------------|-------------------------------------------------------------------|------------------|
| <b>Tabletarako bateragarria izatea</b> | Tabletatarako diseinu espezifikoak egin pantaila oso handietarako | Hobekuntza altua |
|----------------------------------------|-------------------------------------------------------------------|------------------|

#### 4.5.2 Egin gabeko eginkizunak

Aldiz, taula honetan adierazten dira egin gabe geratu diren eginkizunak, hobekuntza maila baxuenetik altuenera ordenatuta eta eginkizuna ez egiteko izan diren arrazoia.

| <b>Eginkizuna</b>                           | <b>Azalpena</b>                                                                                                                                                                  | <b>Hobekuntza-maila</b> | <b>Arrazoia</b>                                                                                              |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------|
| <b>Lematizazioa erabiltzea</b>              | Esaldiak bihurtzerako orduan banan banako bilaketa egin ordez esaldia lematizatzea emaitza hobeak lortzeko.                                                                      | Hobekuntza altua        | Epe barruan ezin izan da egin                                                                                |
| <b>Hizkuntza ezberdinetara moldatzea</b>    | Euskarazko adizkiez gain, gaztelania eta ingelesezko aditz irregularrekin ere balio izatea aplikazioak                                                                           | Hobekuntza altua        | Eskuratutako gaztelania eta ingelesezko datu-baseak zeharo txikiak eta gainera epe barruan ezin izan da egin |
| <b>Herrien arteko hizkuntzak aplikatzea</b> | Webgune batean gehien erabiltzen diren adizkiak jarri eta erabiltzaileek beraien herrian nola esaten den idaztea, honela aldaketa mota herri baten arabera aplikatu ahal izateko | Hobekuntza oso altua    | Epe barruan ezin izan da egin                                                                                |

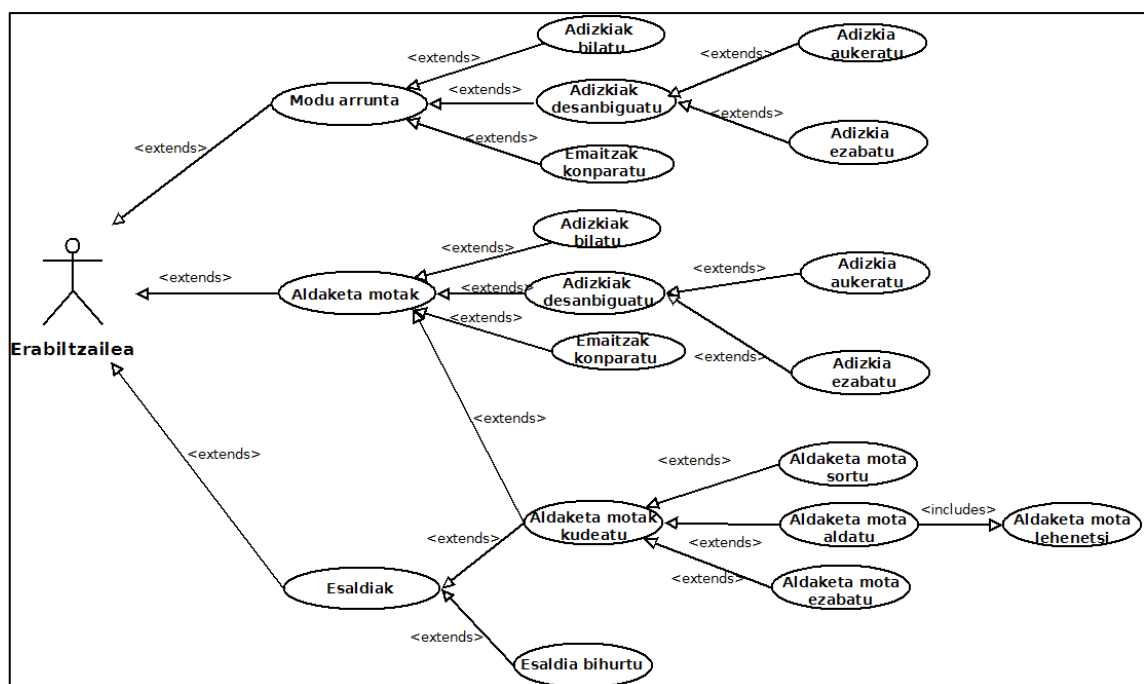
# 5. Garapen teknikoa

## 5.1 Eskakizunen bilketa

### 5.1.1 Erabilpen Kasuen Eredua (EKE)

#### 5.1.1.1 Deskribapen zehatzak

Honakoa izan da proiektu honetan erabili den Erabilpen Kasuen Eredua



6.diagrama: Erabilpen Kasuen Eredua

#### 1. Modu arrunta

Modu arruntean erabiltzaileak adizkiak bilatu, adizkiak desanbiguatu eta emaitzak konpara ditzake.

##### 1.1 Adizkiak bilatu

Hiru adizki emanda, adizki horien ezaugarrien arteko analogia erabiliz beste adizki bat edo gehiago lortzen ditu sistemak. Kontuan hartu behar da adizki bat anbigua izan daitekeela eta interfazeaz arrisku ikur batekin ohartarazte da kasu hauetan.

## **1.2 Adizkiak desanbiguatu**

Adizki batek ezaugarri (Lema,hitanoa,nor,nori,nork, denbora eta aspektuak osatzen dute) ezberdin bat baino gehiago baditu , adizkiak aukeratu edo ezabatu egin daitezke konparaketa-taula baten bitartez non orrialde bakoitzean gehienez 3 adizkiren informazioa gorde daitekeen.

### **1.2.1 Adizkia aukeratu**

Aukeratzen den adizkia automatikoki bere ezaugarriekin, aukera bakarra izatera pasatzen da eta automatikoki sistema aurreko leihora joaten da bertan adizkia desanbiguatuta utziz.

### **1.2.2 Adizkia ezabatu**

Aukeratzen den adizkia konparaketa-taulatik ezabatzen da eta taula eguneratzen da ezberdintasunak ikusteko

## **1.3 Emaitzak konparatu**

Emaiza anbigua bada, hots, emaitzaren ezaugarri bera adizki bat baino gehiago balute, konparaketa taula baten bitartez beraien ezaugarriak agertzen dira, ezberdinak diren ezaugarriak koloreztatuta adieraziz. Orrialde bakoitzean gehienez 3 adizkiren informazioa gorde daiteke.

## **2. Aldaketa motak**

Modu honetan erabiltzaileak, adizkiak bilatu, adizkiak desanbiguatu,emaitzak konparatu eta aldaketa motak kudeatu ditzake

### **2.1 Adizkiak bilatu**

*Ikus 1.1*

### **2.2 Adizkiak desanbiguatu**

*Ikus 1.2*

### **2.3 Emaitzak konparatu**

*Ikus 1.3*

### **2.4 Aldaketa motak kudeatu**

Erabiltzaileak aldaketa mota bat sortu, aldatu , ezabatu edo lehenetsi dezake.

#### **2.4.1 Aldaketa mota sortu**

Aldaketa mota sortzeko aldaketa motaren izena, lehenengo adizkia eta bigarren adizkia ematen zaizkio eta aldaketa mota lehenetsia izango den ala ez adierazi. Lehenetsia balitz, aldaketa moten zerrenda eguneratzen da sortu berri den aldaketa zerrendan lehenengoa jarriz. Horrela, hurrengo saio hasieran aldaketa mota lehenetsia agertuko da zerrendaren hasieran.

#### **2.4.2 Aldaketa mota aldatu**

Aldaketa mota aldatzeko aldaketa mota aukeratu eta aldaketa motaren izena, lehen adizkia, bigarren adizkia eta lehenetsia den edo ez erakusten du sistemak. Nahi diren aldaketak egin daitezke, baldin eta izen hori existitzen ez bada dagoeneko aldaketa moten zerrendan.

#### **2.4.3 Aldaketa mota ezabatu**

Aldaketa mota aukeratzeko eskatzen da eta interfazeaz ezabatzeko aukera ematen duen botoi bistaratu du sistemak. Hau sakatzeaz, aukeratutako aldaketa mota ezabatzen da eta aldaketa mota zerrendatik ezabatzen da. Ezabatzen dena lehenetsia bada, automatikoki zerrendako hurrengoa lehenesten da.

### **3. Esaldiak**

#### **3.1 Aldaketa motak kudeatu**

*Ikus 2.4*

##### **3.1.1 Aldaketa mota sortu**

*Ikus 2.4.1*

##### **3.1.2 Aldaketa mota aldatu**

*Ikus 2.4.2*

##### **3.1.3 Aldaketa mota ezabatu**

*Ikus 2.4.3*

#### **3.2 Esaldia bihurtu**

Erabiltzaileak bi esaldi pasata eta aldaketa mota bat aukeratuta, hitz bakoitza analizatzen da; hitz horri aukeratutako aldaketa motara bihurtzeaz hiru kasu gerta daitezke:

1. Ez dago aldaketa motaren emaitzarik hitza ez delako datu-basean aurkitu edota aldaketa mota horri dagokion emaitzarik ez dagoelako. Kasu honetan zegoen bezala uzten da.
2. Emaidza existitzen da eta bakarra da. Kolore berdez idazten da emaitza.
3. Emaidza existitzen da eta aukera bat baino gehiago daude. Kolore arrosez idazten dira emaitza posibleak '/' banatzailearekin.  
Honetaz gain, TTS(Text To Speech) bidez ere entzuten da emaitza erabiltzailearen bozgorailuak martxan badaude. Emaidzan adizki anbiguoak agertzen badira, lehenengoa soilik entzuten da.

#### 5.1.1.2 Gertaera-fluxuak

##### 1. Modu arrunta

##### 1.1 Adizkiak bilatu

*Gertaera-fluxu normala*

1. **Erabiltzailea:** 3 adizki ematen ditu.
2. **Sistema:** Analogia bidez adizki emaitza edo emaitzak itzultzen ditu atzeko plano berde batez.

*Gertaera-fluxu alternatiboa*

2.urratsa: Ez dago eskaerarekin bat datorren emaitzarik.

1. **Sistema:** Emaitzarik ez dela adierazten da atzeko plano gorriarekin.

##### 1.2 Adizkiak desanbiguatu

##### 1.2.1 Adizkia aukeratu

*Gertaera-fluxu normala*

1. **Sistema:** Konparaketa-taulan anbiguotasunak erakusten ditu
2. **Erabiltzailea:** Adizkiaren zutabea aukeratu eta aukeratzeko eskatzen du
3. **Sistema:** Adizkiaren aukera eginda, bere anbiguotasuna kendu eta aukeratutako aukeraren ezaugarriak aplikatzen zaizkio adizkiari. Gero, automatikoki aurreko leihora itzultzen da datu berriak pasatuz.

##### 1.2.2 Adizkia ezabatu

*Gertaera-fluxu normala*

1. **Sistema:** Konparaketa-taulan anbiguotasunak erakusten ditu
2. **Erabiltzailea:** Adizkiaren zutabea aukeratu eta ezabatzeko eskatzen du
3. **Sistema:** Adizkiaren zutabea taulatik ezabatzen da eta konparaketa taula eguneratzen du. Gainera, adizkiaren anbiguotasun datu berriak eguneratzen ditu.



*Gertaera-fluxu alternatiboa*

3. urratsa: Eguneratu ondoren adizki bakarra dago , beraz desanbiguatuta.

1. **Sistema:** Ezabatzeko botoia ezkututzen da, desanbiguatuta baitago.

### 1.3 Emaizak konparatu

*Gertaera-fluxu normala*

**1. Erabiltzailea:** Emaizen konparaketa eskatzen du

**2. Sistema:** Konparaketa-taula batez emaitzak eta beraien ezaugarriak adierazten ditu , ezaugarri ezberdinak dituztenak kolore ezberdinez agertzen direlarik.

*Gertaera-fluxu alternatiboa*

1. urratsa: Eguneratu ondoren adizki bakarra dago , beraz desanbiguatuta.

1. **Sistema:** Emaiza bakarra badago, ezin da konparaketa egin eta ez du ezer egiten.

## 2. Aldaketa motak

### 2.1 Adizkiak bilatu

*Ikus 1.1*

### 2.2 Adizkiak desanbiguatu

*Ikus 1.2*

### 2.3 Emaizak konparatu

*Ikus 1.3*

### 2.4 Aldaketa motak kudeatu

#### 2.4.1 Aldaketa mota sortu

*Gertaera-fluxu normala*

**1. Erabiltzailea:** Aldaketa mota berri bat sortzeko eskatzen du

**2. Sistema:** Formulario bat azalarazten du izena, 2 adizki eta lehenetsia izango den ala ez idazteko.

**3. Erabiltzailea:** Formularioan aldaketa motaren izena, 1.adizkia, 2.adizkia eta lehenetsia izango den adierazten du

**4. Sistema:** Aldaketa mota gordetzen du erabiltzaileari mezu bat erakutsiz eta aldaketa moten zerrenda eguneratzen du.

*Gertaera-fluxu alternatiboa*

4. urratsa: Existitzen da erabiltzaileak pasatako aldaketa mota izena

1. **Sistema:** Mezu baten bidez ohartaraziko du izen hori existitzen dela dagoeneko.

4. urratsa: 1.adizkia edo 2.adizkia ez da existitzen datu-basean

1. **Sistema:** Mezu baten bidez ohartarazten du adizkiren bat ez dela existitzen

## 2.4.2 Aldaketa mota aldatu

### *Gertaera-fluxu normala*

1. **Erabiltzailea:** Aldaketa mota aukeratu eta aldatzeko eskatzen du
2. **Sistema:** Formulario bat azalarazten du aldaketa motaren izena, 2 adizki eta lehenetsia den edo ez esanez ezabatzeko botoi bat erakutsiz gainera.
3. **Erabiltzailea:** Formularioan aldaketak egiten ditu
4. **Sistema:** Aldaketa mota gordetzen du erabiltzaileari mezu bat erakutsiz eta aldaketa moten zerrenda eguneratzen du.

### *Gertaera-fluxu alternatiboa*

2.urratsa: Aukeratutako aldaketa mota 'hitanoa (toka) edo hitanoa (noka)' da

1. **Sistema:** Bi hauek aldatu ezin direnez, ezabatzeko botoia ezkutatzen da eta bere informazioa erakusten da aldatzeko aukerarik eman gabe

4.urratsa: Existitzen da erabiltzaileak pasatako aldaketa mota izena

1. **Sistema:** Mezu baten bidez ohartaraziko du izen hori existitzen dela dagoeneko.

4.urratsa: 1.adizkia edo 2.adizkia ez da existitzen datu-basean

1. **Sistema:** Mezu baten bidez ohartarazten du adizkiren bat ez dela existitzen

## 2.4.3 Aldaketa mota ezabatu

### *Gertaera-fluxu normala*

1. **Erabiltzailea:** Aldaketa mota aukeratu eta aldatzeko
2. **Sistema:** Formulario bat azalarazten du izena, 2 adizki eta lehenetsia izango den ala ez idazteko ezabatzeko botoiarekin batera
3. **Erabiltzailea:** Aukeratutako aldaketa mota ezabatzeko eskatzen du
4. **Sistema:** Aldaketa mota ezabatzen du eta aldaketa moten zerrendatik kentzen da erabiltzaileari mezu baten bidez adieraziz.

### *Gertaera-fluxu alternatiboa*

4.urratsa: Ezabatu nahi den aldaketa mota lehenetsia da

1. **Sistema:** Zerrendako hurrengo elementua jarriko du lehenetsi gisa.

## 3. Esaldiak

### 3.1 Aldaketa motak kudeatu

#### *Ikus 2.4*

#### 3.1.1 Aldaketa mota sortu

##### *Ikus 2.4.1*

#### 3.1.2 Aldaketa mota aldatu

## Ikus 2.4.2

### 3.1.3 Aldaketa mota ezabatu

## Ikus 2.4.3

### 3.2 Esaldia bihurtu

#### Gertaera-fluxu normala

**1. Erabiltzailea:** Aldaketa mota bat aukeratu eta testu kutxan bihurtu nahi duen esaldia idatzi ondoren, bihurtzeko botoia sakatzen du.

**2. Sistema:** Bihurketa egiten du hitzez hitz emaitzak kolore berdez idatziz eta bihurketa amaitutakoan TTS( *Text To Speech*) bidez esaten du .

#### Gertaera-fluxu alternatiboa

2.urratsa: Uneko hitza anbigua da

1. **Sistema:** Testu kutxan ‘/’ moduko karaktere banatzaile baten bidez kolore arrosarekin idazten dira aukerak. TTS bidez lehenengo emaitza soilik esaten da

2.urratsa: Uneko hitzak ez du emaitzarik

1. **Sistema:** Testu kutxan uneko hitza letra beltzez idazten du.

2.urratsa: Erabiltzaileak ez du bozgorailurik

1. **Sistema:** Ez da TTS bidez esandako entzungo.

## 5.2 Analisia

Analisiaren barnean sekuentzia-diagramak egin eta analizatu dira.

### 5.2.1 Sekuentzia diagramak

#### 1. Modu arrunta

##### 1.1 Adizkiak bilatu



7.diagrama: Adizkiak bilatu sekuentzia-diagrama

Izena: hitzaBihurtu(aditz1,aditz2,aditz3):emaitzak

Sarrera:

- aditz1 = ArrayList <Aditza>
- aditz2 = ArrayList <Aditza>
- aditz3 =ArrayList <Aditza>

Erantzukizunak: Hiru adizki pasa eta emaitza bezala adizki bat edo emaitza anbigua balitz adizki gehiago ematen ditu.

