

POKEDEX DOKUMENTAZIOA

Eneko Puente, Yassin Mehani, Asier Barrio, Yoel Justel, Xinyan Wang eta Unai Rodriguez

Hasiera	6
Sarrera	6
Aldaketen changelog-a	7
1. Bigarren entregagarrirako aldaketak	7
2. Laugarren entregagarriko aldaketak	7
3. Amaierako entregagarriko aldaketak	7
Funtzionalitateen arduradunak	8
1. Entregagarria (Erabilpen kasuak)	9
Erabilpen kasuen diagrama osoa	9
1.1 Erabiltzaileen kudeaketa	10
1.1.1 Hasi saioa	10
1.1.2 Erregistratu	12
1.1.3 Lagun Zerrenda	14
1.1.3.1 Laguna Gehitu	16
1.1.3.2 Laguna Ezabatu	18
1.1.3.3 Lagun baten notifikazioak kendu	20
1.1.4 Menu Admin	21
1.1.4.1 Erabiltzailearen Datuak Editatu	23
1.1.4.2 Kontua Bilatu	25
1.1.4.3 Kontua Ezabatu	27
1.1.4.4 Administratzaile Bihurtu	29
1.1.5 Profila eguneratu	31
1.2. Pokemon taldeak sortzea	33
1.2.1 Talde menua	33
1.2.1.1 Talde sorkuntza	35
1.2.1.2 Talde edizioa	38
1.3. Changelog	43
1.3.1 Jakinarazpen menua	43
1.3.1.1 Bilatu	45
1.4. Pokemonen lista osoa eta bilaketa filtroekin	46
1.4.1 Pokedex-a	46
1.5. Chatbot-a	49
1.5.1 Mugimendu ikasgarriak	49
1.5.2 Taldeko pokemon hoberena	52
1.5.3 Ahuleziak eta indarrak	54
1.5.4 Eboluzio katea	57
1.6. Hautazko funtzionalitatea	61
1.6.1 Itemdex	61
1.6.1.1 Item-a bilatu	63
1.6.1.2 Iragazketa jarri	65
2. Entregagarria (Domeinuaren eredua)	67

2.1 Domeinuaren eredua osoa	67
2.1.1 Azalpen gehigarriak	67
2.2 Entitateak	68
2.2.1. Erabiltzailea	68
2.2.2. Taldea	68
2.2.3. Pokemon (Talde)	68
2.2.4. Pokemon (Pokedex)	68
2.2.5. Mugimendua	69
2.2.6. Abilezia	69
2.2.7. Mota (Pokemon)	69
2.2.8. Item	69
2.2.9. Mota (Item)	69
2.3. Erlazioak	70
2.3.1. Notifikatu	70
2.3.2. Jarraitzen du (Erabiltzaile -> Erabiltzaile)	71
2.3.3. Dauka (Erabiltzaile -> Pokemon (Talde))	72
2.3.4. Dauka (Erabiltzaile -> Talde)	73
2.3.5. Dauka (Talde -> Pokemon (Talde))	74
2.3.6. Dauka (Pokemon (Talde) -> Abilezia)	75
2.3.7. Izan dezake/ditzake	76
2.3.8. Ikas dezake/ditzake	77
2.3.9. Dak(izk)i	78
2.3.10. Da (Pokemon (Talde) -> Pokemon (Pokedex))	79
2.3.11. Eragina	80
2.3.12. Da (Pokemon (Pokedex) -> Mota)	81
2.3.13. Eboluziona dezake	82
2.3.14. Da (Item -> Mota (Item))	83
2.3.15 Da (Mota (Pokemon) -> Mugimendua)	84
3. Entregagarria (Software-aren probak)	85
3.1. Erabiltzaileen kudeaketa (Yassin Mehani)	85
3.2. Pokemon taldeak sortzea (Unai Rodriguez)	89
3.3. ChangeLog (Eneko Puente)	92
3.4. Pokemon lista eta bilaketa (Asier Barrio)	93
3.5. ChatBot (Yoel Justel)	95
3.6. ItemDex (Xinyan Wang)	97
3.7. AI-ren erabilpena	99
4. Entregagarria (Klase diagrama, datu basea eta komunikazio diagrama)	100
4.1 Datu basea	100
4.1.1 Azalpen gehigarriak	100
4.2 Klase diagrama	101
4.2.1 Azalpen gehigarriak	101
4.3 Komunikazio diagrama	102

4.3.1 Erabiltzaileen kudeaketa (Eneko Puente)	102
4.3.1.1 Saioa Hasi	102
4.3.1.2 Erregistratu	103
4.3.1.3 Profila Eguneratu	104
4.3.1.4 Lagun Zerrenda	105
4.3.1.5 Menu Admin	107
4.3.2 Pokemon taldeak sortzea (Asier Barrio)	110
4.3.2.1 Talde berria sortu eta Pokemon sartu	110
4.3.2.2 Taldea gorde eta ezabatu	114
4.3.2.3 Taldea editatu	116
4.3.3 Changelog-a (Unai Rodriguez)	120
4.3.3.1 Notifikazioak kargatu izen baten menpe (Bilatu botoia)	120
4.3.4 Pokedex-a (Yassin Mehani)	122
4.3.5 Chatbot-a (Xinyan Wang)	125
4.3.5.1 Mugimendu ikasgarriak	125
4.3.5.2 Taldeko pokemon hoberena	126
4.3.5.3 Indarrak eta ahuleziak	128
4.3.5.4 Eboluzio katea	130
4.3.6 Itemdex (Yoel Justel)	132
5. Entregagarria (Sekuentzia diagrama eta kodea)	134
5.1 Sekuentzia diagrama	134
5.1.1 Erabiltzaileen kudeaketa (Eneko Puente)	134
5.1.1.1 Saioa Hasi	134
5.1.1.2 Erregistratu	136
5.1.1.3 Datuak Aldatu	138
5.1.1.4 (LAGUNAK) Lagunak Kargatu	140
5.1.1.5 (LAGUNAK) Utzi Jarraitzen	141
5.1.1.6 (LAGUNAK) Erabiltzailea Jarraitu	142
5.1.1.7 (LAGUNAK) Notifikazioak Ezarri/Desgaitu	143
5.1.1.8 (ADMIN) Erabiltzaileak Kargatu	144
5.1.1.9 (ADMIN) Ezabatu Erabiltzailea	145
5.1.1.10 (ADMIN) Admin Bihurtu	146
5.1.1.11 (ADMIN) Editatu	147
5.1.2 Pokemon taldeak sortu (Asier Barrio)	149
5.1.2.1 Talde berria sortu eta Pokemon sartu	149
5.1.2.2 Taldea gorde eta ezabatu	156
5.1.2.3 Taldea editatu	159
5.1.3 Changelog-a (Unai Rodriguez)	164
5.1.3.1 Notifikazioak kargatu izen baten menpe (edo ez)	164
5.1.4 Pokedex-a (Yassin Mehani)	166
5.1.5 Chatbot-a (Xinyan Wang)	168
5.1.5.1 Mugimendu ikasgarriak	168

5.1.5.2 Taldeko pokemon hoberena	170
5.1.5.4 Eboluzio katea	174
5.1.6 Itemdex (Yoel Justel)	177
5.2 Probak	180
5.2.1 Erabiltzaileen kudeaketa (Eneko Puente)	180
5.2.2 Pokemon taldeak sortu (Asier Barrio)	186
5.2.3 Changelog-a (Unai Rodriguez)	190
5.2.4 Pokedex-a (Yassin Mehani)	192
5.2.5 Chatbot-a (Xinyan Wang)	196
5.2.6 Itemdex (Yoel Justel)	200
6. Implementazioan aurkitutako arazoak eta konponbideak	204
7. Ondorioak	205

Hasiera

Sarrera

Praktikaren helburua mundu errealean aplikazioak zelan garatzen diren erakustea da. Hau aurrera eramateko hainbat fasetan banandu da proiektua. Lehenik, aplikazioak izango dituen aukerak defititu dira, alegia, erabiltzaileak zer egin ahal izango du. Ondoren, garatu dugu beharrezko informazioa gordetzeko beharko dugun datu bilketa sistema, informazio garrantzitsuena gordetzeko. Jarraitzeko, aplikazioaren kodean egingo diren deiak, eta datuak nondik lortu edo gordeko diren definitu zen. Azkenik, kodea implementatu dugu, aurreko esandakoarekin bat eginez, eta behar izatekotan aurreko zer edo zer aldatuz.

Aldaketen changelog-a

1. Bigarren entregagarriko aldaketak

Menua eta chatbot erabilpen kasuak ezabatu ditugu pantaila ez direlako erabilpen kasuak, pantaila aldaketak baizik

Dokumentazioaren titulua aldatu dugu “ERABILPEN KASUAK POKEDEX” -> “POKEDEX DOKUMENTAZIOA”

Aurrebaldintzak:

Aurrebaldintza berri batzuk gehitu ditugu

Hasi saioaren erabilpen kasuan baldintza ez funtzionala aldatu egin dugu

Gertaera Fluxua:

Gertaera batzuk aldatu ditugu, orokorrak eta pasuo batzuk falta zirelako. Gainera, gehiago sakondu dugu datuak sartzea dela pantailan errore bat agertzen denean.

Interfazea:

Interfaze batzuk gehitu ditugu falta ziren kasuetan

Interfazeetan ez dezagun zenbakizko ordenarik jarraitu, hori zuzendu dugu erabilera-kasu bakoitzari dagozkion zenbakiak jarriz eta deskripzio txiki bat jartzen azpi irudi bakoitzean

Hierarkia:

Aktoreen azalpena zuzendu ditugu admin aktorearen azalpena faltatzen zelako

Erabiltzaile edozein pertsona identifikatuta sisteman izango dela jarrita, aurrebaldintz asko kendu ditugu.

2. Laugarren entregagarriko aldaketak

Domeinuaren eredua zuzendu dugu, behar diren aldaketekin diagraman eta dokumentazioan.

3. Amaierako entregagarriko aldaketak

Domeinuaren ereduan, efikazia atributua mugimendu taulatik ezabatu da.

Datu basean Eboluzioa izeneko taula ezabatu da eta PokemonPokedex taulan preEboluzioId izeneko atributua sartu da.

PokemonTalde taulan harrapatuld gakoa ezerri da, eta hurrengo tauletako id-a aldatu da ere, PokemonTalde-ko gakoarekin bat egiteko:

PokemonTaldean, taula honetan erabiltzailelzena sartu da gako moduan, MugimendulzanTalde eta Dauka.

Changelog-ean SQL exekutatzen den query-a aldatu egin da bete behar duen funtzionalitatea betetzeko (bilatu izena)

3.3.2 Proba kendu egin da, guztia disenatzerakoan hobeto ikusi dugulako bilatzeko formularioan changelog bertan egotea. Beraz, proba kasu hori ez zeukan zentzurik

Pokemon lista eta bilaketaren komunikazio diagraman, fluxua aldatu egin da Motak, Abileziak eta Mugimenduak txertatzeko datu basean. Honetarako, API-ari dei berriak egin dira eta dagozkien SQL eta JSON-ak gehitu dira

Lagun zerrenda ataleko interfazeak aldatu da, ezabatu botoia ez du beste pantaila batera eramaten, baizik eta jarraian ezabatzen du erabiltzailea.

Erabiltzaile Kudeaketa komunikazio diagramak aldatu dira.

Erabiltzaile kudeaketako proba batzuk aldatu dira, diseinua aldatu delako.

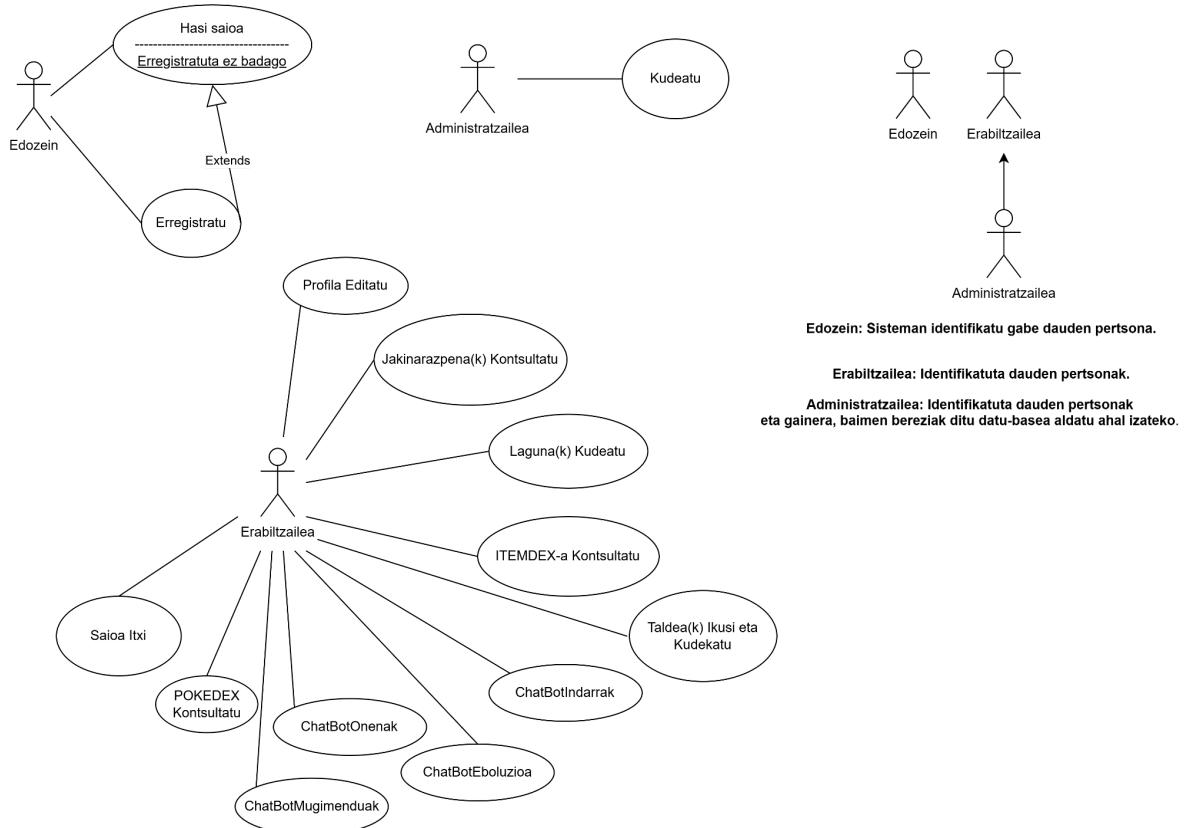
Domeinu ereduan JarraitzenDu erlazioan atributu berri bat gehitu da, notifikazioak boolearra dena.

Funtzionalitateen arduradunak

Funtzionalitatea	Arduraduna
Erabiltzaileen kudeaketa	Eneko Puente
Pokemon taldeak sortzea	Asier Barrio
ChangeLog	Unai Rodriguez
Pokemon lista eta bilaketa	Yassin Mehani
Chatbot	Xinyan Wang
ItemDex	Yoel Justel

1. Entregagarria (Erabilpen kasuak)

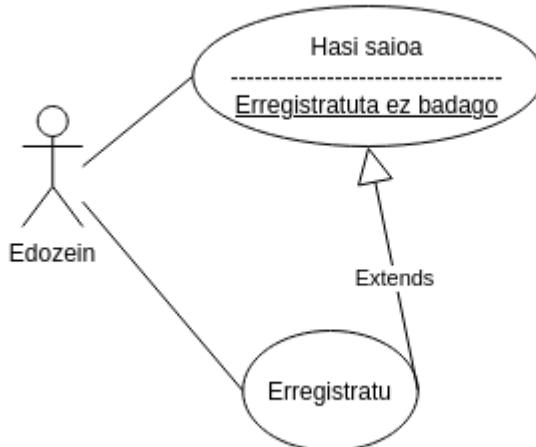
Erabilpen kasuen diagrama osoa



Irudia 1: Erabilpen kasuaren diagrama osoa

1.1 Erabiltzaileen kudeaketa

1.1.1 Hasi saioa

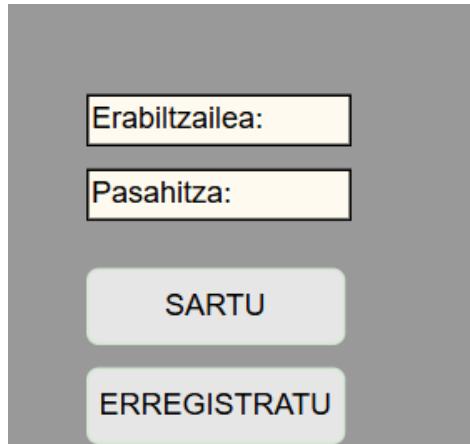


Irudia 2: Hasi saioa erabilpen kasuaren diagrama

Izena:	Hasi saioa
Deskripzioa: Edozein erabiltzaileri bere saioa hastea aukera ematen dio, programan sartu ahal izateko. Erregistratuta ez badago, ez zaio programan sartzen utziko eta erregistratu egin beharko da.	
Aktoreak: Edozein erabiltzaile	
Aurrebaldintzak: -	
Baldintza ez funtzionala: -	
Gertaeren Fluxua:	
1. [Erabiltzaileak "Saioa Hasi" botoia aukeratzen badu]. (Irudia 3) 1.1. [Daturen bat txarto badago]. 1.1.1 Erabiltzailea ez bada existitzen datu basean. 1.1.1.1 Erregistratza eskatuko zaio edo existitzen den erabiltzaile bat sartzea. 1.2.1 Erabiltzailearen pasahitza ez bada egokia 1.2.1.1 Pasahitza berriz saiatzen sartzea. 1.2. [Erabiltzailea edo posta elektronikoa ez badira existitzen] 1.2.1 Datu basean erabiltzaileari erregistratza eskatuko zaio. 1.3. [Datu guztiak existitzen badira eta ondo badaude] 1.3.1 Saioa hasiko da eta menuaren pantaila bistaratuko da.	
2. [Erabiltzailea erregistratu botoiari ematen badu] (Irudia 3) 2.1. EXTENDS erregistratu.	