Aurrebaldintzak: Ø

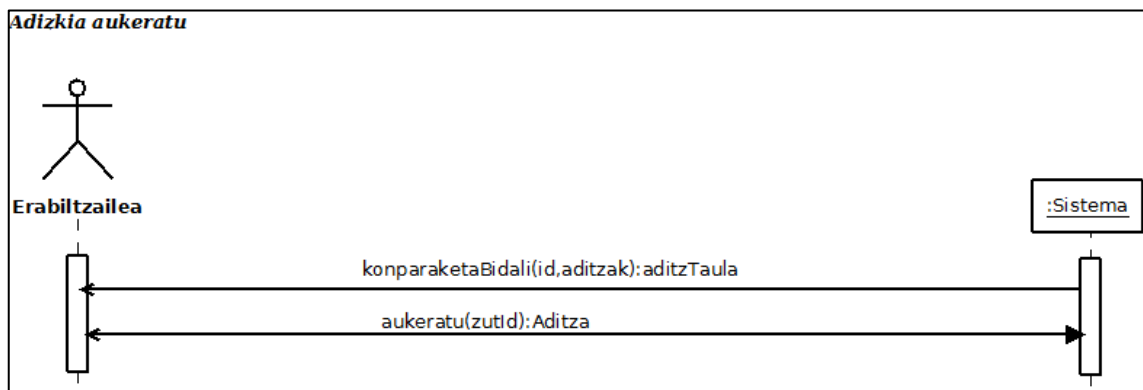
Postbaldintzak: Ø

Irteera:

- emaitzak =ArrayList <Aditza>

## 1.2 Adizkiak desanbiguatu

### 1.2.1 Adizkia aukeratu



8.diagrama: Adizkia aukeratu sekuentzia-diagrama

### Lehen eragiketa

Izena: konparaketaBidali(id,aditzak):aditzTaula

Sarrera:

- id = Integer
- aditzak = ArrayList <Aditza>

Erantzukizunak: Adizki anbiguo bat desanbiguatzeko eskatu ondoren, konparaketa taula bat erakusten da.

Aurrebaldintzak: \_id != 0

Postbaldintzak: aditzTaula.size() >= 2

Irteera:

- aditzTaula=TableLayout<Aditza>

### Bigarren eragiketa

Izena: aukeratu(zutId):aditza

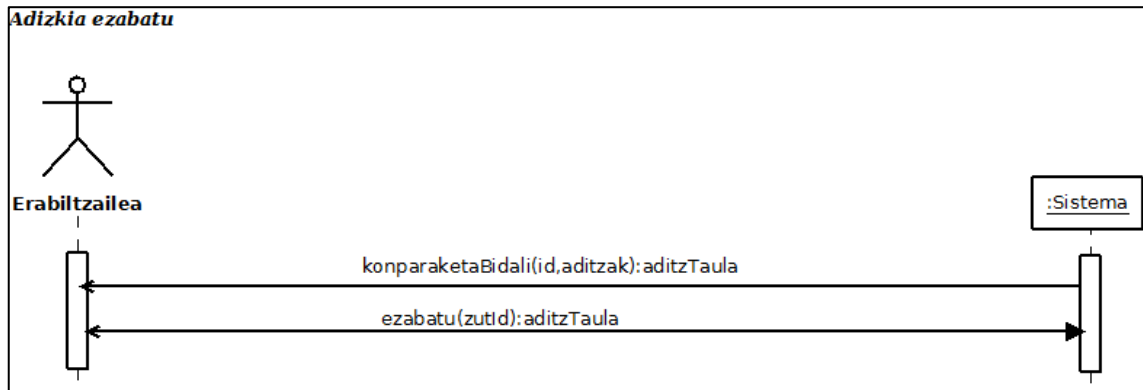
Sarrera:

- zutId = Integer

Erantzukizunak: Aukeratu nahi den zutabea pasa eta horren informazioa itzultzen du sistemak.

Aurrebaldintzak: aditzTaula.size() >= 2  
Postbaldintzak: Aditza desanbiguatuta dago  
Irteera:  
 - aditza=Aditza

### 1.2.2 Adizkia ezabatu



9.diagrama: Adizkia ezabatu sekuentzia-diagrama

#### Lehen eragiketa

Izena: konparaketaBidali(id,aditzak):aditzTaula

Sarrera:

- id = Integer
- aditzak = ArrayList <Aditza>

Erantzukizunak: Adizki anbiguo bat desanbiguatzeko eskatu ondoren, konparaketa taula bat erakusten da.

Aurrebaldintzak: id != 0

Postbaldintzak: aditzTaula.size() >= 2

Irteera:

- aditzTaula=TableLayout<Aditza>

#### Bigarren eragiketa

Izena: ezabatu(zutId):aditzTaula

Sarrera:

- zutId = Integer

Erantzukizunak: Aukeratu nahi den zutabea pasa eta aditz taulatik ezabatzen du sistemak gero taula eguneratuz.

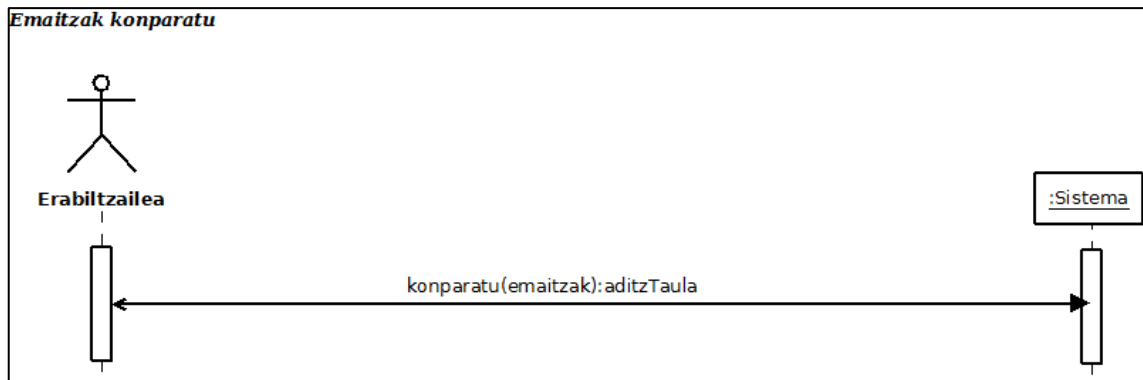
Aurrebaldintzak: aditzTaula.size() >= 2

Postbaldintzak: Aditza desanbiguatuta.

Irteera:

- aditzTaula = TableLayout<Aditza>

## 1.2 Emaizak konparatu



10.diagrama: Emaizak konparatu sekuentzia-diagrama

Izena: konparatu(emaizak):aditzTaula

Sarrera:

- emaizak = ArrayList <Aditza>

Erantzukizunak: Adizkiak bilatutako emaizak konparatzeko eskatu eta aditz taula batean erakusten da.

Aurrebaldintzak: emaizak.size() >= 2

Postbaldintzak: aditzTaula.size() >= 2

Irteera:

- aditzTaula=TableLayout<Aditza>

## 2. Aldaketa motak

### 2.1 Adizkiak bilatu

*Ikus 1.1 sekuentzia-diagrama*

### 2.2 Adizkiak desanbiguatu

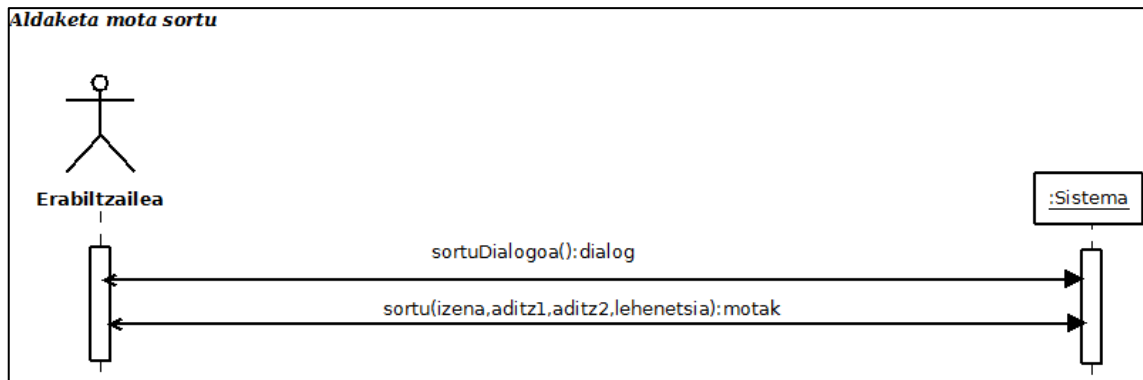
*Ikus 1.2 sekuentzia-diagrama*

### 2.3 Emaizak konparatu

*Ikus 1.3 sekuentzia-diagrama*

### 2.4 Aldaketa motak kudeatu

### 2.4.1 Aldaketa mota sortu



11.diagrama: Aldaketa mota sortu sekuentzia-diagrama

#### Lehen eragiketa

Izena: sortuDialogoa():dialog

Sarrera: Ø

Erantzukizunak: Erabiltzaileak mota berri bat sortzeko eskatu eta sistemak hori lortzeko dialogo bat irekitzen dio datuak sartzeko

Aurrebaldintzak: Ø

Postbaldintzak: Ø

Irteera:

- dialog=Dialog

#### Bigarren eragiketa

Izena: sortuDialogoa(izena,aditz1,aditz2,lehenetsia):motak

Sarrera:

- izena = String
- aditz1 = ArrayList <Aditza>
- aditz2 = ArrayList <Aditza>
- lehenetsia = Boolean

Erantzukizunak: Erabiltzaileak izena, 2 aditzak (desanbiguatuta edo ez) jarri eta lehenetsia den edo ez esatean, sistemak moten zerrendan txertatzen du.

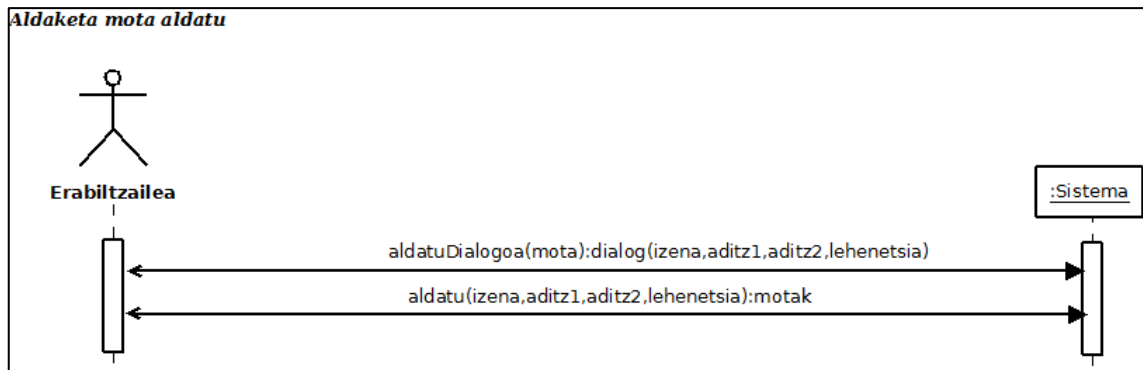
Aurrebaldintzak: Izena ez da existitzen moten zerrendan.

Postbaldintzak: motak.size() >= 3 (Noka eta toka beti)

Irteera:

- motak = ArrayList<Aldaketa\_mota>

## 2.4.2 Aldaketa mota aldatu



12.diagrama: Aldaketa mota aldatu sekuentzia-diagrama

### Lehen eragiketa

**Izena:** aldatuDialogoa(mota):dialog(izena,aditz1,aditz2,lehenetsia)

**Sarrera:**

- mota = Aldaketa\_mota

**Erantzukizunak:** Erabiltzaileak mota bat aukeratu ondoren aldatzeko eskatuko dio eta sistemak aldaketa motaren datuak erakutsiko dizkio dialogo batean.

**Aurrebaldintzak:** mota existitzen da

**Postbaldintzak:** mota.izena != "hitanoa(toka)" eta mota.izena != hitanoa(noka)

**Irteera:**

dialog(izena,aditz1,aditz2,lehenetsia)=Dialog(EditText, EditText, EditText, CheckBox)

### Bigarren eragiketa

**Izena:** aldatu(izena,aditz1,aditz2,lehenetsia):motak

**Sarrera:**

- izena = String
- aditz1 = ArrayList <Aditza>
- aditz2 = ArrayList <Aditza>
- lehenetsia = Boolean

**Erantzukizunak:** Erabiltzaileak izena, 2 aditzak (desanbiguatuta edo ez) jarri eta lehenetsia den edo ez esatean, sistemak mota eguneratzen du.

**Aurrebaldintzak:** motak existitzen da

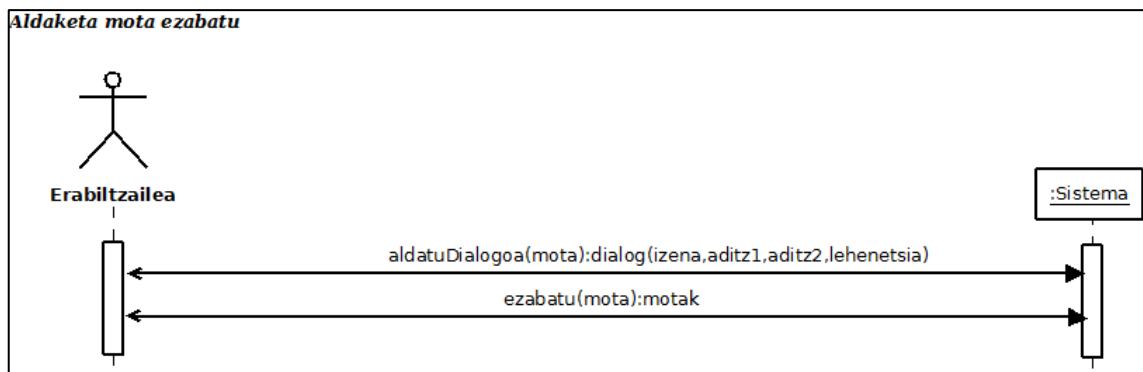
**Postbaldintzak:** motak.size() >= 3 (Noka eta toka beti) , Izena ez da existitzen moten zerrendan.

**Irteera:**

- motak = ArrayList<Aldaketa\_mota>



### 2.4.3 Aldaketa mota ezabatu



13.diagrama: Aldaketa mota ezabatu sekuentzia-diagrama

#### Lehen eragiketa

Izena: aldatuDialogoa(mota):dialog(izena,aditz1,aditz2,lehenetsia)

Sarrera:

- mota = Aldaketa\_mota

Erantzukizunak: Erabiltzaileak mota bat aukeratu ondoren aldatzeko eskatuko dio eta sistemak aldaketa motaren datuak erakutsiko dizkio dialogo batean.

Aurrebaldintzak: mota existitzen da

Postbaldintzak: mota.izena != "hitanoa(toka)" eta mota.izena != hitanoa(noka)

Irteera:

dialog(izena,aditz1,aditz2,lehenetsia)=Dialog(EditText, EditText, EditText, CheckBox)

#### Bigarren eragiketa

Izena: ezabatu(mota):motak

Sarrera:

- izena = String

- aditz1 = ArrayList <Aditza>

- aditz2 = ArrayList <Aditza>

- lehenetsia = Boolean

Erantzukizunak: Aldaketen dialogoa uneko motaren izena duen mota ezabatzeko eskatzen du eta sistemak moten zerrenda eguneratzen du.

Aurrebaldintzak: motak existitzen da

Postbaldintzak: motak.size() >= 2 (Noka eta toka beti)

Irteera:

- motak = ArrayList<Aldaketa\_mota>

## 3. Esaldiak

### 3.1 Aldaketa motak kudeatu

*Ikus 2.4 sekuentzia-diagrama*

### 3.1.1 Aldaketa mota sortu

*Ikus 2.4.1 sekuentzia-diagrama*

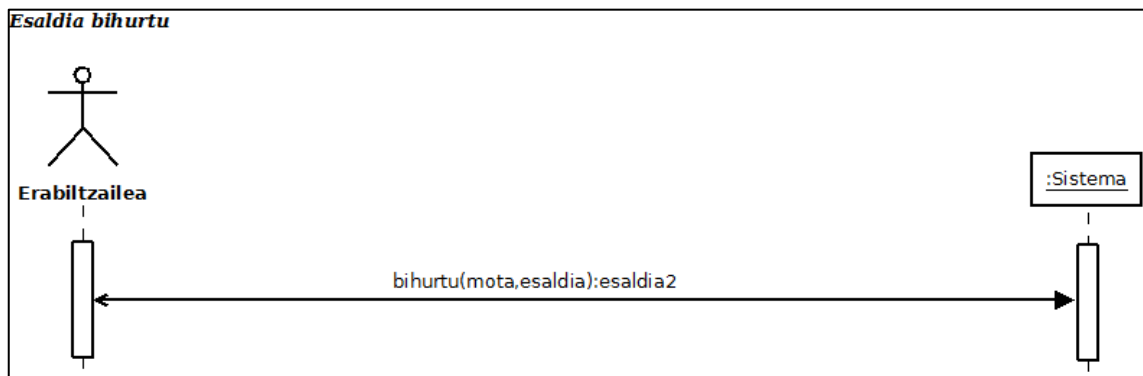
### 3.1.2 Aldaketa mota aldatu

*Ikus 2.4.2 sekuentzia-diagrama*

### 3.1.3 Aldaketa mota ezabatu

*Ikus 2.4.3 sekuentzia-diagrama*

## 3.2 Esaldia bihurtu



14.diagrama: Esaldia bihurtu sekuentzia-diagrama

Izena: bihurtu(mota,esaldia):esaldia2

Sarrera:

- mota = Aldaketa\_mota
- esaldia = String

Erantzukizunak: Erabiltzaileak aldaketa mota bat aukeratu eta esaldia bihurtzeko eskatzean, sistemak emaitza bezala bihurtutako esaldia itzultzen du.