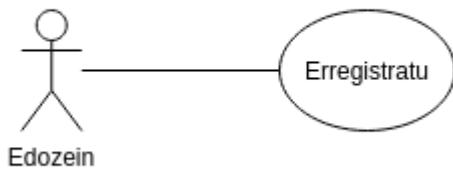
Postbaldintzak: Sartutako datuak egokiak badira erabiltzailea programara sartuko da.

Interfaze grafikoa:



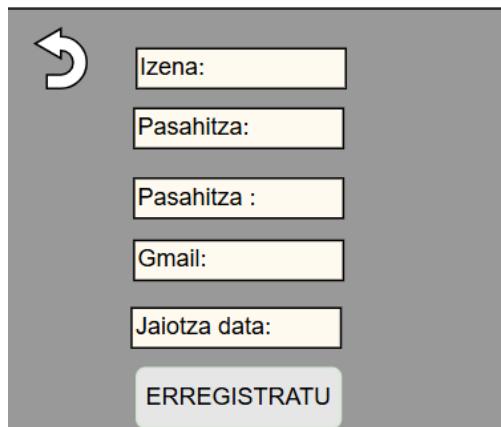
Irudia 3: Saio hasteko pantaila

1.1.2 Erregistratu



Irudia 4: Erregistratu erabilpen kasuaren diagrama

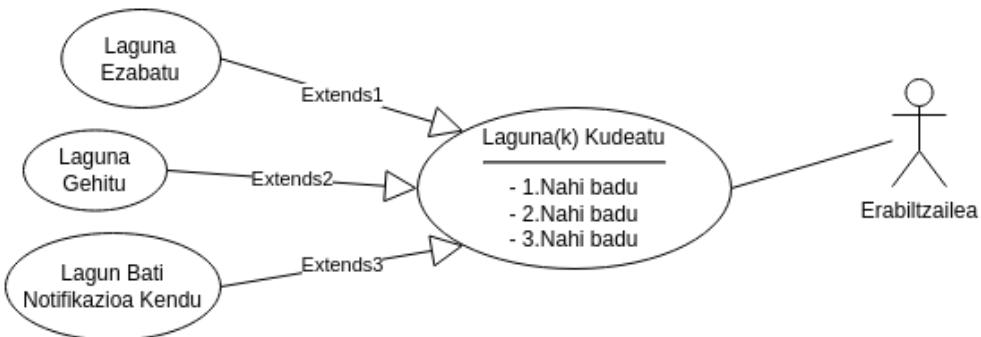
Izena:	Erregistratu
Deskripzioa: Edozein erabiltzaileari bere datuak programan erregistratzeko posibilitatea ematen du.	
Aktoreak: Edozeinek	
Aurrebaldintzak: –	
Baldintza ez funtzionala: Sistemak konprobatuko du ea sartutako datuak egokiak diren ala ez, hau da, pasahitza $!-\$^{\wedge};\#~=@$ karaktereak ez izatea	
Gertaeren Fluxua:	
1. [“Erregistratu” botoia sakatzen badu] (Irudia 5) 1.1. [Izena elektronikoa dagoeneko kontu batekoa bada] 1.1.1 Mezu bat agertuko zaio erabiltzaileari erabiltzailea existitzen dela esanez. 1.2. [Izena elektronikoa ez bada existitzen] (Irudia 5) 1.2.1. Kontua sortuko da.	
Postbaldintzak: Sartutako datuak egokiak badira erabiltzailea programan erregistratuta geratuko da eta hasi saioa-ren pantailara joango da.	
Interfaze grafikoa:	



A registration form interface featuring a grey header bar with white text and input fields. The header includes a back arrow icon and the text "Izena:", "Pasahitza:", "Pasahitza:", "Gmail:", and "Jaiotza data:". Below the header is a large blue button labeled "ERREGISTRATU".

Irudia 5: Erregistratzeko pantaila

1.1.3 Lagun Zerrenda



Irudia 6: Lagun zerrenda erabilpen kasuaren diagrama

Izena: Lagun zerrenda

Deskripzioa: Menuaren azpimenu honetan erabiltzaileak hainbat erabilpen kasu izango ditu bere lagun zerrendaren kudeaketa egiteko

Aktoreak: Erabiltzailea

Aurrebaldintzak: –

Baldintza ez funtzionala: –

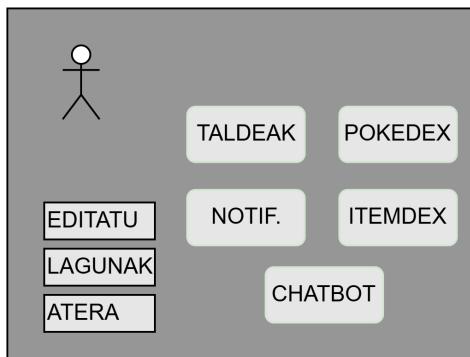
Gertaeren Fluxua:

1. [Erabiltzailea menuan lagunak kudeatu botoiari ematen badu]
 - 1.1 Bere lagun zerrenda kudeatzeko aukerak agertuko zaizkio. (Irudia 7)
2. [Lagunak kudeatu azpimenuaren barruan]
 - 2.1 [Erabiltzaileak gehitu botoiari ematen badu]
 - 2.1.1 EXTENDS Laguna gehitu.
 - 2.2 [Erabiltzaileak ezabatu botoiari ematen badu]
 - 2.2.1 EXTENDS Laguna ezabatu.
 - 2.3 [Erabiltzaileak lagun bati notifikazioak kentzen badizkio]
 - 2.3.1 EXTENDS Lagun bati notifikazioak kendu.
3. [Administratzaileak "Home" botoian klik egiten badu]
 - 3.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 8)

Postbaldintzak: Erabiltzaileak bere lagun zerrenda kudeatuko du.

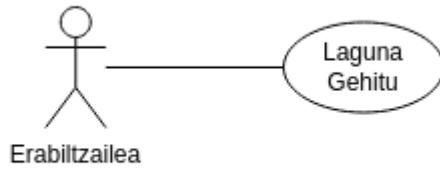
Interfaze grafikoa:

	Pepe
	Juan
	Alonso
	Ignacio
GEHITU	



Irudia 7: Lagun zerrendaren pantaila / Irudia 8: Menu nagusiaren pantaila

1.1.3.1 Laguna Gehitu

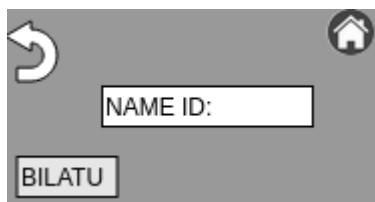


Irudia 9: Laguna gehitu erabilpen kasuaren diagrama

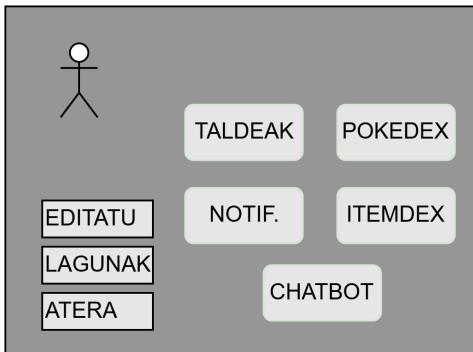
Izena:	Laguna gehitu
Deskripzioa: Erabiltzaile bat beste erabiltzaile baten ID-a erabiliz bere lagun zerrenda gehitzeko erabilpen kasua.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: Lagun zerrenda kudeaketaren pantailan egon.	
Baldintza ez funtzionala: –	
Gertaeren Fluxua:	
1. [Erabiltzailea Gehitu botoia sakatzen badu] (Irudia 10) 1.1 Beste pantaila bat agertuko da non gehitu nahi duzun erabiltzailearen. ID-a sar beharko du (Irudia 11) 1.2 [Sistemak aurkitzen badu erabiltzailea sartutako ID-arekin] 1.2.1 Erabiltzaile hori lagun zerrendara gehituko da. 1.3 [Sistemak aurkitzen ez badu erabiltzailea sartutako ID-arekin] 1.3.1 Sistemak ID-a berriro sartzeko eskatuko dio erabiltzaileari.	
2. [Administratzaleak "Home" botoian klik egiten badu] 2.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 12)	
Postbaldintzak: Erabiltzaileak lagun berri bat gehitu du bere zerrendan.	
Interfaze grafikoa:	



Irudia 10: Lagun zerrendaren pantaila

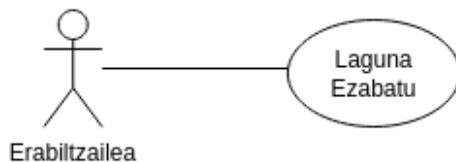


Irudia 11: Erabiltzailea bilatzeko pantaila



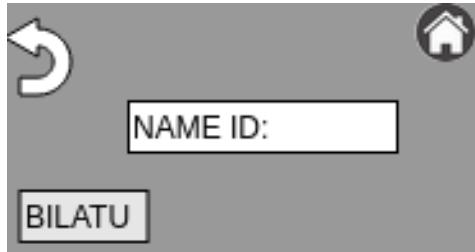
Irudia 12: Menu nagusiaren pantaila

1.1.3.2 Laguna Ezabatu

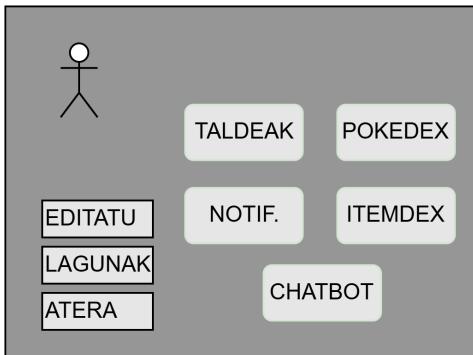


Irudia 13: Laguna ezabatzeko erabilpen kasuaren diagrama

Izena:	Laguna Ezabatu
Deskripzioa: Erabiltzaile bat beste erabiltzaile baten ID-a erabiliz bere lagun zerrenda ezabatzeko erabilpen kasua.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: Lagun zerrenda kudeaketaren pantailan egon.	
Baldintza ez funtzionala: –	
Gertaeraren Fluxua:	
1. [Erabiltzailea Ezabatu botoia sakatzen badu] (Irudia 14) 1.1 Horren ondoan dagoen erabiltzailea lagunen zerredatik kenduko da.	
2. [Administratzaleak "Home" botoian klik egiten badu] 2.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 16)	
Postbaldintzak: Erabiltzaileak lagun bat ezabatu du bere zerrendan.	
Interfaze grafikoa:	
<p>The screenshot shows a user interface for managing contacts. At the top, there are icons for back, forward, and home. Below that is a list of four contacts: Pepe, Juan, Alonso, and Ignacio. Each contact entry consists of a small profile picture, the name, and a rectangular input field. To the right of each name is a small bell icon with a slash over it. At the bottom left is a large 'GEHITU' (Delete) button.</p>	
Irudia 14: Lagun zerrendaren pantaila	

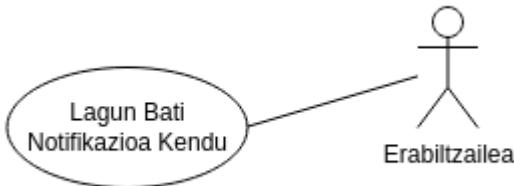


Irudia 15: Erabiltzailea bilatzeko pantaila



Irudia 16: Menu nagusiaren pantaila

1.1.3.3 Lagun baten notifikazioak kendu



Irudia 17: Lagun bati notifikazioak kendu erabilpen kasuaren diagrama

Izena: Lagun baten notifikazioak isilarazi

Deskripzioa: Erabiltzaile bat beste erabiltzaile baten notifikazioak ikusi nahi ez duenean egiten den erabilpen kasua da.

Aktoreak: Erabiltzailea

Aurrebaldintzak: Lagun zerrenda kudeaketaren pantailan egon.

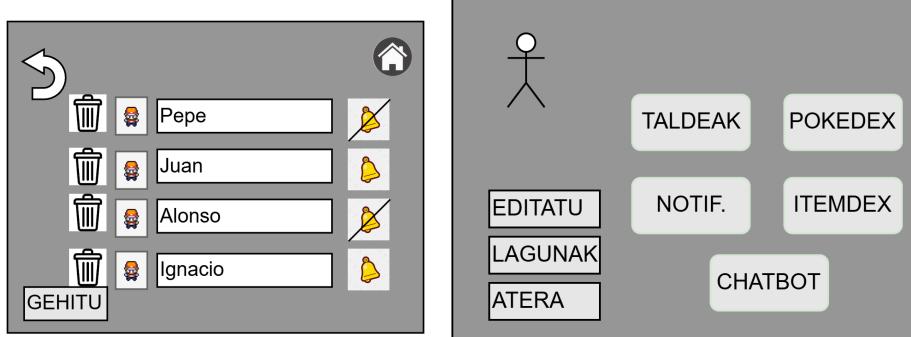
Baldintza ez funtzionala: –

Certaeren Fluxua:

1. [Erabiltzailea isalarazi botoia] (Irudia 18)
1.1 Lagun horren notifikazioak isalariko dira eta ez dira gehiago agertuko changelog-ean.
2. [Administratzaleak "Home" botoian klik egiten badu]
2.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 19)

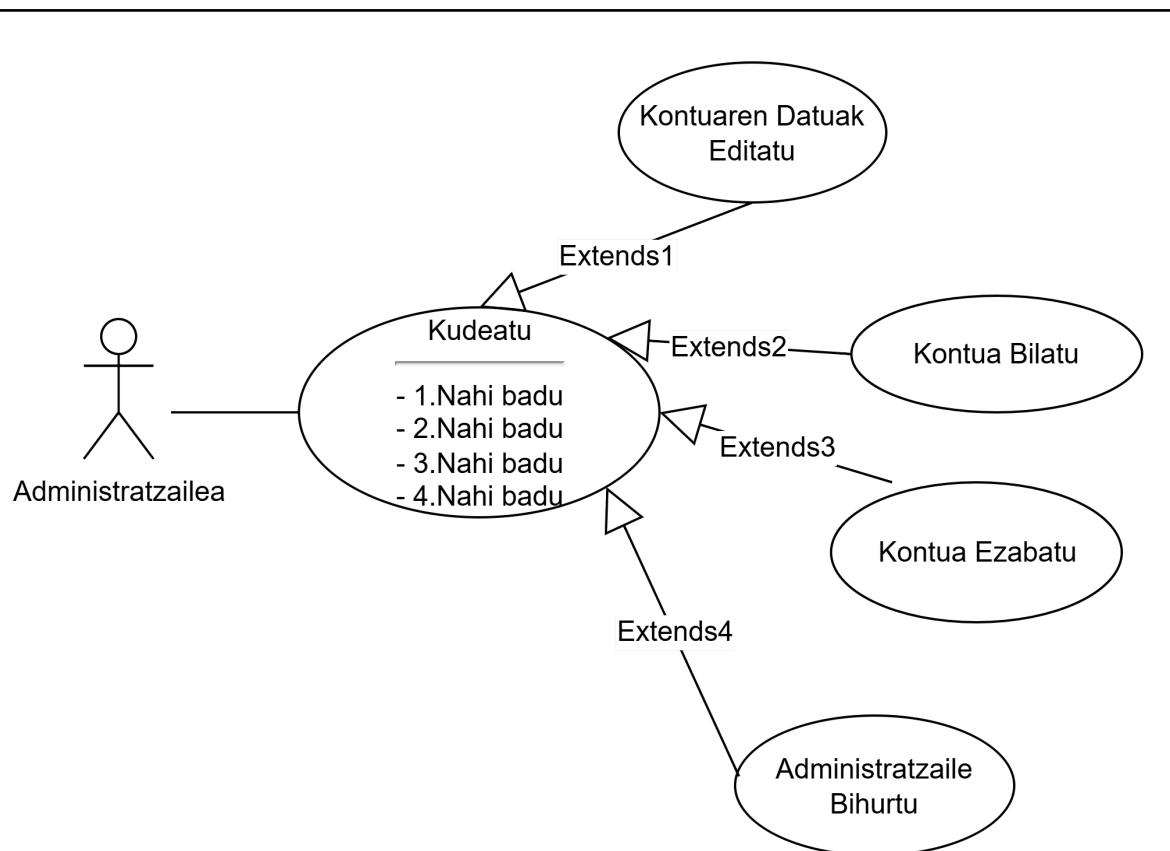
Postbaldintzak: Erabiltzaileak lagun baten notifikazioak isalarazi ditu changelog-ean gehiago ez agertzeko.

Interfaze grafikoa:



Irudia 18: Lagun zerrendaren pantaila / Irudia 19: Menu nagusiaren pantaila

1.1.4 Menu Admin



Irudia 20: Menu admin-aren erabilpen kasuaren diagrama

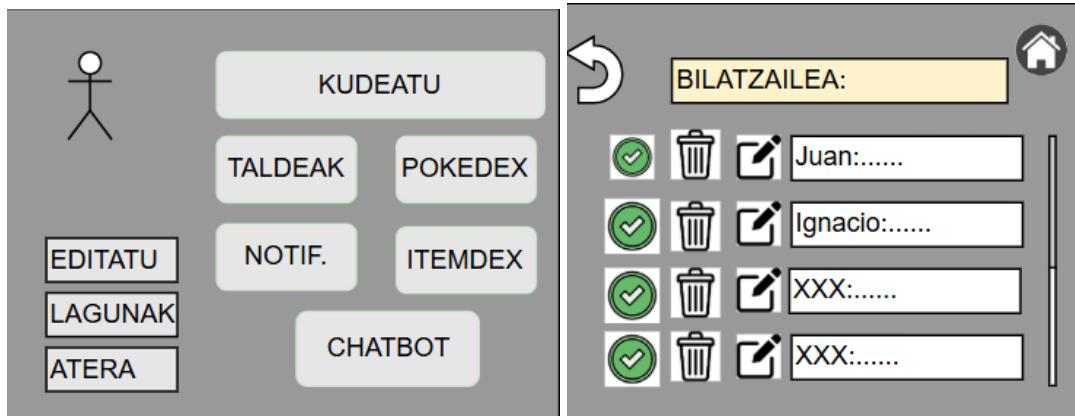
Izena:	Menu Administratzaile
Deskripzioa: Administratzaileak erabiltzailearen menu bera izango du, baina honek aukera gehigarri bat izango du beste kontuen kudeaketa egiteko.	
Aktoreak:	Administratzailea
Aurrebaldintzak:	-
Baldintza ez funtzionala:	-
Gertaeren Fluxua:	<ol style="list-style-type: none"> 1. Administratzaileak menu nagusian dagoen "Kudeatu" botoia sakatzen du. (Irudia 21) 2. Sistemak kontu guztien zerrenda erakutsiko du. (Irudia 22) 3. [Erabiltzaileak "Editatu" botoian klik egiten badu] <ol style="list-style-type: none"> 3.1. EXTENDS Kontuaren Datuak Editatu. 4. [Erabiltzaileak "Zakarrontzi" botoian klik egiten badu] <ol style="list-style-type: none"> 4.1. EXTENDS Kontua Ezabatu. 5. [Erabiltzaileak "Egoki" botoian klik egiten badu] <ol style="list-style-type: none"> 5.1. EXTENDS Administratzaile Bihurtu