Aurrebaldintzak: mota existitzen da

Postbaldintzak: Ø

Irteera

- esaldia2 = String

## 5.3 Diseinua

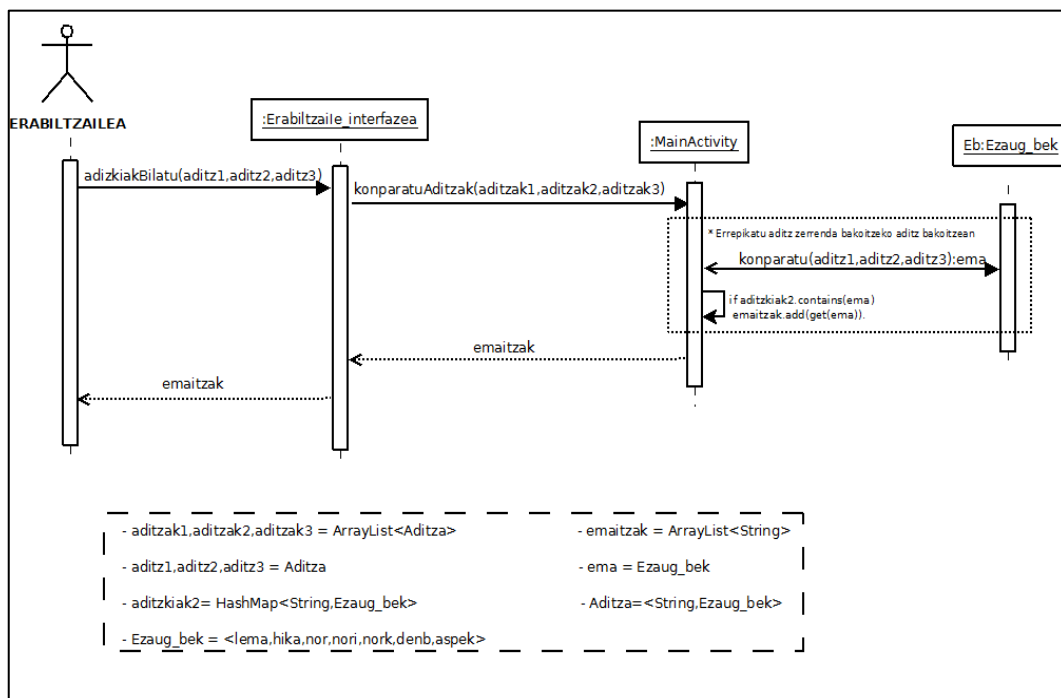
Diseinuaren barnean sistema sekuentzia-diagramak sortu dira

### 5.3.1 Sistema sekuentzia-diagramak

#### 1. Modu arrunta

##### 1.1 Adizkiak bilatu

- hitzaBihurtu(aditz1,aditz2,aditz3):emaitzak

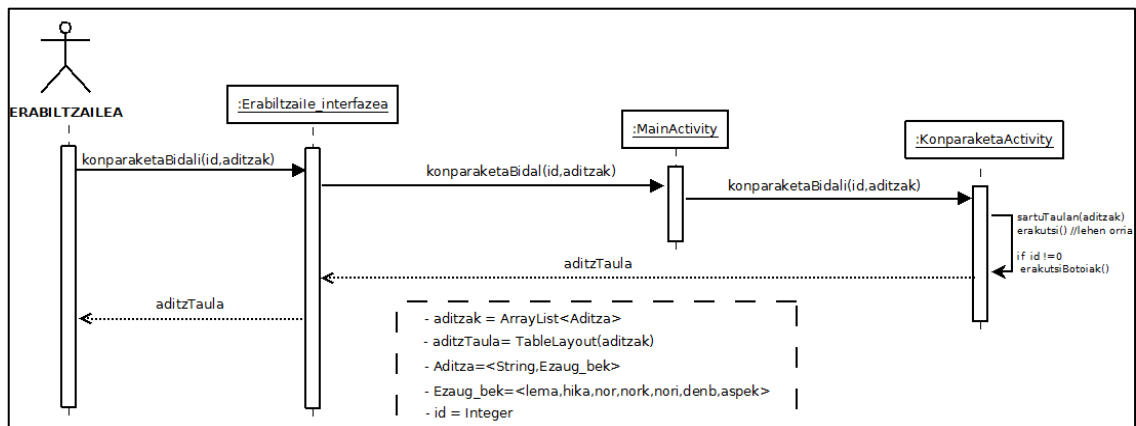


15.diagrama: hitzaBihurtu() sistema sekuentzia-diagrama

## 1.2 Adizkiak desanbiguatu

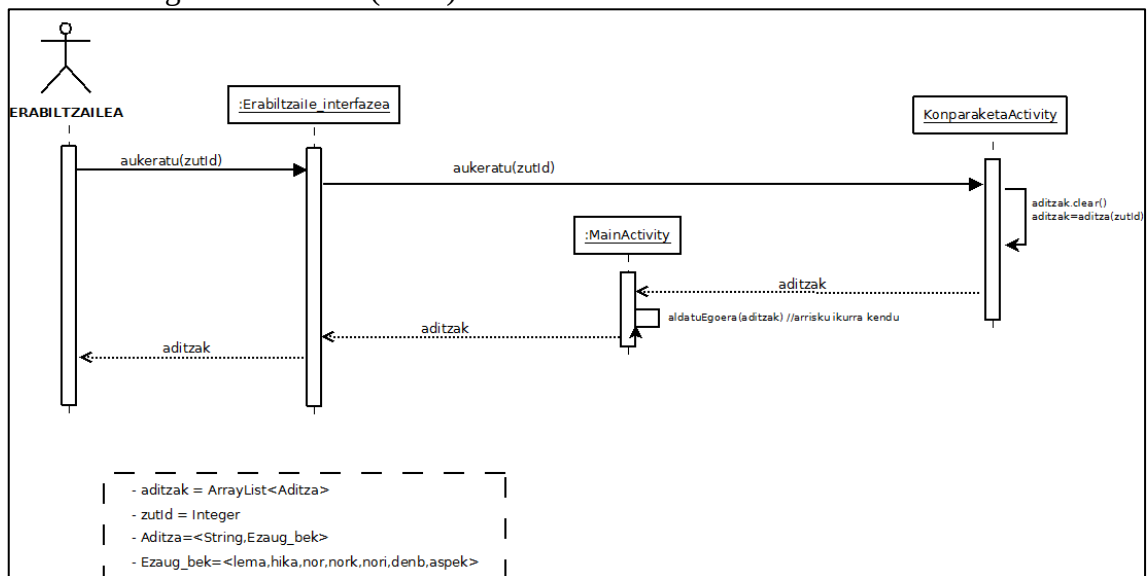
### 1.2.1 Adizkia aukeratu

- 1.eragiketa: konparaketaBidali(id,aditzak):aditzTaula



16.diagrama: konparaketaBidali() sistema sekuentzia-diagrama

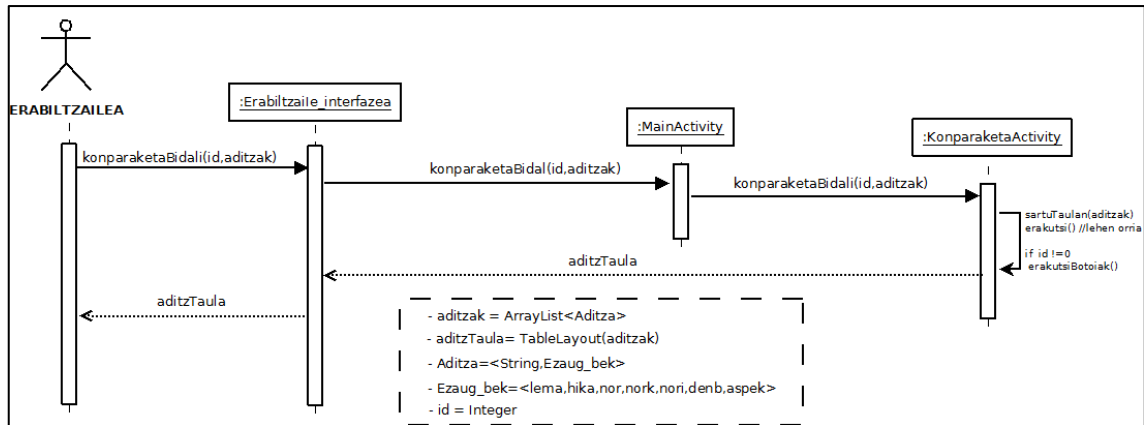
- 2.eragiketa: aukeratu(zutId):aditza



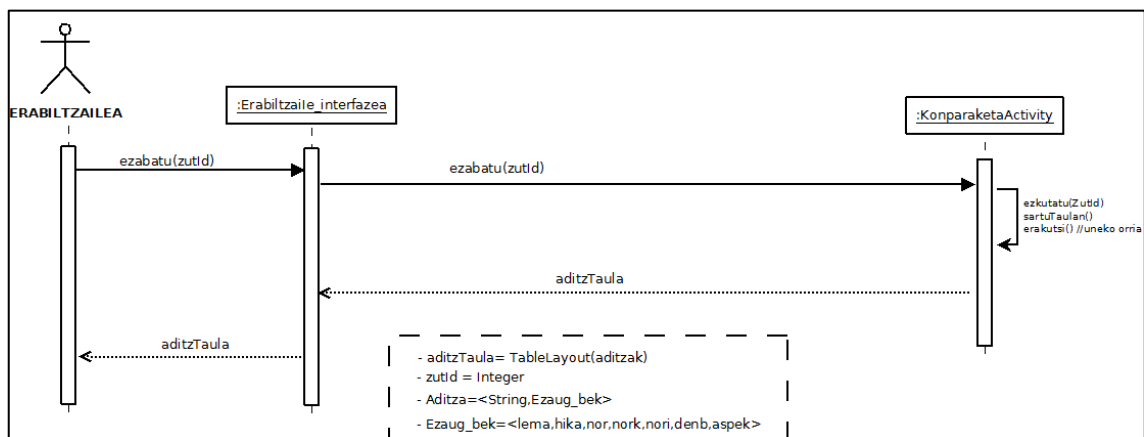
17.diagrama: aukeratu() sistema sekuentzia-diagrama

## 1.2.2 Adizkia ezabatu

- 1.eragiketa: konparaketaBidali(id,aditzak):aditzTaula

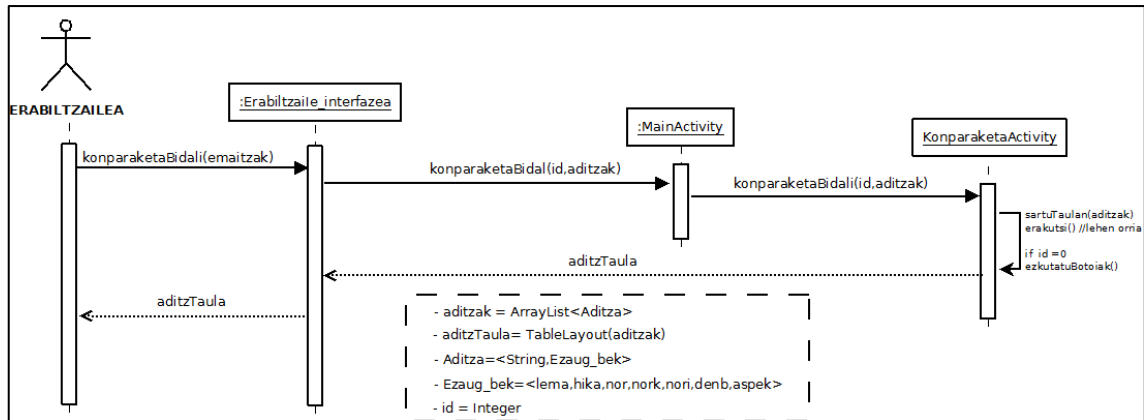


- 2.eragiketa: ezabatu(zutId):aditzTaula



### 1.3 Emaitzak konparatu

- konparatu(emaitzak):aditzTaula



20.diagrama: konparaketaBidali() sistema sekuentzia-diagrama

## 2. Aldaketa motak

### 2.2 Adizkiak bilatu

*Ikus 1.1 sistema sekuentzia-diagramak*

### 2.2 Adizkiak desanbiguatu

*Ikus 1.2 sistema sekuentzia-diagramak*

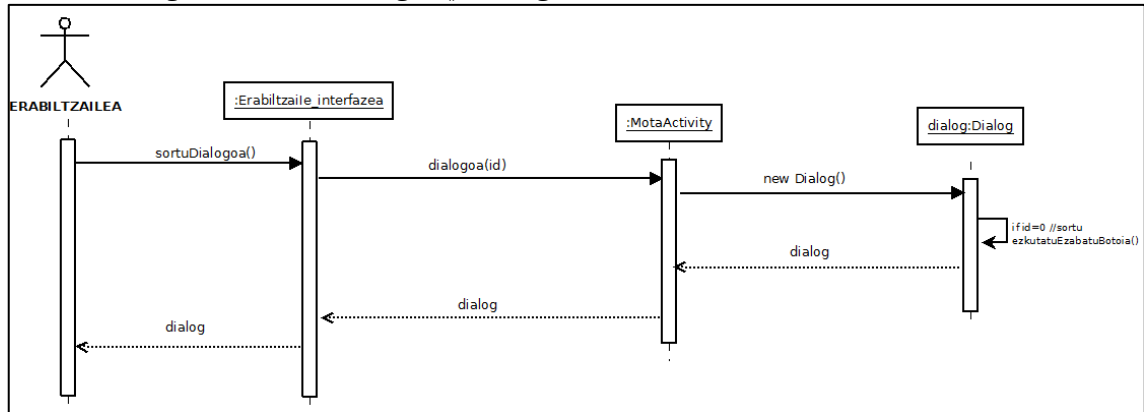
### 2.3 Emaitzak konparatu

*Ikus 1.3 sistema sekuentzia-diagramak*

## 2.4 Aldaketa motak kudeatu

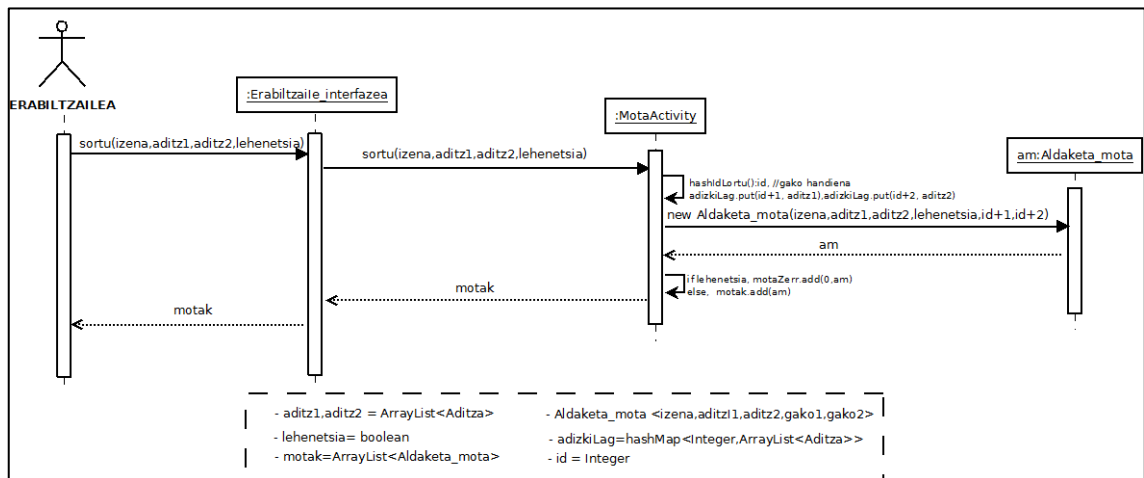
### 2.4.1 Aldaketa mota sortu

- 1.eragiketa: sortuDialogoa():dialog



21.diagrama: sortuDialogoa() sistema sekuentzia-diagrama

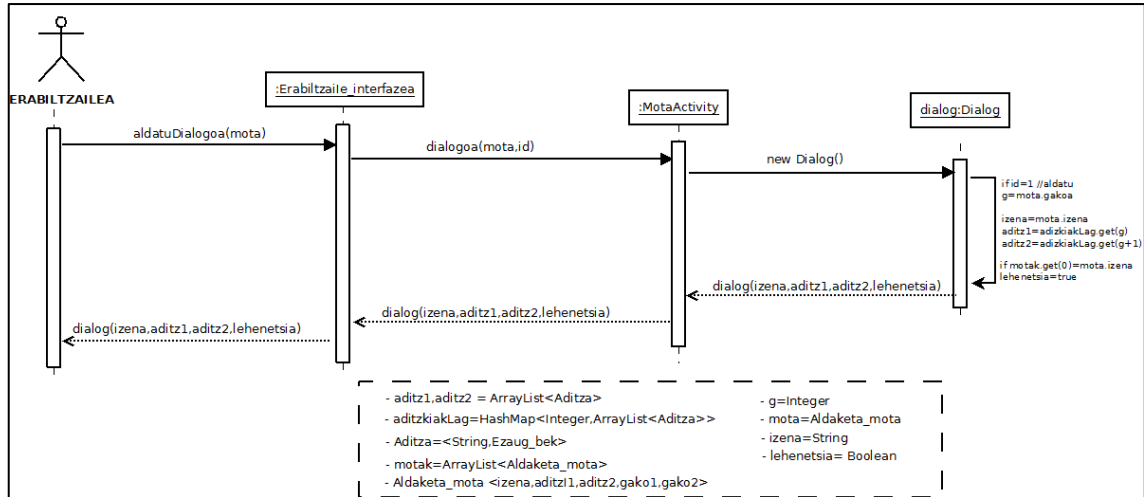
- 2.eragiketa: sortu(izena,aditz1,aditz2,lehenetsia):motak



22.diagrama: sortu() sistema sekuentzia-diagrama

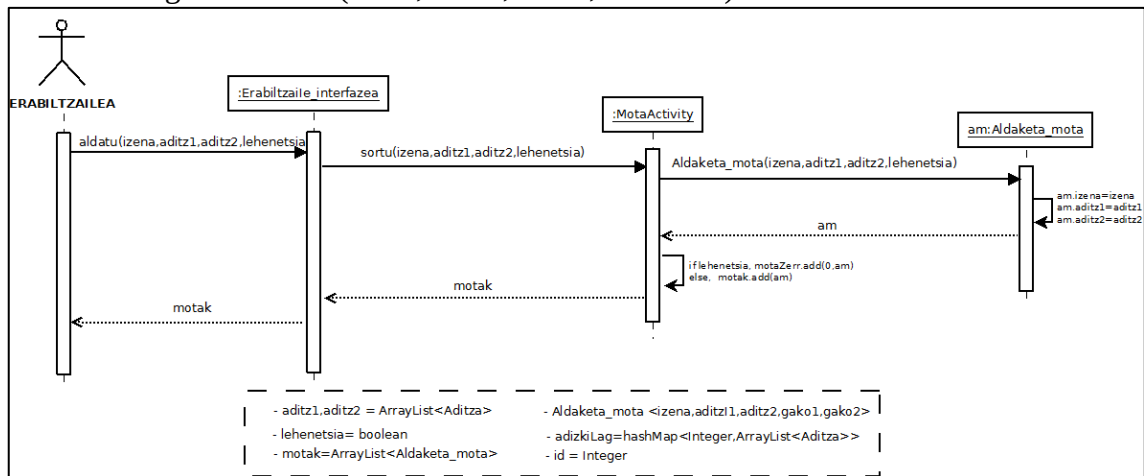
## 2.4.2 Aldaketa mota aldatu

- 1.eragiketa: aldatuDialogoa(mota):dialog(izena,aditz1,aditz2,lehenetsia)