6. [Erabiltzaileak "Bilatu" botoian klik egiten badu]

6.1. EXTENDS Kontua Bilatu

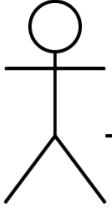
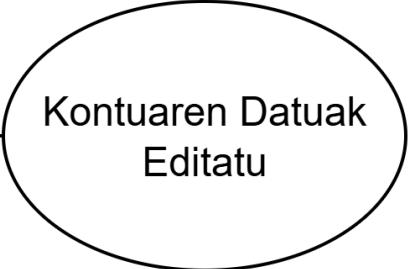
Postbaldintzak: Erabiltzaileak sakatutako botoaren arabera pantaila ezberdin bat agertuko zaio.

Interfaze grafikoa:



Irudia 21: Administratzailearen menu / Irudia 22: Erabiltzaileak kudeatzeko nagusiaren pantaila

1.1.4.1 Erabiltzailearen Datuak Editatu

	 <p>Kontuaren Datuak Editatu</p>
Administratzailea	
<p>Irudia 23: Erabiltzailearen datuak editatu erabilpen kasuaren diagrama</p>	
<p>Izena: Erabiltzailearen Datuak Editatu</p>	
<p>Deskripzioa: Administratzaileak erabiltzailearen datuak editatu ahalko ditu.</p>	
<p>Aktoreak: Administratzailea</p>	
<p>Aurrebaldintzak: –</p>	
<p>Baldintza ez funtzionala: –</p>	
<p>Gertaeren Fluxua:</p> <ol style="list-style-type: none">1. Administratzaileak datu berriak sartuko ditu, aldatu ez diren datuak ere sartu beharko ditu eta gorde botoiari emango dio. (Irudia 25)<ol style="list-style-type: none">1.1 Datuak zuzenak badira, hau da, izena datu basean ez badago edo pasahitzak karaktere berizirik ez baditu datuak gordeko dira.1.2 Datuak zuzenak ez badira, izena datu basean badago eta pasahitzak karaktere berizirik badu, orduan mezu bat agertuko da, datu ez-egokiak direla esanez.2. [Administratzaileak “Home” edo “Itzuli” botoian klik egiten badu]<ol style="list-style-type: none">2.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 26)	
<p>Postbaldintzak: Erabiltzailaren datuak aldatuko dira.</p>	
<p>Interfaze grafikoa:</p>	

The screenshot shows a list of users under the heading "BILATZAILEA:". Each user entry includes a checkmark icon, a trash can icon, and a edit icon. The entries are:

- Juan:.....
- Ignacio:.....
- XXX:.....
- XXX:.....

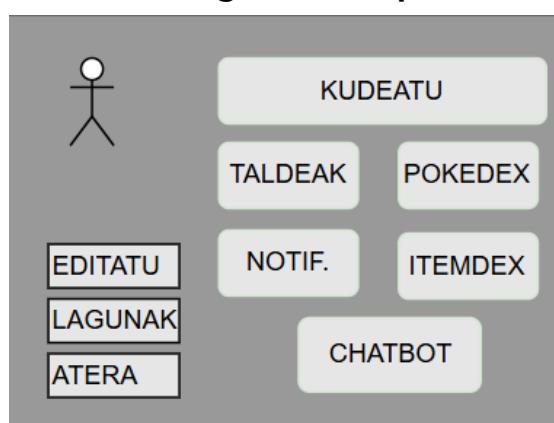
Irudia 24: Erabiltzaileak kudeatzeko pantaila

The screenshot shows a registration form with the following fields:

- Izena: [Text input]
- Pasahitza: [Text input]
- Pasahitza : [Text input]
- Gmail: [Text input]
- Jaiotza data: [Text input]

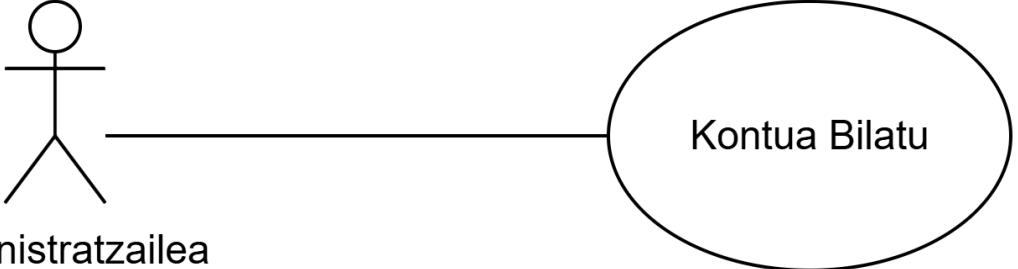
At the bottom is a large "GORDE" button.

Irudia 25: Erregistrozko pantaila



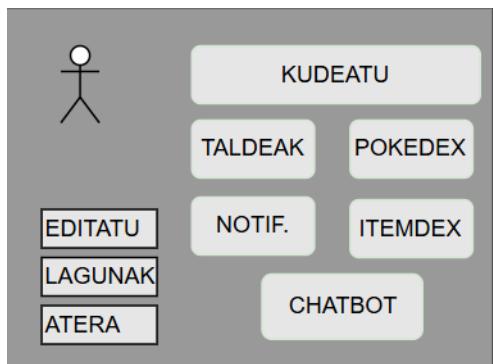
Irudia 26: Administratzailearen menu nagusiaren pantaila

1.1.4.2 Kontua Bilatu

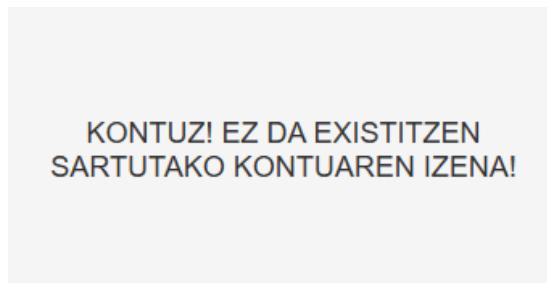

Administratzailea
Irudia 27: Erabiltzailearen kontua bilatzeko erabilpen kasuaren diagrama
Izena: Kontua Bilatu
Deskripzioa: Administratzaileak kontuen zerrendan, kontuak bilatzeko aukera izango du kontuen izena idatziz.
Aktoreak: Administratzailea
Aurrebaldintzak: –
Baldintza ez funtzionala: –
Gertaeren Fluxua:
<ol style="list-style-type: none">1. Administratzaileak, kontu bat bilatu dezake bere izena jarri. (Irudia 29)<ol style="list-style-type: none">1.1 [Kontua existitzen bada]<ol style="list-style-type: none">1.1.1 Pantailan erakutsiko du.1.2 [Itema ez bada existitzen]<ol style="list-style-type: none">1.2.1 Errore mezu bat agertuko da ez dela existitzen esanik. (Irudia 30)2. [Administratzaileak "Itzuli" botoian klik egiten badu]<ol style="list-style-type: none">2.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 29)3. [Administratzaileak "Home" botoian klik egiten badu]<ol style="list-style-type: none">3.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 29)
Postbaldintzak: Bilaketaren emaitzak agertuko dira.
Interfaze grafikoa:



Irudia 28: Erabiltzaileak kudeatzeko pantaila

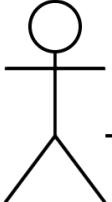


Irudia 29: Administratzailearen menu nagusiaren pantaila



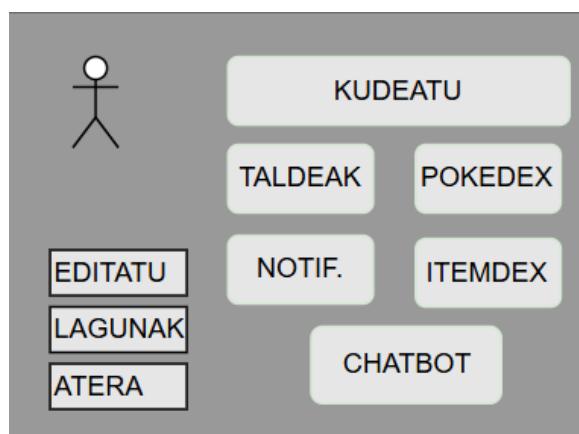
Irudia 30: Kontua ez dela existzen errore mezua

1.1.4.3 Kontua Ezabatu

	 Kontua Ezabatu
Administratzailea	
Irudia 31: Erabiltzailearen kontua ezabatzeko erabilpen kasuaren diagrama	
Izena: Kontua Ezabatu	
Deskripzioa: Administratzaileak erabiltzailearen baten kontua ezabatzeko aukera izango du.	
Aktoreak: Administratzailea	
Aurrebaldintzak: –	
Baldintza ez funtzionala: –	
Gertaeren Fluxua:	
1. Administratzaileak, kontu bat ezabatu dezake. 2. [Administratzaileak "Ezabatu" botoian klik egiten badu] 2.1 Kontu hori sistematik ezabatuko da. (Irudia 32) 3. [Administratzaileak "Itzuli" botoian klik egiten badu] 3.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 33) 4. [Administratzaileak "Home" botoian klik egiten badu] 4.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 33)	
Postbaldintzak: Kontua ezabatzea	
Interfaze grafikoa:	

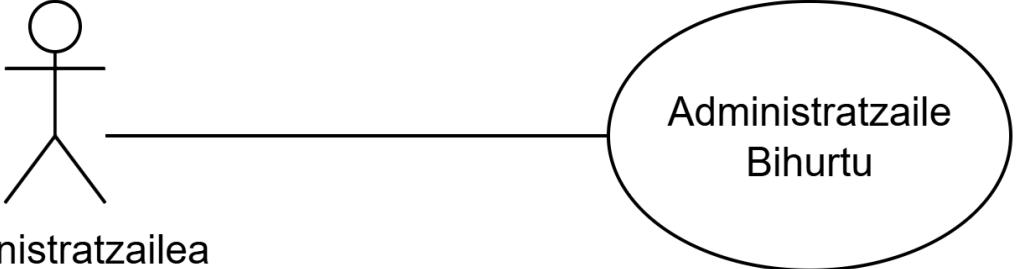


Irudia 32: Erabiltzaileak kudeatzeko pantaila



Irudia 33: Administratzailearen menu nagusia pantaila

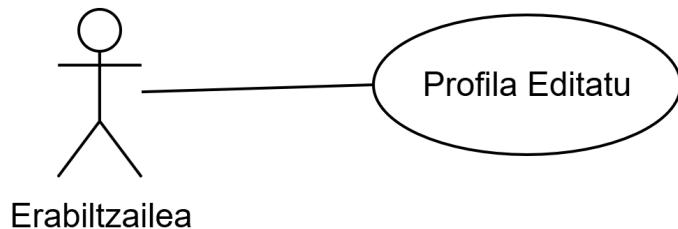
1.1.4.4 Administratzaile Bihurtu


Administratzailea
Irudia 34: Erabiltzaile bati administratzaile bihurtzeko erabilpen kasuaren diagrama
Izena: Administratzaile Bihurtu
Deskripzioa: Administratzaileak beste erabiltze batzuk administratzaile bihurtu ditzake.
Aktoreak: Administratzailea
Aurrebaldintzak: –
Baldintza ez funtzionala: –
Gertaeren Fluxua:
<ol style="list-style-type: none">1. [Administratzaileak "Egoki" botoian klik egiten badu]<ol style="list-style-type: none">1.1 Sistemak erabiltzaile hori, administratzaile bilakatuko du.2. [Administratzaileak "Itzuli" botoian klik egiten badu]<ol style="list-style-type: none">2.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 35)3. [Administratzaileak "Home" botoian klik egiten badu]<ol style="list-style-type: none">3.1 Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 35)
Postbaldintzak: Erabiltzailea administratzaile bilakatuko da.
Interfaze grafikoa:



Irudia 35: Erabiltzaileak kudeatzeko pantaila

1.1.5 Profila egunerau



Irudia 36: Erabiltzailea bere profila editatzeko erabilpen kasuaren diagrama

Izena:	Profila editatu
Deskripzioa: Erabiltzaileak bere datu pertsonalak editatu ahal izango du. Modu berean, administratzaileak edozein erabiltzailearen datuak modifika ditzake.	
Aktoreak: Erabiltzailea eta administratzailea	
Aurrebaldintzak: Datuen edizio pantailan egotea.	
Baldintza ez funtzionala: –	
Gertaeren Fluxua:	
1. Erabiltzailea bere datuak sartuko ditu, aldatu ez diren datuak ere sartu beharko ditu eta gorde botoiari emango dio. (Irudia 37) 1.1. Datuak zuzenak badira, hau da, izena ez badago lotuta beste kontu bati edo pasahitza karaktere berezirik ez baditu, orduan datuak gordeko dira. 1.2. Datuak zuzenak ez badira, hau da, izena lotuta badago beste kontu bati edo pasahitza karaktere berezirik baditu, orduan orduan mezu bat agertuko da, datu ez-egokiak direla esanez. 2. [Erabiltzaileak “Home” edo “Itzuli” botoian klik egiten badu] 2.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 38)	
Postbaldintzak: Sartutako datuak egokiak badira, aurreko datuak ordezkatuko dira berriekin eta menuko pantailara joango da.	
Interfaze grafikoa:	

Izena:

Pasahitza:

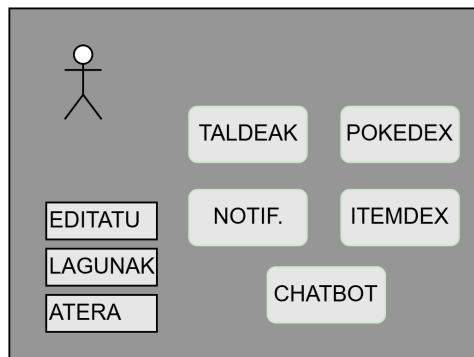
Pasahitza :

Gmail:

Jaiotza data:

GORDE

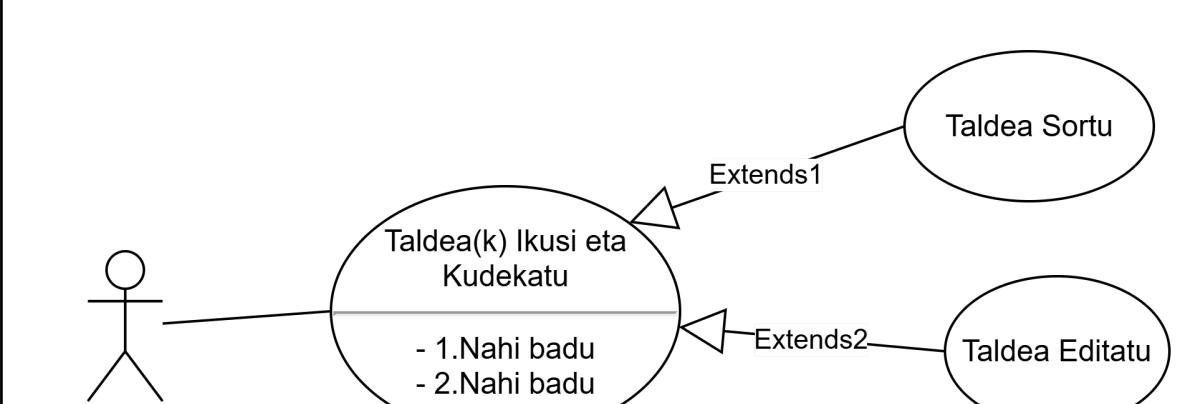
Irudia 37: Datuak editatzeko pantaila

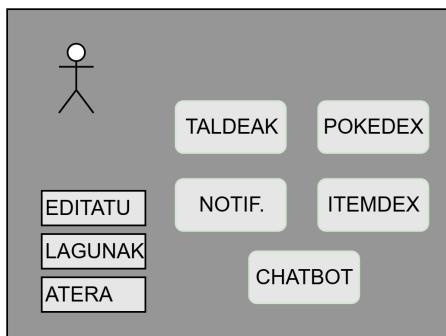


Irudia 38: Menu nagusiaren pantaila

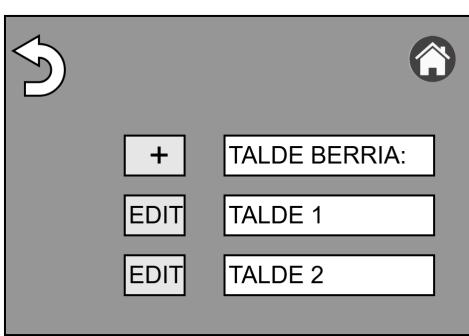
1.2. Pokemon taldeak sortzea

1.2.1 Talde menua

	
Irudia 39: Taldeak ikusi eta kudeatzeko erabilpen kasuaren diagrama	
Izena:	Pokemon taldeen kudeaketa
Deskripzioa: Erabiltzailea dituen Pokemon taldeak ikusatzeko erabili ahal izango duen funtzioa.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: Sistemarekin identifikatuta egotea.	
Baldintza ez funtzionala: –	
Gertaerent Fluxua: <ol style="list-style-type: none">1. Erabiltzaileak menu nagusian dagoen "TALDEAK" botoia sakatzen du. (Irudia 40)2. Sistemak erabiltzaileak dituen Pokemon taldeen zerrenda erakutsiko du, baldin eta talderik bad(it)u. (Irudia 41)3. [Erabiltzaileak "+" botoian klik egiten badu]<ol style="list-style-type: none">3.1 EXTENDS Taldea Sortu.4. [Erabiltzaileak "EDIT" botoian klik egiten badu (Edozein taldetan)]<ol style="list-style-type: none">4.1 EXTENDS Taldea Editatu.	
Postbaldintzak: –	
Interfaze grafikoa:	