23.diagrama: aldatuDialogoa() sistema sekuentzia-diagrama

- 2.eragiketa: aldatu(izena,aditz1,aditz2,lehenetsia):motak

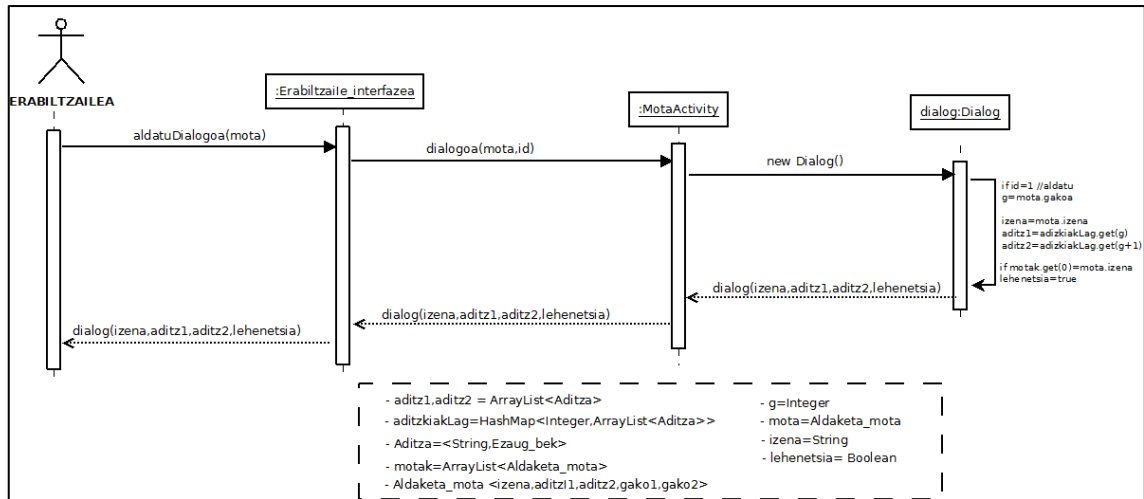


24.diagrama: aldatu() sistema sekuentzia-diagrama



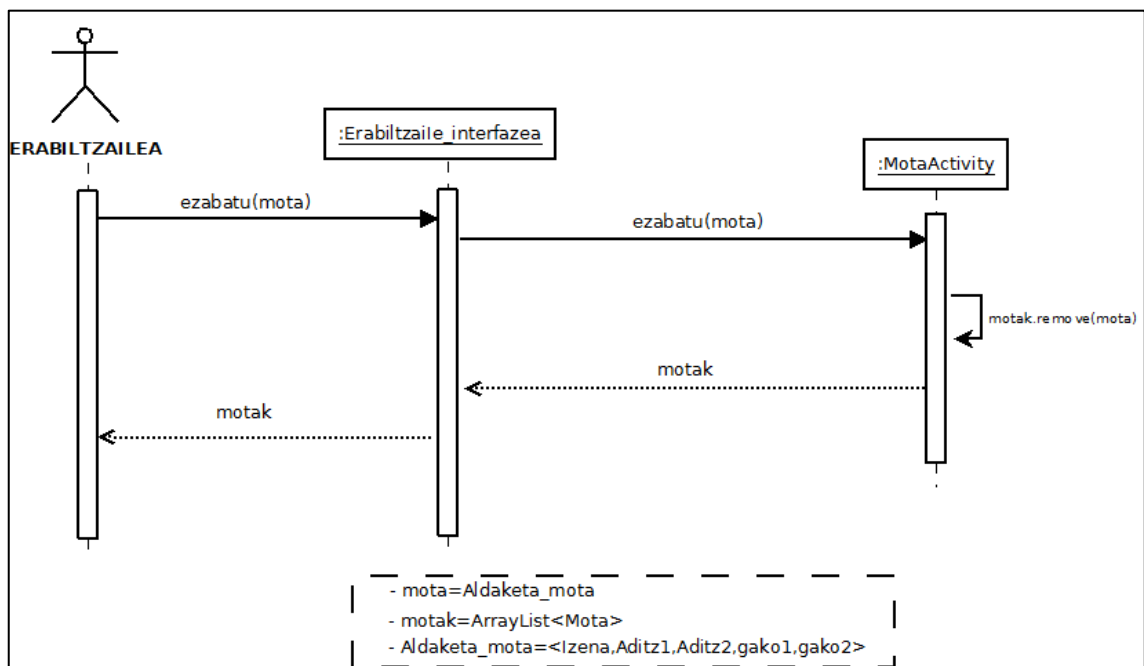
### 2.4.3 Aldaketa mota ezabatu

- 1.eragiketa: aldatuDialogoa(mota):dialog(izena,aditz1,aditz2,lehenetsia)



25.diagrama: aldatuDialogoa() sistema sekuentzia-diagrama

- 2.eragiketa: ezabatu(mota):motak



26.diagrama: ezabatu()sistema sekuentzia-diagrama

### 3. Esaldia

#### 3.1 Aldaketa motak kudeatu

*Ikus 2.4 sistema sekuentzia-diagrama*

##### 3.1.1 Aldaketa mota sortu

*Ikus 2.4.1 sistema sekuentzia-diagrama*

##### 3.1.2 Aldaketa mota aldatu

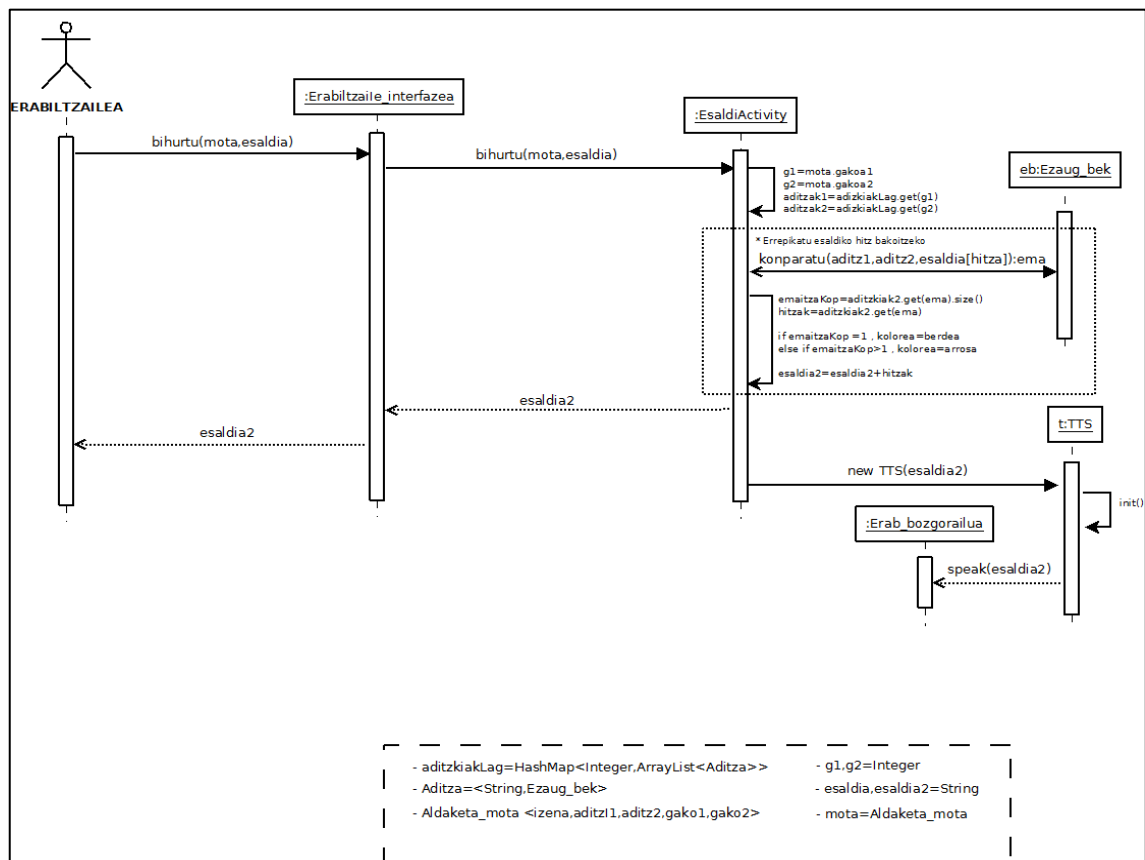
*Ikus 2.4.2 sistema sekuentzia-diagrama*

##### 3.1.3 Aldaketa mota ezabatu

*Ikus 2.4.3 sistema sekuentzia-diagrama*

#### 3.3 Esaldia bihurtu

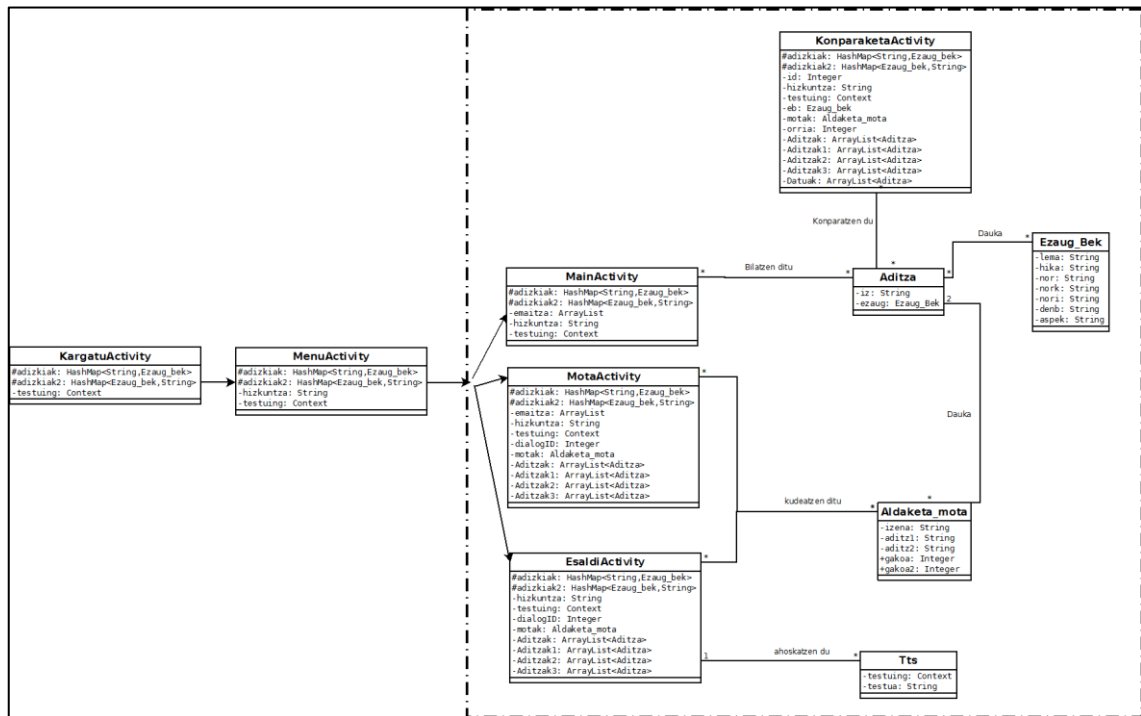
- Bihurtu(mota,esaldia):esaldia2



27.diagrama: bihurtu() sistema sekuentzia-diagrama

### 5.3.2 Klase diagrama

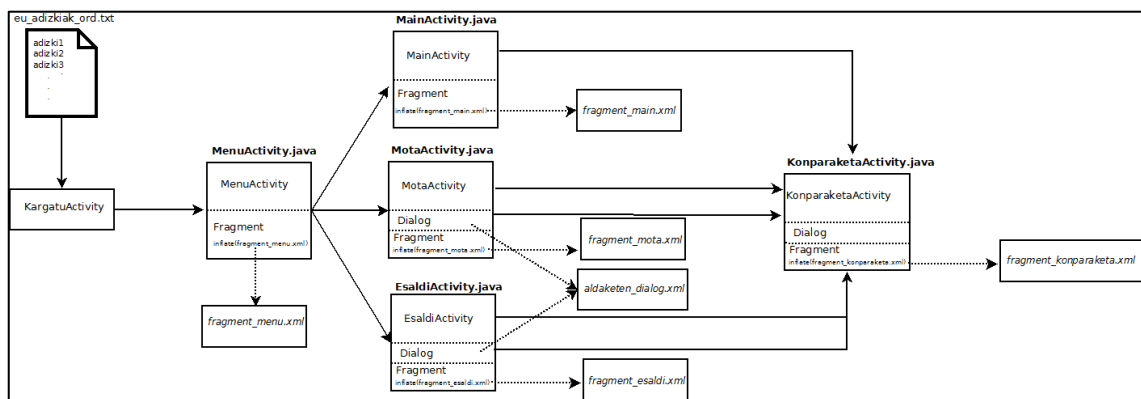
Aplikazioan erabili diren klase eta objektuak, Javarenak izan dira eta eskema honetan ikus daiteke klaseen egitura orokorra eta erlazioak:



28.diagrama: Klase diagrama

### 5.4 Sistemaren arkitektura

Aplikazioko sistemaren arkitektura orokorra ondoko diagrama honetan ikus daiteke.



29.diagrama: Sistemaren arkitektura orokorra

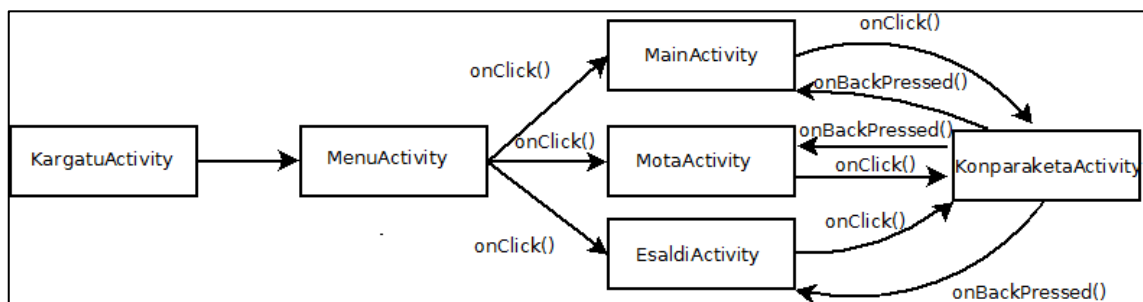
Aplikazioaren abiarazlea (launcher) KargatuActivity da. Bertan, datu-basea kargatzen da teknologiaren azterketan aipatutako ‘assets’ katalogoan dagoen ‘eu\_adizkiak\_ord.txt’ fitxategitik.

Kargatzen amaitutakoan, MenuActivity aktibitateara jotzen du sistemak eta honek (eta aktibitate guztiek) Fragment klase bat du barnean nondik aktibitateari dagokion .xml interfazea deitzen den (layout katalogoan dagoena) eta horrela diseinatutako interfazea abiarazten da, kasu honetan ‘fragment\_menu.xml’ delarik.

Prozesu hau, aktibitate bat martxan jartzen den bakoitzean gauzatzen da eta aktibitatean dagoela, beste aktibitateetara jo dezakete edota itzuli aurrerago aipatuko den moduan.

Hala ere, badira 2 aktibitate ezberdinak besteekin konparatuta, ‘MotaActivity’ eta ‘EsaldiActivity’ hain zuzen. Hauek izan ere , barnean fragmentu bat izateaz gain, Dialogo bat ere badute aldaketak kudeatzeko erabiltzen dena eta biek dialogo berdina erabiltzen dute (aldaketen\_dialog.xml) interfazeari dagokiona hain zuzen eta beraz, dialogoa hasieratzean, interfaze hori azaleratzen du sistemak aktibitate beraren barnean.

Aurretik aipatu bezala, aktibitate batetik bestera mugitu daiteke aplikazioa eskema honetan ageri den moduan.



30.diagrama: Aktibitateen fluxu-diagrama

Aktibitate aldaketa hauek botoien bidez egiten dira beti, aldaketa eragiten duen botoiaren gainean ekintza nabari eta onClick() entzungailu baten bitartez. Gailuko atzerako botoia sakatzen bada berriz, sistemak ezer adierazi ezean aurreko pantaila berdinerara eramaten du baina aplikazio honek konparaketetatik emaitza batzuk itzuli behar ditunez, atzeko botoia kontrolatu beharra dago aktibitate horretan, Androidek berak lehenetsita ekartzen duen onBackPressed() metodoa erabiliz.

Aipatu beharra dago baita ere, aktibitateen arteko aldaketa hauetan elementuak pasa nahi direnean (esaterako uneko hizkuntza beti pasatzen da) parametroen erabilera egin behar dela(Bundle klasea) eta parametroak sartzeko eta lortzeko getExtra() eta putExtra() izeneko metodoak erabiltzen dira.

## 5.5 Implementazioa

Erabili den teknologia aztertu den moduan, implementazioa Eclipse eta Android Ingurunea bateratuz gauzatu da, lehenengoa Java kodea idazteko erabiliz eta bigarrena berriz Android diseinuak egiteko.

Garatzaileak Java objektu bidezko programazio lengoaia urte askotan eman duenez eta gehien menperatzen duen programazio lengoaia izaki, alde horretatik ez da inolako arazorik suertatu, oinarritzko Javaren ia berdina baita Androiden erabiltzen dena baina Androidek bere berezitasunak ditu eta hor arazo asko izan dira, garatzaileak proiektua hasi aurretik Androiden inguruan ezer gutxi baitzekien.

Adizkiak erabiltzeko ideia sinplea da. Programa martxan jartzen denean, kargatzeko sortutako aktibitate batek datu-basea hartzen du eta bi hash taula estatiko sortzen dituenean menuaren aktibitatera joaten da. Honek eta beste aktibitate guztiek datu-baseko adizki guztiak hasierako hash taula estatikoetatik eskuratzen dute handik aurrera bilaketen ordena  $O(1)$ , orden konstantea izango delarik.

Izan diren arazoak gehientsuenak konpontzea ez da hain zaila izan gaur egun sarean dagoen informazio ugariari esker eta Androidek berak duen garatzaile gidari esker:

### 5.5.1 Implementazioan arazoak

Esan berri den moduan, ia arazo guztiak Android eta Java uztartzeko momentuan agertu dira eta beste batzuk Androiden diseinu garaian eta Androidek erabiltzen duen manifestuan aldaketak egitean. Honakoak izan dira arazo nagusienak.

- Fitxategi lokalen erabilera: Androiden fitxategiak tratatzeko modua ez da oinarritzko Javaren berdin berdina. Hasieran proiektuaren erroan datu-basea den fitxategia sartu eta irakurtzean ikusten zen fitxategia ez dela aurkitzen. Ondoren ikusi zen Androidek fitxategi lokalak kudeatzeko katalogo propio bat daukala ‘assets’ izenekoa eta `InputStreamReader` bat aurkitzean zuzenean katalogo horretara jotzen duela.
- Barruko memoriaren erabilera: Aipatu berri den soluzio honek ordea, arazo larri bat dauka. ‘assets’ izeneko katalogo horretan irakurketak soilik egin daitezke baina idazketak ez daude onartuak. Aurreko arazoan datu-base bat irakurri nahi zen soilik eta ez zegoen eragozpenik baina garapenean iritsi zen momentu bat erabiltzaileak sartzen dituen datu pertsonalak (kasu honetan aldaketa motak eta beraien informazioa) mugikorrean gorde behar zituen eta katalogo horretan ezin dira idazketarik egin.

Hori dela eta, Androiden gidara joan (Erref5) eta bertan aipatzen dira barruko memoria (Internal Storage) kudeatzeko aukerak eta bertan ikus daiteke, testuingurua erabilia `FileInputStream.(OpenFileInput(“Fitxategia”))` metodoa

eskuragarri dago eta honek testuinguruan bertan fitxategi bat gordetzen du barruko memoria dinamikoan bertan eta hau horrela behar denean bertatik irakurri eta eguneratu behar denean bertan idaztea nahikoa da `openFileInput("Fitxategia")` eginez.

Hau ikusirik, aldaketak motak nola gorde pentsatu eta lerro bakoitzeko aldaketa motaren izena, lehen aditza eta bigarren aditza gordetzen hasi zen '\*' banatzaile baten bitartez eta gero `split()` bidez zatitu irakurtzerako garaian. Soluzio hau izan da garapenean zehar egin den akatsik handiena izan ere soluzio zikina izateaz gain (Aldaketa motaren izenean \* jarri nahiko balitz arazoak egongo lirатеke esaterako) irakurketak eta idazketak egiteko nahaspila handia sortzen da eta horrelakoetan hutsak aurkitzeko zailtasuna handiagoa da.

Hau gutxi ez balitz, idazketa zikina egiten zen eta tarteka idaztean '◆' karaktere berezia idazten zela ikusten zen. Arazo hau konpontzea nahikoa nekeza izan zen, sarean zehar ez baitzen honen inguruko soluziorik aurkitu eta arazoa agian garapena Windows sistema eragilean egiteagatik zela zirudien. Gerora ikusi zen arazo hau `ObjectOutputStream` objektua erabiltzean gertatzen zela baina proba batzuk ondo egin ondoren berriro agertu zen arazoa.

Arazo hau konpondu gabe, handiago bat azaldu zen. Aldaketa mota bakoitza gordetzen zen baina aldaketa motaren izena gordetzea bakarrik ez da nahikoa, izan ere, adizki bat desanbiguatu ondoren informazio ezberdina izango du adizkia desanbiguatu gabe izan behar duen informazioarekin. Beste era batera esanda, adizki zerrendak gorde behar dira nonbait memorian adizki horien informazioarekin eta ez da modu egokia testu fitxategietan banatzaileen bitartez guztia gordetzen joateko. Hau ikusirik, soluzio bila joan eta zuzenean objektu bat fitxategian idatzi eta irakurtzeko aukera zegoela ikusi zen, askoz modu sinpleago batean `ObjectInputStream.readObject(objektua)` erabiliz irakurri eta `ObjectOutputStream.writeObject(objektua)` bidez fitxategian objektua idatz daiteke eta hain modu sinplean aurreko arazoak konponduta zeuden. Hasieran objektua irakurri, objektuari nahi diren aldaketak egin eta gero memoria eguneratzeko berriro irakurtzea nahikoa da banatzaileak edota karaktere arraroekin kezkatu beharrik gabe. Hau dela eta aldaketa moten informazioan adizki bakoitzeko gako bat sartu da eta beste fitxategi batean Hash taula bat non gako bakoitzean adizkiaren informazioa agertzen den, desanbiguatuta egon edo ez bakoitzak berea izango duena.

- Aplikazioaren tamaina: Aplikazioaren helburu tekniko nagusiak atzipen azkarrekoak eta tamaina txikikoak izateak dira eta horren bila jo da. Lehenengo baldintza betetzeko zuzenean Hash-ak erabiltzea pentsatu da, bilaketa bat egin orduko 17.000 sarreratik gora dituen fitxategi bat lerroka analizatzea ez baita bideragarria.

Honek ordea arazo bat du, izan ere datu-baseko informazioa nonbait gorde behar da bilaketak egin ahal izateko eta hor aukera batzuk azaltzen dira:

- Fitxategia erabili: Kasu honetan 700KBeko fitxategia kargatzen da hasieran baina horretarako denbora behar du. 1GBeko RAM-a daukan mugikorretan, 1,5 segunduko atzerapena dakar hau egiteak, mugikor bakoitzaren RAM memoriaren araberakoa delarik atzerapen hau.
- Memorian sartu: Lehenengo aldiz programa abiaraztean (edo eskuz datuak ezabatzean) kargatzen da eta Hash taula barruko memoria gorde fitxategi batean. Hash taulak 17000 sarreratik gora izango ditu eta barruko memoria 28MB izatera pasatzen da. Abantaila bezala aurreko aukeran galtzen den denbora lehen aldian soilik egiten da.

Hau ikusita lehenengo aukera erabaki da, batez ere memoria sartzea ez delako bideragarria ikusten. Gaur egun gainera erabiltzaileak oso kexu dira memoria faltagatik eta denboraren latentzia arazo txikiagoa dela iruditzen da. Beste soluzio bat agian datu-basean gutxi erabiltzen diren aditz ugari daudenez, horiek eskuz kentzea litzateke baina hala eta guztiz ere fitxategiak milaka sarrera izan eta memoriaren tamaina handiegia litzateke bideragarria izateko. Demagun erdia kentzen dela, hala ere 14 MB-eko datuak izango ziren eta hori gehiegi kontsideratzen da.

- Parametroen erabilera: Android aztertzerakoan aipatu bezala, aplikazioko pantaila bakoitzak bere aktibitate ('activity') propioa dauka eta aktibitate bakoitzak bere diseinu('layout') propioa. Aktibitate batean elementu bat dinamikoki aldatu nahi bada nahikoa da R.id.elementua eskuratzea eta Objektu gisa erabili bere metodoak aldaketak egiteko.

Aplikazio honetan ordea aktibitate batek beste aktibitate baten elementua eskuratu beharra dauka, izan ere demagun erabiltzaileak adizki bat desanbiguatu nahi duela eta konparaketa taulara joaten dela. Bertan zutabe bat ezabatu edo aukeratu ondoren aldaketa horiek aplikatu beharra daude interfazearen (anbiguotasun ikurra aldatzea) eta gainera aldaketa horiek (adizkien informazio berria) ere pasatu behar dira.

Bigarrenengo hau egiteko elementu estatikoak erabil daitezke baina horien erabilera arriskutsua da Androiden eta ez da komeni erabiltzea gehiegi. Hori dela eta, androiden aktibitate batetik bestera parametroak bidali eta jaso daitezke Bundle izeneko objektuen bidez eta putExtra() edo getExtra() bidez datu egiturak pasa edo lortu hurrenez hurren. Proiektu honetan aukera honi heldu zaio parametroak pasatzerako garaian baina zehazki aplikazio honetan nahaspilak sortzeko erabakia izan zen.

Izan ere, pantailan aurrera eta atzera ibiltzerako garaian, beti KonparaketaActivity izeneko aktibitate batera pasatzen da baina arazoa da aktibitate horretara joateko modu ugari daudela:

- MainActivity: 3 testu kutxetako adizkiak desanbiguatzeko edota emaitzak konparatzeko.
- MotaActivity: 3.adizkia idazten den testu kutxatik desanbiguatzeko edota aldaketa motak kudeatzean sortzen den dialogotik desanbiguatzera joateko.
- EsaldiActivity: Aldaketa motak kudeatzean sortzen den dialogotik desanbiguatzera joateko

Hau horrela, nola jakingo du konparaketaren aktibitateak nondik datorren eta zer egin behar duen? Horretarako identifikadore bat ere pasatzen da datuez gain eta gero konparaketako aktibitateak identifikadorearen arabera parametro batzuk edo besteak hartuko ditu egin beharrekoa egiteko.

```
Bundle parametroak = getIntent().getExtras();
id = parametroak.getInt("id");
nondik = parametroak.getString("nondik");
hizk=parametroak.getString("hizkuntza");
if(nondik.equals("main")) {
 aditzak=(ArrayList<Aditza>) parametroak.getSerializable("aditzak");
 aditzak1=(ArrayList<Aditza>) parametroak.getSerializable("aditzak1");
 aditzak2=(ArrayList<Aditza>) parametroak.getSerializable("aditzak2");
 aditzak3=(ArrayList<Aditza>) parametroak.getSerializable("aditzak3");

 g1=parametroak.getString("gelaxka1");
 g2=parametroak.getString("gelaxka2");
 g3=parametroak.getString("gelaxka3");

}
else if(nondik.equals("mota")){
 aditzak=(ArrayList<Aditza>) parametroak.getSerializable("aditzak");
 aditzak3=(ArrayList<Aditza>) parametroak.getSerializable("aditzak3");
 emaitza=(ArrayList<String>) parametroak.getSerializable("emaitza");
 g3=parametroak.getString("gelaxka3");

}
else if(nondik.equals("motaDialog")){
 aditzak1=(ArrayList<Aditza>) parametroak.getSerializable("aditzak1");
 aditzak2=(ArrayList<Aditza>) parametroak.getSerializable("aditzak2");
 aditzak3=(ArrayList<Aditza>) parametroak.getSerializable("aditzak3");

 g1=parametroak.getString("gelaxka1");
 g2=parametroak.getString("gelaxka2");
 g3=parametroak.getString("gelaxka3");
 iz=parametroak.getString("izena");
 dialogID=parametroak.getInt("dialogID");

}
else if(nondik.equals("esaldi")){
 aditzak1=(ArrayList<Aditza>) parametroak.getSerializable("aditzak1");
```

14.irudia: KonparaketaActivity-ko parametroen erabilera

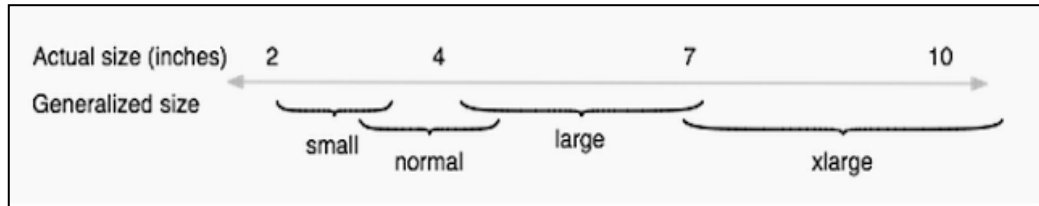
Beste aukera bat, konparaketara joateko aukera bakoitzari aktibitate eta diseinu kopia bat jarrita bakoitza berera joatea da baina kode errepikatu asko egongo litzatekeenez beste aukerara jo da, nahiz eta garatzerako orduan kontu gehiagorekin ibili behar den nahaspilarik ez sortzeko.

- Pantailen bateragarritasuna: Gaur egun pantaila tamaina ugari daude eta gero eta tamaina gehiagotakoak sortzen ari dira. Honek diseinuan arazo larria dakar, pantaila mota guztietarako bateragarriak egin behar baitira Android aplikazioak eta horrek diseinuan hori kontrolatzea dakar.

Hasieran garapena garatzailearen mugikorrean egin denez eta hango pantailara moldatuta soilik egin da. Denbora aurrera joan ahala ordea pantaila txikiagoetan



botoi batzuk ez zirela ikusten nabaritu zen. Horren arrazoia ordu arte pixelak erabiltzea izan da neurriak hartzeko, izan ere gaur egun Androidek bereizmenaren eta dentsitatearen arabera egiten ditu sailkapenak eta gomendatzen duen neurria dp edo dip (density-independent pixels) izenekoa da horrela mugikor bakoitzaren dentsitatearekin kezkatu beharrik izan gabe diseinatzea dagoelarik.



15.irudia: Androideko sailkapena pantaila tamainaren arabera

Hau honela, androidek 5 tamaina motetan egiten du gaur egun sailkapena. Pantaila txikiei ldpi deitzen zaie, hau da dentsitate baxukoak. Gero mdpi dator dentsitate normaltzat hartzen dena eta ondoren handiak datoz. Handiak 4" eta 7" artekoak dira eta hdpi deitzen zaie eta handiagoak xhdpi . Azkenik, nahiz eta aurreko irudian ez agertu , oraindik eta pantaila handiagoak egon daitezke eta horiek xxhdpi multzoan sartzen dira, 10" ko pantaila baino handiagoa dutenetan.

Dentsitate ezberdinek ordea irudiekin arazoak ekartzen zituzten izan ere irudi bera dentsitate ezberdineko pantailetan ez da berdina ikusten eta irudiak pixel gehiegirekin ikusteko arriskua dago. Hori dela eta Androidek berak espezifiki aurreko sailkapeneko tamaina mota bakoitzeko karpeta bana sortzen du eta tamaina bakoitzaren arabera irudiak jartzea gomendatzen da irudien bateragarritasuna lortu ahal izateko.

## 5.6 Probak

### 5.6.1 Garatzailearen probak

Hauek izan dira inplementatu ondoren garatzaileak eginiko proba nagusienak. Esperotako emaitza eman diote garatzaileari proba fasean.

- Existitzen ez den adizkia sartu

**Sarrera:** 1.adizkia='daukal'

**Egoera:** Modu arrunta

**Testuinguruaren informazioa:** Adizkia bilatzean

**Espero den emaitza:** Ezabatzeke ikurra testu kutxaren ondoan

**Lortutako emaitza:** Ezabatzeke ikurra testu kutxaren ondoan

- Emaitza bakarra denean konparatzen saiatu

**Sarrera:** emaitza= zerrenda <zeukaat,zaukaat>

**Egoera:** Modu arrunta

**Testuinguruaren informazioa:** Adizkiaren bilaketa eginda

**Espero den emaitza:** Ez ireki leiho berririk

**Lortutako emaitza:** Ez da ireki leiho berririk

- Desanbiguatu ondoren itzuli

**Sarrera:** adizkia = zoazte

**Egoera:** Modu arrunta

**Testuinguruaren informazioa:** Adizkia desanbiguatu da eta konparaketa taulatik itzuli da

**Espero den emaitza:** Arrisku ikurra desagertzea

**Lortutako emaitza:** Arrisku ikurra desagertu da.

- Desanbiguatzerantz joan , zutabe bat ezabatu guztiz desanbiguatu gabe eta itzuli

**Sarrera:** adizkia = zoazte

**Egoera:** Modu arrunta

**Testuinguruaren informazioa:** Adizkiaren desanbiguatzea egiteko eskatu baina oraindik adizki anbigua da.

**Espero den emaitza:** Arrisku ikurra mantentzea

**Lortutako emaitza:** Arrisku ikurra mantentzen da.

- Aldaketa mota sortzen saiatu existitzen den izenarekin

**Sarrera:** aldaketa mota izena = "hitanoa (toka)"

**Egoera:** Aldaketa motak

**Testuinguruaren informazioa:** Aldaketa mota sortzeko eskatu eta dialogo berri bat ireki denean, mota sortzen du.

**Espero den emaitza:** Errore mezua

**Lortutako emaitza:** Aldaketa mota dagoeneko existitzen dela adierazten duen mezua agertzen da.

- Hitanoaren aldaketa moda aldatzen saiatu

**Sarrera:** aldaketa = "hitanoa (toka)"

**Egoera:** Aldaketa motak

**Testuinguruaren informazioa:** Aldaketa mota aldatzeko eskatu eta dialogo berri bat ireki da.

**Espero den emaitza:** Aldaketaren informazioa bistaratzea aldatzen eta ezabatzen utzi gabe baina lehenesteko aukera emanez.

**Lortutako emaitza:** Aldaketaren informazioa bistaratu da aldatzen eta ezabatzen ez da utzi baina lehenetsi daiteke.

- Aldaketa mota lehenetsi

**Sarrera:** aldaketa = "<"pertsonalizatua",dauka(anbiguo), zizkian, lehenetsia = true>

**Egoera:** Aldaketa motak

**Testuinguruaren informazioa:** Aldaketa mota aldatzeko eskatu eta lehenesteko esaten da

**Espero den emaitza:** Aldatu beharrekoa, zerrendan lehenengoa bihurtzea eta lehenetsitako ikurra agertzea

**Lortutako emaitza:** Aldatu berri dena zerrendako lehen posizioan txertatu eta lehenetsitako ikurra agertu zaio.

- Aldaketa mota aldatu desanbiguatuz

**Sarrera:** aldaketa = "pertsonalizatua"

**Egoera:** Aldaketa motak

**Testuinguruaren informazioa:** Aldaketa motaren 'dauka' adizkia desanbiguatzen da 'banadi' modu-denbora soilik hartuz eta aldaketa mota gorde.

**Espero den emaitza:** Emaitarik ez 'diat' adizkia pertsonalizatu bihurtuta

**Lortutako emaitza:** Emaitarik ez da azaldu 'diat' adizkia pertsonalizatu bihurtuta.

- Aldaketa mota ezabatu

**Sarrera:** aldaketa = "pertsonalizatua"

**Egoera:** Aldaketa motak

**Testuinguruaren informazioa:** Aldaketa mota ezabatzeko eskatzen da

**Espero den emaitza:** Aldaketa ezabatu eta zerrendatik kendu.

**Lortutako emaitza:** Aldaketa ezabatzen da eta zerrendatik kendu

- Esaldia entzun TTS bidez

**Sarrera:** bihurtzeko esaldia = “Ez daukat etxerik lo egiteko”, aldaketa mota=”hitanoa(toka)”

**Egoera:** Esaldiak

**Testuinguruaren informazioa:** Esaldia bihurtzeko eskatzen da eta bozgorailuak aktibatuta daude.

**Espero den emaitza:** Dauzkat anbigua denez, lehen emaitza bakarrik entzun.

**Lortutako emaitza:** ‘Zaukaat/zeukaat’ idazten dira baina zaukaat soilik entzuten da.

- Desanbiguatzean zutabe bakarra geratu

**Sarrera:** Ezabatu ‘dauka’ adizkiaren ‘banadi’ forma

**Egoera:** Konparaketa taula

**Testuinguruaren informazioa:** ‘Dauka’ adizki anbiguo desanbiguatzen ari da eta naiz modu-denbora duen forma ezabatu da.