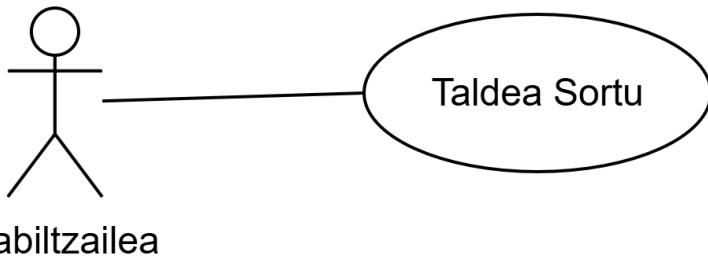


**Irudia 40: Menu nagusiaren
pantaila**



**Irudia 41: Pokemon taldeak kudeatzeko
pantaila**

1.2.1.1 Talde sorkuntza



Irudia 42: Pokemon taldea sortzeko erabilpen kasuaren diagrama

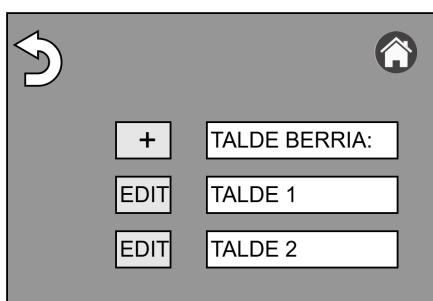
Izena:	Talde sorkuntza
Deskripzioa: Erabiltzaile bat Pokemon taldeak sortzeko erabilpen kasua.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: Pokemon taldeen kudeaketa pantailan egotea.	
Baldintza ez funtzionala: –	
Gertaeren Fluxua:	
<ol style="list-style-type: none">1. Erabiltzaileak Pokemon taldeen kudeaketan menuan dagoen "+" botoia sakatzen du. (Irudia 43)2. Sistemak hutsik dagoen talde berri bat erakusten du. (Irudia 44)3. [Erabiltzaileak "+" botoian klik egiten badu]<ol style="list-style-type: none">3.1. Sistema pantaila Pokedex-a erakutsiko du. (Irudia 46 eta Irudia 47)<ol style="list-style-type: none">3.1.1. Erabiltzaileak pokemon baten izena idazten du edo listatik Sistema pantaila Pokedex-a erakutsiko du. (Irudia 46 eta Irudia 47)<ol style="list-style-type: none">3.1.1.1. [Pokemon-a existitzen bada]3.1.1.1.1. Pokemon-a agertuko da.<ol style="list-style-type: none">3.1.1.1.1.1. [Erabiltzaileak Pokemon-a klikatzean]3.1.1.1.1.2. Sistema pantaila berri bat erakutsiko du, hautatutako Pokemon-aren deskribapen laburra erakutsiz. (Irudia 48)3.1.1.1.3. [Erabiltzaileak "Aukeratu" botoian klik egiten badu]3.1.1.1.3.1. Pokemon-a taldean sartuko da eta Taldeko pantailara bueltatuko da. (Irudia 45)3.1.1.1.4. [Erabiltzaileak "Home" botoian klik	

- egiten badu]
- 3.1.1.1.4.1. Sistemak menu nagusiaren pantailara bueltatzen da.
- 3.1.1.1.5. [Erabiltzaileak "Itzuli" botoian klik egiten badu]
- 3.1.1.1.5.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 46 eta Irudia 47)
- 3.1.1.1.2. [Pokemon-a ez bada existitzen]
- 3.1.1.1.2.1. Pokemon lista osoa agertuko da.
- 3.1.1.1.2.2. [Erabiltzailea "Itzuli" botoian klik egiten badu]
- 3.1.1.1.2.2.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 46 eta Irudia 47)
- 3.1.1.1.2.3. [Erabiltzailea "Home" botoian klik egiten badu]
- 3.1.1.1.2.3.1. Menu nagusira bueltatuko da.
- 3.1.1.1.3. [Hutsik utzi badu bilatzailea]
- 3.1.1.1.3.1. Pokemon lista osoa erakutsiko zaio.
- 3.1.1.1.4. [Hitz partziala sartu badu]
- 3.1.1.1.4.1. Hitz partzial horrekin bat egiten duten Pokemon guztiak agertuko dira zerrendan.
4. [Erabiltzaileak "Home" botoian klik egiten badu]
- 4.1. Sistemak menu nagusiaren pantailara bueltatzen da.
5. [Erabiltzaileak "Ezabatu" botoian klik egiten badu]
- 5.1. Talde osoa ezabatuko da eta Pokemon taldearen kudeaketa pantailara bueltatuko da.
6. [Erabiltzaileak "Itzuli" edo "Gorde" botoian klik egiten badu]
- 6.1. Sistemak Pokemon taldearen kudeaketa menura bueltatzen da.

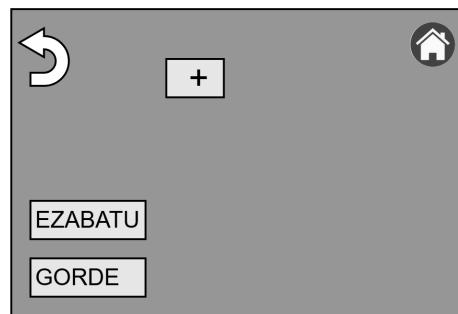
Postbaldintzak:

Taldea ondo sortu bada, pantailan talde berria agertuko da.

Interfaze grafikoa:



Irudia 43: Pokemon taldeak kudeatzkeo pantaila



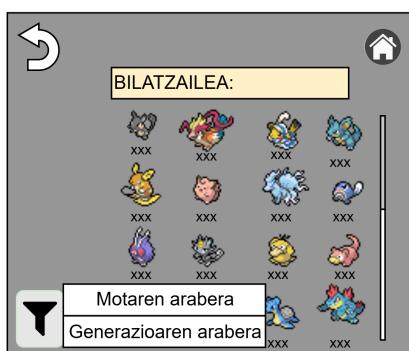
Irudia 44: Pokemon taldearen pantaila (Hutsik)



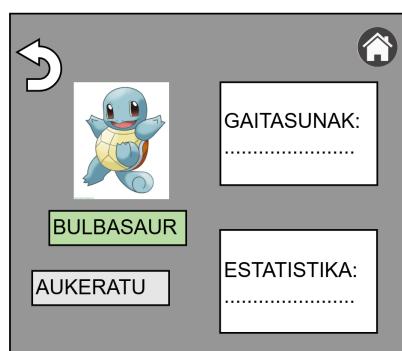
Irudia 45: Pokemon taldearen pantaila (Ez hutsik)



Irudia 46: Pokedex pantaila

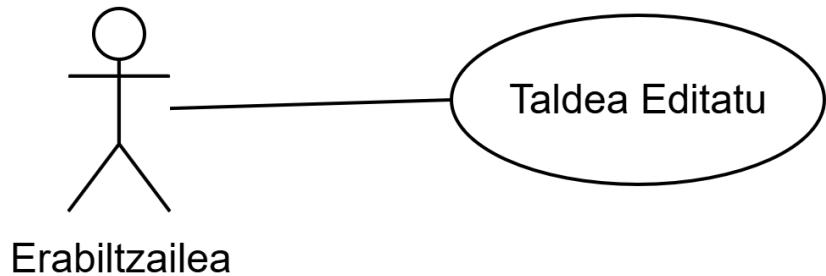


Irudia 47: Pokedex pantailaren iragazketak



Irudia 48: Pokemon baten gaitasunen pantaila

1.2.1.2 Talde edizioa



Irudia 49: Pokemon taldea editatzeko erabilpen kasuaren diagrama

Izena:	Talde edizioa
<p>Deskripzioa: Erabiltzaile bat aurretik edukitako Pokemon taldeak editatzeko erabilpen kasua.</p>	
<p>Aktoreak: Erabiltzailea</p>	
<p>Aurrebaldintzak: Pokemon taldeen kudeaketa pantailan egotea eta Pokemon talde bat, gutxienez, izatea.</p>	
<p>Baldintza ez funtzionala: –</p>	
<p>Gertaerent Fluxua:</p> <ol style="list-style-type: none">1. Erabiltzaileak Pokemon taldeen kudeaketan menuan dagoen "EDIT" botoia sakatzen du. (Irudia 50)2. Sistemak aukeratutako Pokemon taldea erakusten du. (Irudia 51)3. [Erabiltzaileak "+" botoian klik egiten badu]<ol style="list-style-type: none">3.1. Sistema pantaila Pokedex-a erakutsiko du. (Irudia 52 eta Irudia 53)<ol style="list-style-type: none">3.1.1. Erabiltzaileak pokemon baten izena idazten du edo listatik Sistema pantaila Pokedex-a erakutsiko du. (Irudia 52 eta Irudia 53)3.1.1.1. [Pokemon-a existitzen bada]<ol style="list-style-type: none">3.1.1.1.1. Pokemon-a agertuko da.<ol style="list-style-type: none">3.1.1.1.1.1. [Erabiltzaileak Pokemon-a klikatzean]3.1.1.1.1.2. Sistema pantaila berri bat erakutsiko du, hautatutako Pokemon-aren deskribapen laburra erakutsiz. (Irudia 54)3.1.1.1.3. [Erabiltzaileak "Aukeratu" botoian klik egiten badu]<ol style="list-style-type: none">3.1.1.1.3.1. Pokemon-a taldean sartuko da eta Taldeko pantailara bueltatuko da.3.1.1.1.4. [Erabiltzaileak "Home" botoian klik	

egiten badu]