**Espero den emaitza:** Zutabe bakarra dagoenez ezabatu eta aukeratzeko botoiak kendu.

**Lortutako emaitza:** Ezabatu eta aukeratzeko botoiak kendu dira.

### 5.6.2 Proba errealak

Proba hauek proiektuaren garapenean zerikusirik ez dutenek egindakoak dira. Jakinarazi diren proba hauek guztiak espero gabeko emaitza eman diote erabiltzaileari.

- Testu-kutxan idaztean kurtsorea ikustea

**Sarrera:** Testu-kutxa

**Egoera:** Modu arrunta

**Testuinguruaren informazioa:** 1.Testu-kutxan idazteko prest

**Espero den emaitza:** Kurtsorea ikustea

**Lortutako emaitza:** Testu kutxako ertzak ez du ikusten uzten (Konponduta)

- Esaldietan lehenetsia gorde

**Sarrera:** -

**Egoera:** Esaldia

**Testuinguruaren informazioa:** Aldaketa mota lehenetsi bat gorde da esaldi moduan eta programa berriz martxan jarri eta esaldira sartu da berriz.

**Espero den emaitza:** Aukeratutako aldaketa mota zerrendan lehenengo ikustea

**Lortutako emaitza:** Aldatutakoa zerrendako azkena izatera pasa da (Konponduta)

- Interfaze horizontalean puntu beltzak

**Sarrera:** desanbiguatzeko adizkia = ”da”

**Egoera:** Konparaketa taula

**Testuinguruaren informazioa:** Adizkiaren informazioa erakusten da taulan eta zutabea ezabatzeko eskatzen da modu horizontalean.

**Espero den emaitza:** Zutabea ezabatu eta ezabatutako urdin izatera pasa.

**Lortutako emaitza:** Ezabatzen den zutabeetako botoiak atzeko plano beltza izatera pasa dira.

- Teklatuak testu-kutxa estali

**Sarrera:** -

**Egoera:** Modu arrunta

**Testuinguruaren informazioa:** Testu-kutxa pantailaren erdi/behe aldean dagoenean erabiltzailea idazten hasi

**Espero den emaitza:** Idazten ari dena ikustea

**Lortutako emaitza:** Teklatuak testu-kutxa estaltzen du eta ez da ikusten idatzitakoa

- Hitanoan aditz ohikoen falta

**Sarrera:** Adizkia = naiz, gara, zara, etab...

**Egoera:** Aldaketa mota

**Testuinguruaren informazioa:** Tokaz edo nokaz bihurtzean oso ohiko diren aditzen emaitzak falta

**Espero den emaitza:** nauk/naun, gaituk/gaitun, zara/haiz,

**Lortutako emaitza:** Emaitzik ez (Konponduta)

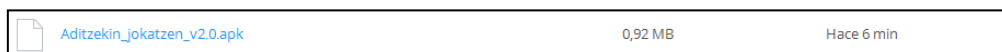
## 5.7 Ezarpen plana

Garatu berri den aplikazio hau oraindik ez dago oso osorik eta ez da guztiz egonkorra, Beta fasean baitago eta erroreak izan ditzake.

Hasiera batean ez zegoen asmorik Google Play denda arrakastatsura eramateko eskatzen duten kostuagatik, baina ikusita antzeko aplikazioek (Aditzak.com, Aditz laguntzailea eta aurrekarien atalean aipaturiko besteak) mila deskargatik gora dituztela dagoeneko, aplikazioaren egonkortasuna egonkorra denean bertara igotzeko asmoa dago dohainik mila horiek edo gehiagori aplikazioa lagungarri izan diezaielakoan.

Bien bitartean, aplikazioa bi ostalaritan jarriko da eskuragarri, Dropbox eta DroidBin izenekotan hain zuzen eta Internet konexioa beharrezkoa izango da deskargatzeko.

- Dropbox:  
<http://tinyurl.com/pgz7hhe> estekara joan eta bertan agertzen den “aditzak\_jokatzen.apk” fitxategia deskargatu behar da.



*16.irudia: Dropboxeko apk fitxategia*

- DroidBin:
- <http://tinyurl.com/aditzakJokatzenv2>  
estekara joan eta automatikoki deskargatzen hasiko da apk fitxategia mugikorrean.

Aplikazioak erabiltzeko honako baldintzak betetzea beharrezkoa da:

- Android sistema eragilea duen gailua
- Android sistema eragilearen bertsio minimoa = 2.2 (Android Froyo)
- Android SDK/API bertsio minimoa = 8

Honetaz gain, aplikazioaren aukera denak aprobetxatzeko oso gomendagarria da mikrofonoa eta bozgorailua izatea gailuak.

## 6. Ondorioak

---

### 6.1 Proiektuaren balorazioa

Egin den proiektuaren helburua aplikazio sinple baina batez ere praktikoa eta nonahikoa egitea izan da. Horregatik aukeratu da proiektua Androiden egitea eta erabiltzaileak ahalik eta botoi gutxien sakatuta euskara ikasteko edota zalantzaren bat argitzeko aukera modu erabilgarri batean eskaintzen du erabilgarritasuna bermatuz. Erabilgarritasunaz gain, irisgarritasuna ere ematen dio erabiltzaileari 3 hizkuntzatako interfazeak eskainita edonoiz aldatzeko moduan.

Proiektuaren kudeaketari dagokionez proiektuaren helburu minimoak aise bete dira finkatuta zegoen epean eta nahiz eta hobekuntza posible guztiak egiteko denborarik ez izan, gehientsuenak egin dira, beti epearen barruan ibiliz. Memoria bera ere espero baino lehenago entregatu da nahiz eta aipatu beharra dagoen azken bi hilabetetako esfortzua oso handia izan dela.

Teknologia aukeraketari dagokionez ere balorazioa positiboa da. Android gaur egun oso erabili izanik informazio asko dago beraien inguruan eta erabili diren tresnak praktikoak dira bai diseinatzeko garaian bai datu-maila inplementatzerakoan ere. Eclipse ingurunearen ezagutzak ere asko lagundu du teknologia aukeraketa behar bezala egiten.

Laburbilduz, proiektuak 4 hilabete iraun du eta hasieran eskatu zena ongi bete du. Horretaz gain, irakaslearekin hitzartutako eta hitzartzen joandako tresna berriak txertatu dira inplementazio fasean zehar guztiak epe barruan eginez eta kudeaketa onarekin.

Ondorioz, proiektuaren balorazioa **positiboa** dela esan daiteke.

### 6.2 Balorazio pertsonala

Proiektu bila hasi nintzenean, garbi nuen irakasleak eskainita edo adostutako zerbait egin nahi nuela ingeniaritza teknikoko proiektua enpresan egin eta eskutatik alde egin zidalako hango esfortzuak eta gehiegizko lehentasunak ondorioak ekarri zizkidalarik. Hori dela eta, proiektu kontrolatuago bat nahi nuen eta horixe lortu dut. Gainera, euskal zale amorratua izanik euskararen inguruko proiektu bat aurkitzea desio nuen eta helburu hori ere lortu dut Kepa Sarasolari eskainitako proiektuari esker. Hau gutxi ez balitz, proiektu hau Androiden garatu da eta nahiz eta gaur egun hasiko diren ikasleek ziurrenik unibertsitatean eman, guk ez dugu aukerarik izan eta teknologia hau ikastea oso baliagarria izango zaidala uste dut etorkizunera begira.

Teknikoko proiektuan ez bezala gainera, proiektua zuzendariarekin hartu-eman erregular batean egin dut eta hori ondo etorri zait esfortzua kontrolatzeko garaian epe bat beti errespetatuz eta nahiz eta azkeneko hilabetetako lan-karga handia izan, proiektua garatzen gozatu egin dut.

Beraz proiektuaren balorazioa nire aldetik **oso positiboa** izan da, aurrekoan lortu ez nuena lortu baitut, eginkizunak kontrolatzea. Orain itxoin egin beharko da ikusteko herritarren edota euskal ikasleek erabilgarritasuna aurkitzen dioten egindakoari eta gerora begira norbaitek hobekuntza nabarmen gehiago eskatzen dituen. Nahiz eta lan gehiago izan, horixe desioko nuke.

### 6.3 Etorkizunerako hobekuntzak

Hobekuntzen atala ikusterakoan aldeztatik planifikatutako hobekuntzak aztertu dira, zein egin diren eta zein ez adieraziz. Zoritxarrez proiektuaren epea iritsi aurretik ezin izan dira planifikatutako hobekuntza guztiak egin baina proiektua amaitutakoan lanak aurrera jarraituko balu, hobekuntza hauek egiteko asmoa dago:

1. Esaldi osoen itzulpenak egiteko lematizazioa egitea: Horrela hobeto identifikatuko dira adizkiak direnak bakarrik aldatu ahal izateko, Eustagger API baten bidez esaldiak morfologikoki eta sintaktikoki analizatu eta desanbiguatuz. Honek, esaldien itzulpenen fidagarritasuna asko hobetuko luke.
2. Adizkiak eleanitzak izatea: Dagoeneko interfazea eleanitza da, baina aplikazioa euskarazko adizkientzat soilik dago garatuta. Ongi legoke gaztelaniazko zein ingelesezko aditz irregularrekin ere erabiltzeko aukera izatea. Horretarako ordea lehenik eta behin adizki zerrenda oso bat eskuratu beharko da eta bertarako datuak prestatu.
3. Herri-hizkeretara moldatuta: Aplikazioa oraindik eta erabilgarriagoa izateko, herri bakoitzak bere hizkera propioa duenez, hauen adizkien itzulpenak egiteko aukera emango luke. Horretarako, web bidez gehien erabiltzen diren adizkiak itzultzeko aukera emango litzaioke erabiltzaileari eta horrela herri-hizkeren araberako adizki-aldaketak ere egin litezke.
4. Adizkien transformazioa foma bidez bideratu: Adizkien informazioa eta transformazioa foma analizatzailea erabiliz egin daiteke eraginkortasuna hobetuko litzatekeelarik.



# 7. Bibliografia

---

## 7.1 Erreferentziak

(Erref1). <http://tinyurl.com/aditzak> helbidetik eskuratua

(Erref3). <http://tinyurl.com/mintzalagun> helbidetik eskuratua

(Erref2). <http://tinyurl.com/aditzlaguntzailea> helbidetik eskuratua

(Erref4). <http://tinyurl.com/adizkitegia> helbidetik eskuratua

(Erref5). <http://developer.android.com/guide/topics/data/data-storage.html> helbidetik eskuratua

## 7.2 Laguntza-gidak

*Androiden garatzaile gida ofiziala:*

<http://developer.android.com/index.html>

*Android wikipedia*

[http://en.wikipedia.org/wiki/Android\\_%28operating\\_system%29](http://en.wikipedia.org/wiki/Android_%28operating_system%29)

*Android zalantzak*

<http://stackoverflow.com/questions/tagged/android>

*Android hasi berrientzat*

<http://www.elandroidelibre.com/2014/02/aprende-android-en-20-conceptos-empezando-a-programar-para-android.html>

*ADT plugina*

<http://developer.android.com/tools/sdk/eclipse-adt.html>

*Android Studio*

<http://developer.android.com/tools/sdk/eclipse-adt.html>

Android diseinua

<http://developer.android.com/design/index.html>

*Android bertsioak*

[http://en.wikipedia.org/wiki/Android\\_version\\_history](http://en.wikipedia.org/wiki/Android_version_history)

Android eclipserekin uztartzeko

[http://en.wikipedia.org/wiki/Android\\_version\\_history](http://en.wikipedia.org/wiki/Android_version_history)

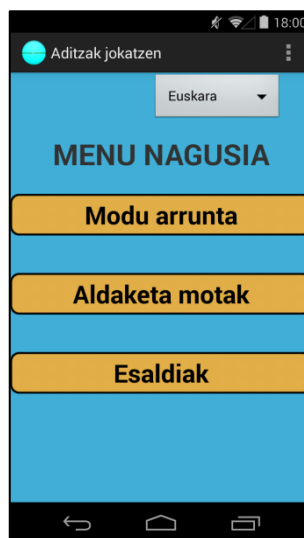
## 8. Eranskinak

---

### 8.1 Eskuliburua

Garatu den aplikazio honek 3 aukera nagusi ditu: Modu arruntean erabiltzea, Aldaketa motak kudeatuz erabiltzea edota esaldi mailan aritzea. Interfaze aldetik berriz, orientazio bertikala nahiz horizontala onartzen ditu. Gainera, bateragarritasun osoa lortzeko asmoz, norbaitek zerbait ikusi ezin balu (zerrenda osoa ezin ikusi pantaila txikiegia delako adibidez) Scroll baten bidez elementu guztiak ikus ditzake.

Aplikazioa kargatzen denean, menu bat erakusten da aipatutako 3 aukera horiek emanez. Interfazerako nahi den hizkuntza aukeratu ondoren (hizkuntza edozein momentutan alda daiteke) erabiltzaileak egindako aukerara joaten da.



17.irudia: Menu nagusia

#### 8.1.1 Modu arrunta

Modu arruntean hiru adizki emanda adizkien bilaketa egin daiteke analogia bidez emaitzak erakutsiz.



18.irudia: Modu arrunta

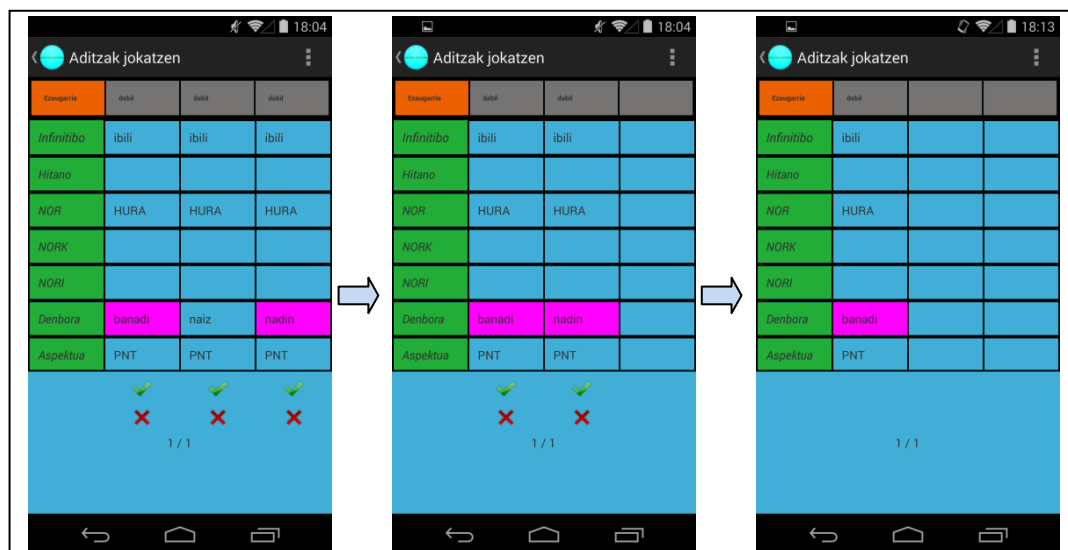
Adizkiak idazten doazen heinean testu-kutxaren alboan ikur bat ager daiteke aplikazio osoan zehar. Honen esanahiak hauek izan daitezke:

- Ikurrik ez: Adizkia bakarra da eta existitzen da.
- ⚠ ikurra: Adizkia anbigua da eta beraz, ezaugarri ezberdinak dituzten adizkiak daude bertan.
- ✗ ikurra: Idatzitako adizkia ez da existitzen.

Aurreko irudiko kasuan ikus daiteke ‘dabil’ adizkia anbigua dela eta aplikazio honek desanbiguatzeko aukera ematen du. Horretarako, ikurraren gainean klik egin eta konparaketa taula bat adierazten da anbigutasun ezaugarriekin eta anbiguoak diren ezaugarriak kolorez bereizita. Bertan, zutabe bakoitzean ✓ eta ✗ ikurrak ikus daitezke.

Lehenengo ikurrean klik eginda, zuzenean aukeratutako zutabeari dagokion ezaugarriak nahi direla esaten da eta zuzenean ezaugarri hori duen adizki desanbiguatuekin aurreko pantailara itzultzen da sistema. Aldiz, ezabatzeko botoiarekin, zutabeak banan bana kentzen dira gutxienez bat geratu arte.

Aurreko kasuan adibidez, emaitzaren bat ateratzen den modu denbora orainaldiko indikatiboa da (‘naiz forma’) eta demagun desanbiguatzean ‘banadi’ forma hartu nahi dugula soilik. Hori egiteko honelako prozesuak jarraituko luke, zutabeak banaka kenduz:



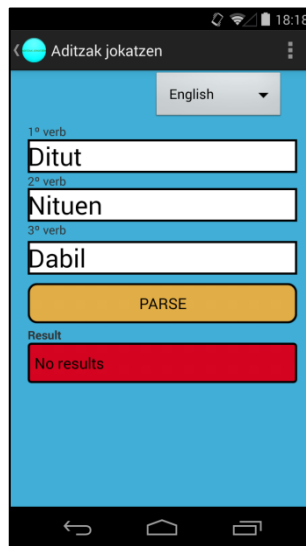
19.irudia: Zutabe ezabaketa prozesua

Hau honela, ‘dabil’ adizkia desanbiguatu ondoren, ‘banadi’ forma eman zaionez, ez du emaitzarik itzuliko.



20.irudia: Desanbiguatutako adibidea

Gainera, ikus daitekeen moduan arrisku ikurra joan egin da adizkia desanbiguatuta baitago. Pantaila bera esaterako ingelesez ikusi nahiko balitz, nahikoa da goiko aukeran ‘English’ aukeratu eta interfazea dinamikoki aldatuko da.



21.irudia: Hizkuntza aldaketaren adibidea

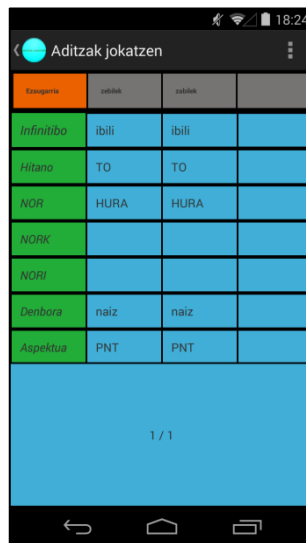
### 8.1.2 Aldaketa motak

Aldaketa motak kudeatuz erabili nahi bada aplikazioa, aldaketa moten zerrendako lehenengoa lehenetsia izango da beti, ✓ ikurra duena hain zuzen. Honen helburua da norbaitek beti aldaketa mota berdina erabili nahi badu lehenesteko aukerak izatea saio bakoitzean aldatzen ibili behar izan gabe. Aldaketa mota berririk sortzen ez bada, derrigorrez Hitanozko bi moduak (toka eta noka) agertuko dira eta horien gainean galdetu daiteke.



22.irudia: Eraitza anbiguoaren adibidea

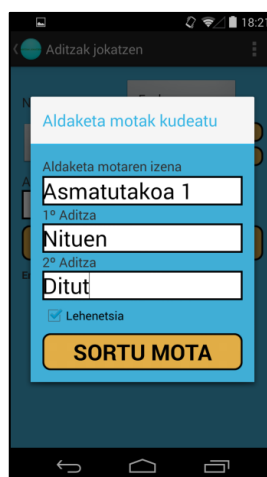
Eraitza anbigua denez, eraitzan klik egin eta bien arteko ezberdintasunak ikus daitezke konparaketa taula baten bitartez.



23.irudia: Emaizten arteko konparaketa

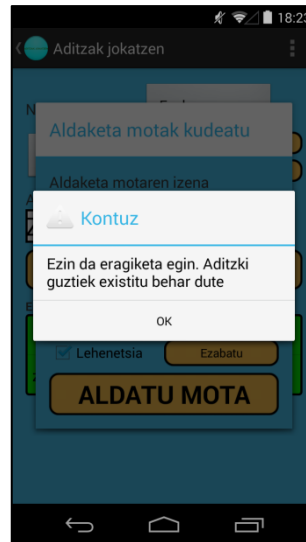
Beti agertzen den hitanoekin konforme ez bada, aldaketa motak sortu, aldatu (nahi izanez gero lehenetsi) eta ezabatu egin daitezke. Baldintza batekin beti ere hitanozko bi formak ezin baitira ez aldatu eta ez ezabatu. Kasu hauetan aldatzeko eskatuz gero informazioa erakutsiko da baina lehenesten bakarrik utziko du.

Aldaketa mota sortzeko, zerrendaren alboan agertzen den “SORTU” botoia klikatu eta dialogo bat azaleratzen da pantailan aldaketa motak sortzeko. Bertan, aldaketa motaren izena, bi adizkiak (Anbigua edo ez aukeran aurretik erakutsi den moduan) eta lehenetsia izango den edo ez idatzi behar dira.



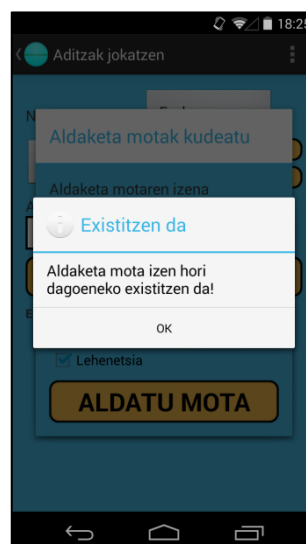
24.irudia: Aldaketa mota sortzen

Adizki hauek ordea modu arruntean ez bezala, behar beharrezko dira aldaketa motak sortu edo eguneratzeko, aldaketa mota faltsurik egon ez daitezen. Aldaketaren bat gaizki badago, sortzerakoan ohartarazi egiten da.



25.irudia: Adizkia ez da existitzen

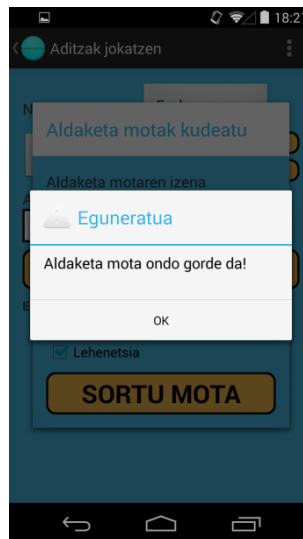
Aldaketa motaren izena gakotzat hartzen denez, ezin da errepikatu eta existitzen bada, honako mezua adierazten da:



26.irudia: Aldaketa mota existitzen da



Aldiz, eragiketa ondo egin bada, honako mezua agertuko da:



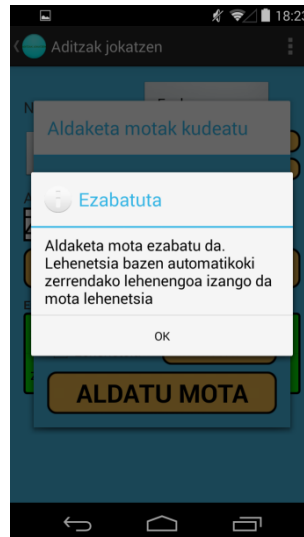
27.irudia: Aldaketa mota ondo sortua

Aldiz existitzen den aldaketa mota aldatu edo ezabatu nahi bada, “ALDATU” botoiari klikatu behar zaio. Honela, sortzerakoan agertzen zen antzeko dialogo bat ikus daiteke baina ezabatzeko botoiarekin eta aldaketa motaren datuak dialogoan txertatuta



28.irudia: Aldaketa mota aldatzeko dialogoa

“ALDATU MOTA” botoian klik eginda eguneratu egingo da eta “Ezabatu” botoia klikatuta berriz ezabatu eta zerrendatik ateratzen da ohartarazpen hau erakutsiz.

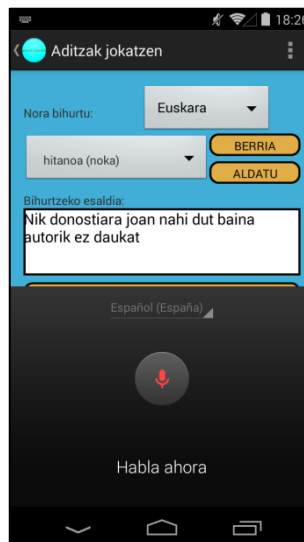


29.irudia: Aldaketa mota ezabatua

### 8.1.3 Esaldiak

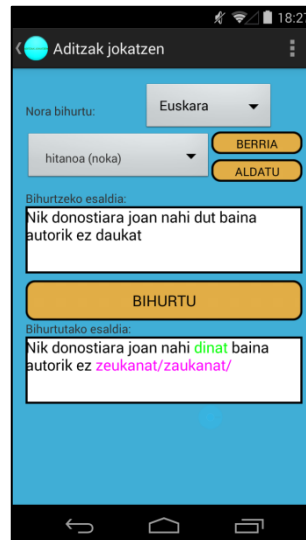
Azkenik, hirugarren aukera esaldi mailan lan egitea da. Bertan azaldu berri diren aldaketa moten kudeaketa edo adizkien desanbiguatzeko aukeraz gain, esaldi mailan lan egiteko modua ematen du.

Horretarako aldaketa mota aukeratu eta esaldia idatzi edo ahotsez mikrofonotik esan behar da



30.irudia: Esaldia mikrofono bidez ahoskatzen

‘BIHURTU’ botoia sakatu ondoren emaitza agertzen da azpiko testu-kutxan. Bien bitartean gainera, emaitza entzuten da gailuaren bozgorailuetatik , beti ere hauek aktibatuta badaude.



31 irudia: Esaldiaren emaitzaren adibidea

Ikus daitekeen moduan, emaitza bakarra denean, berdez koloreztatuta agertzen da eta aldiz anbigua bada, kolore arrosez aukera guztiak idazten dira. Hala ere, kontuan izan behar da bozgorailutik entzungo dena beti lehenengo aukera izango dela.

## 8.2 Kategorien itzulpena

Datu-basean kategoriaren izena A1,A2,B1,C1,B5B moduko kodeekin agertzen da. Aplikazioan hori edota izena jarri beharrean (oso izen luze eta erabiltzaileentzat ulergaitzak askotan), adibideak jarri dira; naiz, nintzen,nadin,etab. Hona hemen taula batean adierazita esanahi guztiak kodearen arabera:

| Laburdura | Kategoria                                     | Adibidea     |
|-----------|-----------------------------------------------|--------------|
| A1        | Indikatibozko orainaldia                      | Naiz         |
| A2        | Indikatibozko geroaldi arkaikoa (supositiboa) | Naizateke    |
| A3        | Subjuntibozko orainaldia                      | Nadin        |
| A4        | Subjuntibozko baldintza                       | Banadi       |
| A5        | Ahalerazko orainaldia                         | naiteke      |
| B1        | Indikatibozko lehenaldia                      | nintzen      |
| B2        | Indikatibozko baldintza (ondorioa,orain-gero) | Nintzateke   |
| B3        | Indikatibozko baldintza (ondorioa,lehen)      | nintzatekeen |
| B4        | Indikatibozko baldintza (aurrekoa)            | Banintz      |
| B5        | Subjuntibozko lehenaldia                      | Nendin       |
| B6        | Subjuntibozko baldintza (lehenaldia)          | Banendi      |
| B7        | Ahalerazko lehenaldia                         | Ninteke      |
| B8        | Ahalerazko lehenaldi urruna                   | Nintekeen    |
| B5A       | Subjuntibozko alegiazkoa                      | Ledin        |
| B5B       | Subjuntibozko lehenaldia                      | Zedin        |
| C         | Inperatibozko orainaldia                      | Hadi         |

## 8.3 Proiektuaren helburu dokumentua

### 8.3.1 Sarrera

Proiektu honen ardatza adizkiak izango dira. Aditzen artean jolasteko aukera emango du garatuko den proiektu honek. Berau, Android ingurunerako sortuko da mugikorretan modu erabilgarri eta xume batean erabiltzaileek aditzei buruz ikasi edo zalantzak argitu ditzaten. Garatuko den programa honek, 1.mailako “Oinarrizko programazioa” irakasgaiko praktikaren enuntziatuak egiten duena egingo du. Erabiltzaileak hiru aditz-forma sartuko ditu eta aditz forma hauetako bakoitzak beraien ezaugarri gramatikalak izango dituzte gordeta programan nonbait. Ezaugarri hauek hasiera batean euskaraz honakoak izango dira:

- Lema: Zein aditz trinkotik datorren
- Hitano: Hikaz adierazita dagoen ala ez , Toka eta Noka bereiziz
- Nor: Adizkiaren nor pertsona (Baldin badu)
- Nork: Adizkiaren Nork pertsona (Baldin badu)
- Nori: Adizkiaren nori pertsona (Baldin badu)
- Modu-denbora: Modu eta denboraren ezaugarriak (Adib. Subjuntibozko lehenaldia)

Hau horrela sistemak aditz forma horien ezaugarri gramatikalak erabiliz gero eta beraien arteko nabarmentasun nagusienak banatuz gero, aditz forma bat hitanora pasatzeko edota orainalditik lehenaldia pasatzeko, ez da beharrezkoa gramatika konplexura jotzea eta ezaugarri horietatik abiatuta lor daiteke nahi den aditz forma.

Adibidez, *Ditut-Ditiat-Daukat* aditz forma hirukoteak, hizkuntza ezaugarriak aurkitu ondoren honako emaitza emango luke.

| Adizkia       | Lema  | Hitanoa | Nor   | Nori | Nork | Modu denbora             |
|---------------|-------|---------|-------|------|------|--------------------------|
| <b>ditut</b>  | Ukan  | -       | Haiek | -    | nik  | Indikatibozko orainaldia |
| <b>ditiat</b> | Ukan  | toka    | Haiek | -    | nik  | Indikatibozko orainaldia |
| <b>daukat</b> | Eduki | -       | Hura  | -    | nik  | Indikatibozko orainaldia |

Erlazio-ezberdintasunak begiratzuz gero , honakoa lortuko litzateke:

| Aditza | Lema  | Hitanoa | Nor  | Nori | Nork | Modu denbora             |
|--------|-------|---------|------|------|------|--------------------------|
| ?????  | Eduki | toka    | hura | -    | nik  | Indikatibozko orainaldia |

Beraz, emaitza ezaugarri hori duen edo duten aditz formak izango dira. Kasu honetan, aditz zerrenda begiratzuz gero bi emaitza posible agertzen dira, hauek izango direlarik emaitzak.

| Adizkia | Lema  | Hitanoa | Nor  | Nori | Nork | Modu denbora             |
|---------|-------|---------|------|------|------|--------------------------|
| zaukaat | Eduki | toka    | hura | -    | nik  | Indikatibozko orainaldia |
| zeukaat | Eduki | toka    | hura | -    | Nik  | Indikatibozko orainaldia |

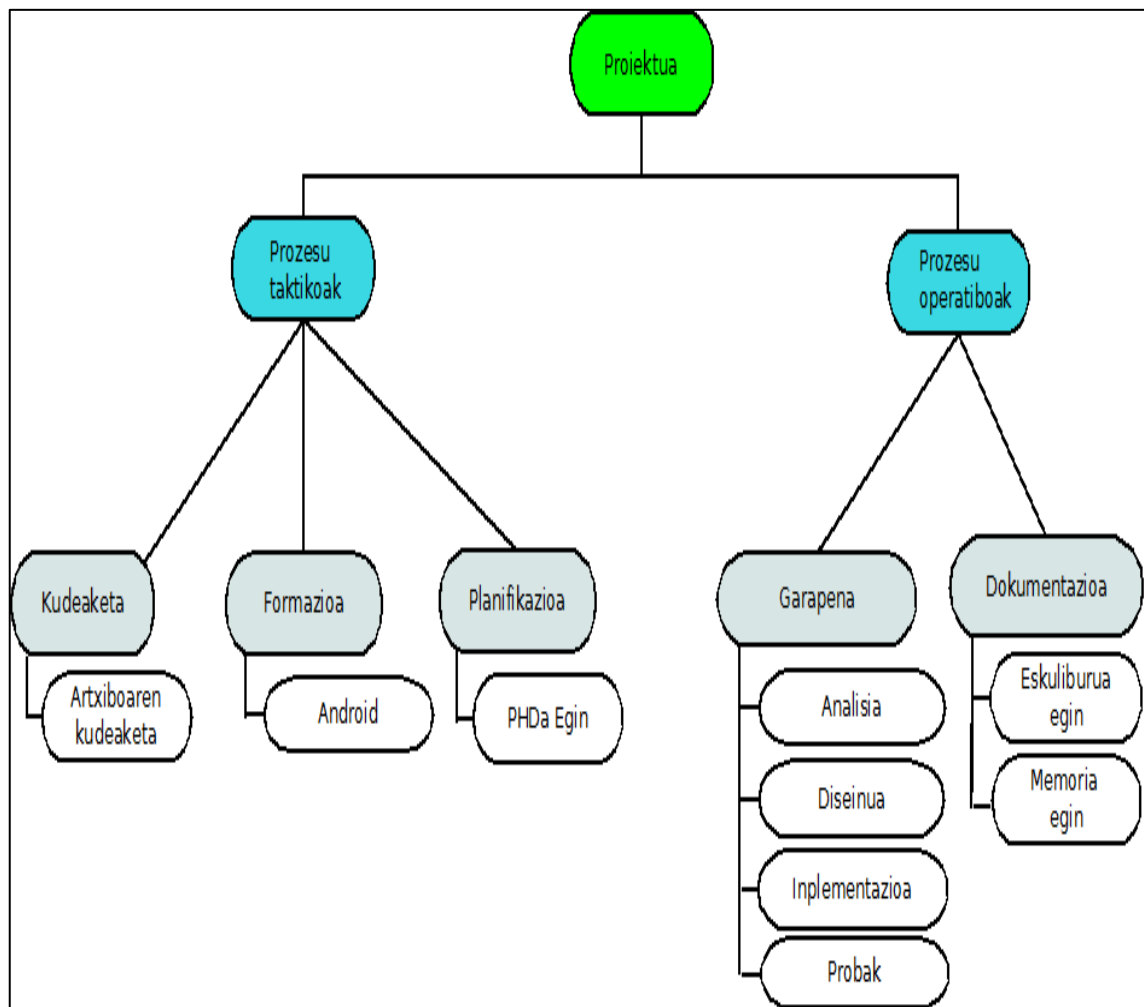
### 8.3.2 Helburuak

Proiektuaren helburuak honakoak izango dira lehentasunez ordenatuta:

1. mailako “Oinarrizko programazioa” irakasgaiko praktikaren enuntziatua egiten duena egitea. Hau da, aditz forma batzuk sartu eta aditz horien arteko konbinazioa den aditz-forma berria lortu bere ezaugarriak begiratzuz. Gainera kontuan izan behar da aditz-forma batzuk anbiguoak direla eta horiek ere kontrolatu beharko dira.
- Ikasgai horretan Ada programazio lengoia egin bada ere, proiektu honetan Javan garatuko da Androidera pasa ahal izateko.
2. Aurreko lana Androiden martxan jartzea. Erabiltzaileak Android bertsioaren arabera ahotsez esan dezake. Gainera, hobekuntza gisa, emaitza irakurri eta soinu bidez azaltzea ere egin daiteke, beti ere une honetan martxan dagoen beste proiektu bat amaitu bada.
3. Interfaze eleanitza egitea. Euskaraz gain, gaztelania eta ingelesezko hizkuntzen aukera izatea.
4. Erabiltzaileari erabilerraztasuna emateko, aditzei egiten zaizkien aldaketa motak definitu eta gordetzea. Adibidez, orainalditik lehenaldira pasatzeko aditz konbinazioak gorde eta erabiltzaileak azken aditz forma orainaldian emanda sistemak aditz hori lehenaldian aurkitu beharko du.
5. Programak hizkuntz gehiagorekiko malgutasuna izatea, hau da programa hau euskaraz gain gaztelania edo ingelesezko aditz irregularrekin bateragarria izatea etorkizunean begira hauek inplementatzeko ahalik eta aldaketa gutxien egin ditzan garatzaileak.
6. Esaldi mailan lan egitea. Gordetako aldaketa mota bat aukeratu, esaldi bat esan eta esaldi horren aditz guztiak mota aldatuta itzultzea.

### 8.3.3 Lanaren Deskonposaketa Egitura diagrama(LDE)

Honakoa da plangintzako Gantt Diagrama:



31.diagrama: Plangintzako Lanaren deskonposaketa egitura

## 8.3.4 Proiektuko ataza eta azpi-atazak

### 8.3.4.1 Ataza *taktikoak*

K: Kudeaketarekin lotutakoak

K1: Artxiboaren kudeaketa

F: Formazioarekin lotutakoak

F1: Androiden formakuntza

P: Planifikazioarekin lotutakoak

P1: Proiektuaren Helburu Dokumentua egin

### 8.3.4.2 Ataza *operatiboak*

G: Garapenarekin lotutakoak

G1: Analisia

G11: Analisi funtzionala

G2: Diseinua

G21: Aurkezpen maila diseinatu

G22: Datu maila diseinatu

G22: Probak diseinatu

G3: Inplementazioa

G31: Arkitektura definitu

G32: Aurkezpen maila inplementatu

G33: Datu maila inplementatu

G34: Androiden integratu

G4: Probak

G41: Arazketak egin

G42: Probak egin

G43: Martxan jarri

D: Dokumentazioarekin lotutakoak

D1: Eskuliburua egin

D2: Memoria



### 8.3.5 Planifikazioa

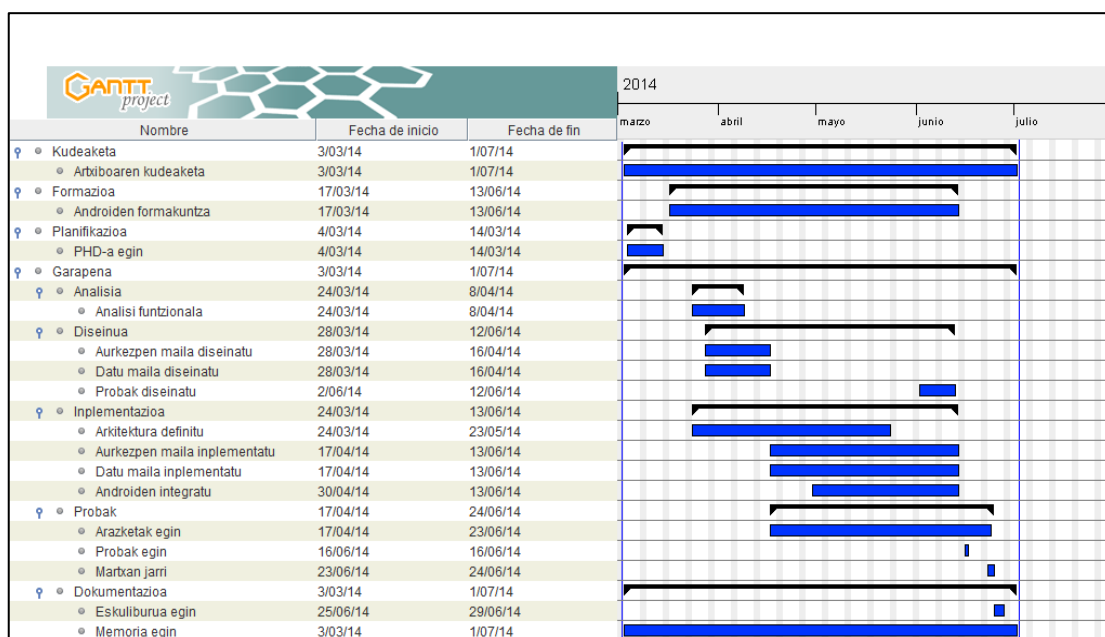
#### 8.3.5 .1 Planifikazioaren estimazioa

| Atazak                                   | Emangarriak                  | Estimazioa    |
|------------------------------------------|------------------------------|---------------|
| <b>K:Kudeaketa</b>                       |                              | <b>12:00</b>  |
| <b>K2: Artxiboaren kudeaketa</b>         | Diskoa                       | 12:00         |
| <b>F:Formazioa</b>                       |                              | <b>30:00</b>  |
| <b>F1: Androiden formakuntza</b>         |                              | 30:00         |
| <b>P:Planifikazioa</b>                   |                              | <b>14:00</b>  |
| <b>P1: PHD-a egin</b>                    | PHD                          | 14:00         |
| <b>G:Garapena</b>                        | Aplikazioa                   | <b>236:00</b> |
| <b>G1: Analisia</b>                      | Sekuentzia diagramak         | 25:00         |
| <b>G11: Analisi Funtzionala</b>          | EKE,Domeinuaren eredua       | 25:00         |
| <b>G2: Diseinua</b>                      | Sistema sekuentzia diagramak | 58:00         |
| <b>G21: Aurkezpen maila diseinatu</b>    |                              | 25:00         |
| <b>G22: Datu maila diseinatu</b>         |                              | 25:00         |
| <b>G23: Probak diseinatu</b>             |                              | 8:00          |
| <b>G3: Implementazioa</b>                | Kodea                        | 110:00        |
| <b>G31: Arkitektura definitu</b>         |                              | 15:00         |
| <b>G32: Aurkezpen maila implementatu</b> |                              | 25:00         |
| <b>G33: Datu maila implementatu</b>      |                              | 40:00         |
| <b>G34: Androiden integratu</b>          |                              | 30:00         |
| <b>G4: Probak</b>                        |                              | 43:00         |
| <b>G41: Arazketak egin</b>               |                              | 25:00         |
| <b>G42: Probak egin</b>                  |                              | 10:00         |
| <b>G43: Martxan jarri</b>                |                              | 8:00          |
| <b>D:Dokumentazioa</b>                   |                              | <b>66:00</b>  |

|                             |             |               |
|-----------------------------|-------------|---------------|
| <b>D2: Eskuliburua egin</b> | Eskuliburua | 6:00          |
| <b>D3: Memoria egin</b>     | Memoria     | 60:00         |
| <b>GUZTIRA</b>              |             | <b>358:00</b> |

### 8.3.5.2 Gantt diagrama

Honakoa da plangintzako Gantt diagrama



32.diagrama: Plangintzako Gantt Diagrama

### 8.3.6 Eginkizunak

Proiektu honek inplementazio ugari izango du baina inplementazio guztien moduan ia beti hobekuntzak egin daitezke, honako taula honetan adierazten direlarik izango diren eginkizunak hobekuntza-mailaren arabera, denbora epe bat baitago lana egiteko , 2014ko Uztailak 1 hain zuzen ere eta epe horretan ahalik eta hobekuntza gehien egingo dira zuzendariak bidalitako eginkizun minimoak bete ondoren.

| <b>Eginkizuna</b>                          | <b>Azalpena</b>                                                                                                                         | <b>Hobekuntza-maila</b> |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| <b>Adako praktika Androidera eraman</b>    | Adan ‘Oinarrizko programazioa’ ikasgaiaren eginiko praktika Androiden inplementatzea, hau da 3 adizki jarrita adizki emaitza itzultzea. | Egin beharreko minimoa  |
| <b>Adizkien anbiguitasuna tratatu</b>      | Sartzen den 3 adizkietakoren bat anbiguo bada, hori ere tratatu eta emaitzan sartu                                                      | Egin beharreko minimoa  |
| <b>Esaldi osoak bihurtzea</b>              | Adizki bakarra pasa ordeez esaldi osoa pasa eta bihurtzea                                                                               | Hobekuntza baxua        |
| <b>Emaitzak entzutea</b>                   | Adizkiak ahotsez esateko aukera eta lortzen diren emaitzak bozgorailu bidez entzuteko aukera izatea                                     | Hobekuntza baxua        |
| <b>Interfaze eleanitza</b>                 | Interfazea eleanitza izatea eta gutxienez euskara,gaztelania eta ingeleserako interfazea sortzea                                        | Hobekuntza ertaina      |
| <b>Aldaketa motak sortzea</b>              | 3 adizki jartzen ibili beharrean, erabiltzaileak bere aldaketa mota propioak jartzeko aukera izatea lehenengo bi aldaketa motak emanez. | Hobekuntza ertaina      |
| <b>Emaitzak konparatzeko aukera izatea</b> | Ateratzen den emaitza anbiguo bada, beraien arteko konparaketa ikustea taula bidez.                                                     | Hobekuntza ertaina      |
| <b>Interfaze horizontala</b>               | Erabiltzaileak nahi badu, interfazea modu horizontalean ikusteko aukera ematea gailua iraultzen denean.                                 | Hobekuntza ertaina      |
| <b>Adizkiak desanbiguatzea</b>             | Erabiltzailea adizki anbiguo bat idatzi orduko jakinarazi eta desanbiguatzeak aukera ematea.                                            | Hobekuntza altua        |
| <b>Aldaketa motak kudeatzea</b>            | Aldaketa motak sortzeaz gain, berauek ezabatu, aldatu edota lehenetsia jartzeko aukera izatea.                                          | Hobekuntza altua        |
| <b>Tabletarako bateragarria izatea</b>     | Tabletarako diseinu espezifikoak egin pantaila oso handietarako.                                                                        | Hobekuntza altua        |
| <b>Hizkuntza</b>                           | Euskarazko adizkiez gain, gaztelania eta                                                                                                | Hobekuntza altua        |

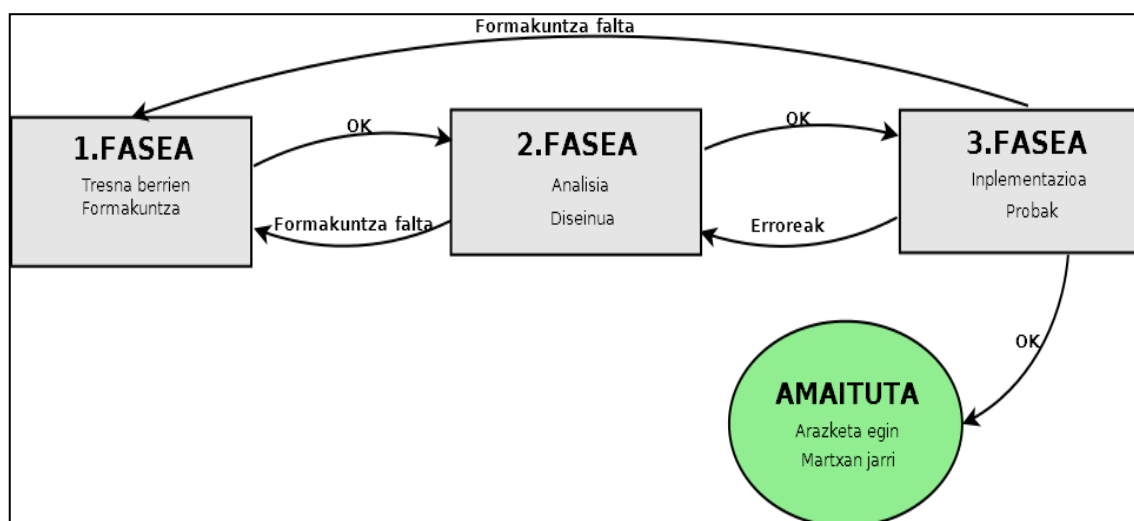
|                                             |                                                                                                                                                                                   |                      |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| <b>ezberdinetara moldatzea</b>              | ingelesezko aditz irregularrekin ere balio izatea aplikazioak.                                                                                                                    |                      |
| <b>Lematizazioa erabiltzea</b>              | Esaldiak bihurtzerako orduan banan banako bilaketa egin ordez esaldia lematizatzea emaitza hobeak lortzeko.                                                                       | Hobekuntza altua     |
| <b>Herrien arteko hizkuntzak aplikatzea</b> | Webgune batean gehien erabiltzen diren adizkiak jarri eta erabiltzaileek beraien herrian nola esaten den idaztea, honela aldaketa mota herri baten arabera aplikatu ahal izateko. | Hobekuntza oso altua |

### 8.3.7 Garapen prozesua

Proiektu honek orokorrean 3 fase izango ditu. Lehenik eta behin, proiektuaren garatzaileak erabili behar dituen teknologia berriak aztertuko dira ahalik eta formakuntza gehiago izateko diseinatzeko eta implementatzeko garaian

Ondoren, analisi funtzional bat eta egingo den aplikazioaren diseinu bat egingo da eta diseinua onartzen bada implementazio fasera pasako da, hirugarren fasera hain zuzen.

Implementazioko fase honek, diseinuan egindakoari jarraitu beharko dio salbuespen handirik gertatzen ez den bitartean. Behin implementatuta probak egingo dira eta gehienak, aurreko fasean diseinatu ditugun probak izango dira. Gerta daiteke implementazioan proba berriak egin beharra izatea errore berriak saihesteko. Probak ondo daudela ikusten bada, aplikazioaren garatzaileak beste langile bati/batzuei emango die denbora batean probak egin ditzaten eta errorerik bada jakinarazi dezaten. Errorerik ez bada aurkitzen proiektuaren arazketa egin eta martxan jartzeari ekingo zaio. Azkenik, tresna berriekin lanean ari garenez, baliteke edozein momentutan formazio beharra izatea eta bai 2.fasetik bai 3.fasetik, 1.fasera joateko arriskua daukagu.



33 diagrama: Fase eskema

### 8.3.8 Lan metodologia

#### Artxiboaren kudeaketa

Proiektu hau garatzailearen ordenagailu pertsonalean burutuko da egiten den guztien segurtasun kopiak egingo dira lana egiten den bakoitzean. Uneoro gutxienik kopia berdina egon beharko du bai garatzailearen PCan baita Dropboxeko hodeian biltegitratuta eta proiektuaren amaieran egin den guztiarekin disko bat sortuko da.

#### Erabiliko den teknologia

Karrera amaierako proiektu hau esan bezala garatzailearen ordenagailu pertsonalean egingo da eta garapenerako Windows 7 sistema eragilea erabiliko da eta programazio eta diseinurako Eclipse ingurunea erabiltzea aurreikusten da

Memoriaren kasuan berriz, Microsoft Word 2007 tresnarekin garatuko da eta bertako diagramak *DIA Diagram Editor* erabiliz egingo dira. Egin beharreko Gantt diagramak berriz Gantt Project-ekin egingo dira eta proiektuaren kudeaketako grafikoak aldiz Microsoft Excel 2007rekin.

### 8.3.9 Arriskuak

#### 8.3.9.1 Arriskuen zerrenda

- A01. Datuen galera
- A02. Espero gabeko lan gehikuntza
- A03. Lantoki desegokia aukeratzea
- A04. Bezeroarekin gaizki ulertuak
- A05. Ezagutza falta
- A06. Epeetan aldaketak

#### 8.3.9.2 Arriskuen azalpena

Honakoak izan daitezke proiektuan zehar aurkituko diren arazoetako batzuk:

- A01. Datuen galera

**Deskribapena:** Datuak galtzen direnean.

**Larritasun maila:** Aldakorra, galdutako datuen garrantzia eta tamainaren arabera.

**Eragina:** Galdutako lana berregin beharrak planifikazioan atzerapenak ekarriko ditu.

**Aurrezaintza plana:** Bertsioen kontrolerako biltegi bat erabiliko da, eta garatzailearen ordenagailuan segurtasun kopiak egingo dira.

**Kontingentzia plana:** Galdutako bertsioaren aurreko bertsioa berreskuratu eta galdutako lana berriz egin.

- A02. Espero gabeko lan gehikuntza

**Deskribapena:** Proiektu zuzendariak, espero ez denean, proiektuan gauza berriak sartzea eskatzea.

**Larritasun maila:** Aldakorra, gehitzea eskatzen den lanaren zailtasun mailaren eta planifikazioan eragin dezakeen atzerapenaren arabera.

**Eragina:** Planifikazioan atzerapenak, esfortzua handitzea edota eskaera guztiak epean ez betetzea.

**Aurrezaintza plana:** Lana bukatzeko epeak baino lehenago bukatzea.

**Kontingentzia plana:** Lan berria ongi banatu eta denbora gehiago sartzea, eta beharrezkoa bada garrantzia gutxiko atazak egiteari uztea.

#### A03. Lantoki desegokia aukeratzea

**Deskribapena:** Lanak egiteko edo bilerak burutzeko lekua desegokia izatea (txikiegia, denentzat lekurik gabe, zaratatsua, lanerako beharrezkoak diren tresnak falta izatea...).

**Larritasun maila:** Aldakorra, lekuaren desegokitasunaren arabera.

**Eragina:** Planifikazioan atzerapenak, lanean erroreak (kontzentrazio falta dela eta...).

**Aurrezaintza plana:** Lanerako beti eskuragarri egongo den leku eroso bat aurkitzea.

**Kontingentzia plana:** Ahalik eta azkarren beste leku bat aurkitu, eta ezin bada bertara moldatzen saiatu, falta diren gauzaren bat eramanez.

#### A04. Zuzendariarekin gaizki ulertuak

**Deskribapena:** Proiektua jasotzean, proiektu zuzendariak nahi duenaren ordeztasun, beste zerbait ulertzea.

**Larritasun maila:** Aldakorra, gaizki ulertuaz zenbat eta beranduago konturatu geroz eta larriagoa izango da.

**Eragina:** Zuzendariak egiterik nahi ez zituen gauzak egitea (lan alferria), edota zuzendariak eskaturiko gauzak ez burutzea.

**Aurrezaintza plana:** Denbora pasatzen doan heinean, lana bezeroari erakutsi.

**Kontingentzia plana:** Zuzendariarekin bilera sakon bat egin gauzak ondo ulertu arte, ondoren egin behar ez zen zatia ahalik eta gehien aprobetxatu.

#### A05. Ezagutza falta

**Deskribapena:** Ataza bat egiteko erabili beharreko tresneria erabiltzen ez jakitea.

**Larritasun maila:** Aldakorra, erabili beharreko tresnaren garrantziaren, edota konplexutasunaren arabera.

**Eragina:** Ataza jakin batzuetan atzerapena, eta honekin batera proiektuaren atzerapena.

**Aurrezaintza plana:** Erabili beharreko tresnak aurreikusi eta hauei buruz dokumentatu.

**Kontingentzia plana:** Tresna erabiltzen dakien taldekideren bati laguntza eskatu edo lana egin aurretik ongi dokumentatu.

A06. Epeetan aldaketak

**Deskribapena:** Emangarri bat (edo produktua/zerbitzua bera) entregatzeko eguna aurreratu, eta lanerako denbora gutxiago izatea.

**Larritasun maila:** Aldakorra, epe aldaketa zenbateko aurrerapenarekin adierazten den, eta lana bera zenbateraino aurreratua dagoen.

**Eragina:** Produktuan kalitate galera gerta daiteke, denbora laburrean lan gehiago egin beharko baita.

**Aurrezaintza plana:** Lanak ezarritako epea baino lehen bukatuko direla planifikatzea.

**Kontingentzia plana:** Esfortzua handitu beharko da, eta beharrezkoa bada garrantzia gutxiko atazak egiteari utzi beharko zaio.