3.1.1.1.4.1. Sistemak menu nagusiaren pantailara bueltatzen da.

3.1.1.1.5. [Erabiltzaileak "Itzuli" botoian klik egiten badu]

3.1.1.1.5.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 52 eta Irudia 53)

3.1.1.1.2. [Pokemon-a ez bada existitzen]

3.1.1.1.2.1. Pokemon lista osoa agertuko da.

3.1.1.1.2.2. [Erabiltzailea "Itzuli" botoian klik egiten badu]

3.1.1.1.2.2.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 52 eta Irudia 53)

3.1.1.1.2.3. [Erabiltzailea "Home" botoian klik egiten badu]

3.1.1.1.2.3.1. Menu nagusira bueltatuko da.

3.1.1.1.3. [Hutsik utzi badu bilatzailea]

3.1.1.1.3.1. Pokemon lista osoa erakutsiko zai

3.1.1.1.4. [Hitz partziala sartu badu]

3.1.1.1.4.1. Hitz partzial horrekin bat egiten duten Pokemon guztiak agertuko dira zerrendan.

4. [Erabiltzaileak Pokemon baten argazkian klik egiten badu]

4.1. Pantaila berri bat agertuko da. (Irudia 55)

4.2. [Erabiltzaileak "Ezabatu" botoian klik egiten badu]

4.2.1. Pokemon-a ezabatuko da taldetik eta taldeko pantailara bueltatuko da.

4.3. [Erabiltzaileak "Aldatu" botoian klik egiten badu]

4.3.1. Sistema pantaila Pokedex-a erakutsiko du. (Irudia 52 eta Irudia 53)

4.3.1.1. Erabiltzaileak pokemon baten izena idazten du edo listatik hautatzen du. (Idaztean zenbakiak eta karaktere bereziak onartuta daude)

4.3.1.1.1. [Pokemon-a existitzen bada]

4.3.1.1.1.1. Pokemon-a agertuko da.

4.3.1.1.1.1.1. [Erabiltzaileak Pokemon-a

- klikatzean]
- 4.3.1.1.1.2. Sistema pantaila berri bat erakutsiko du, hautatutako Pokemon-aren deskribapen laburra erakutsiz. (Irudia 54)
- 4.3.1.1.1.3. [Erabiltzaileak "Aldatu" botoian klik egiten badu]
- 4.3.1.1.1.3.1. Pokemon-a taldean sartuko da, aurrekoan ordezkatuz eta Taldeko pantailara bueltatuko da.
- 4.3.1.1.1.4. [Erabiltzaileak "Home" botoian klik egiten badu]
- 4.3.1.1.1.4.1. Sistemak menu nagusiaren pantailara bueltatzen da.
- 4.3.1.1.1.5. [Erabiltzaileak "Itzuli" botoian klik egiten badu]
- 4.3.1.1.1.5.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 52 eta Irudia 53)
- 4.3.1.1.2. [Pokemon-a ez bada existitzen]
- 4.3.1.1.2.1. Pokemon lista osoa agertuko da.
- 4.3.1.1.2.2. [Erabiltzailea "Itzuli" botoian klik egiten badu]
- 4.3.1.1.2.2.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 52 eta Irudia 53)
- 4.3.1.1.2.3. [Erabiltzailea "Home" botoian klik egiten badu]
- 4.3.1.1.2.3.1. Menu nagusira bueltatuko da.
- 4.3.1.1.3. [Hutsik utzi badu bilatzailea]
- 4.3.1.1.3.1. Pokemon lista osoa erakutsiko zaio
- 4.3.1.1.4. [Hitz partziala sartu badu]
- 4.3.1.1.4.1. Hitz partzial horrekin bat egiten duten Pokemon guztiak agertuko dira zerrendan.
- 4.3.2. [Erabiltzaileak "Home" botoian klik egiten badu]
- 4.3.2.1. Sistemak menu nagusiaren pantailara bueltatzen da.

4.3.3. [Erabiltzaileak "Itzuli" botoian klik egiten badu]

4.3.3.1. Pokemon talde pantaila erakutsiko da. (Irudia 51)

4.4. [Erabiltzaileak "Home" botoian klik egiten badu]

4.4.1. Sistemak menu nagusiaren pantailara bueltatzen da.

5. [Erabiltzaileak "Ezabatu" botoian klik egiten badu]

5.1. Talde osoa ezabatuko da eta Pokemon taldearen kudeaketa pantailara bueltatuko da. (Irudia 50)

6. [Erabiltzaileak "Home" botoian klik egiten badu]

6.1. Sistemak menu nagusiaren pantailara bueltatzen da.

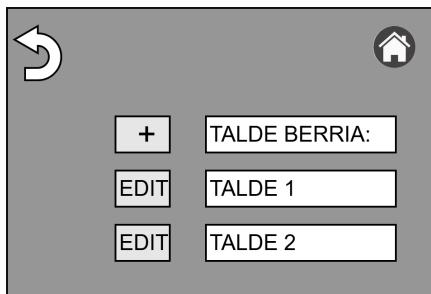
7. [Erabiltzaileak "Itzuli" edo "Gorde" botoian klik egiten badu]

7.1. Sistemak Pokemon taldearen kudeaketa menura bueltatzen da. (Irudia 50)

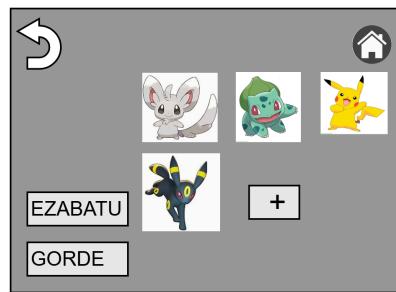
Postbaldintzak:

Taldea ondo gorde edo ezabatu bada, taldeko aldaketak gordeko dira.

Interfaze grafikoa:



**Irudia 50: Pokemon taldeak
kudeatzeko pantaila**



Irudia 51: Pokemon taldearen pantaila

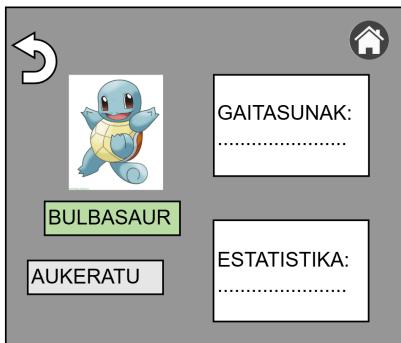


Irudia 52: Pokedex pantaila



Irudia 53: Pokedex pantailaren

iragazketak



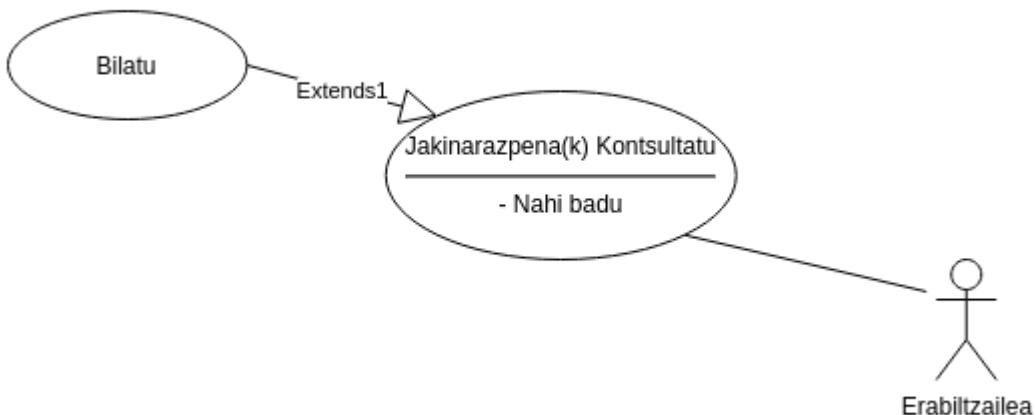
**Irudia 54: Pokemon baten
gaitasunen pantaila**



**Irudia 55: Pokemon bat taldetik aldatzeko
edo ezabatzeko pantaila**

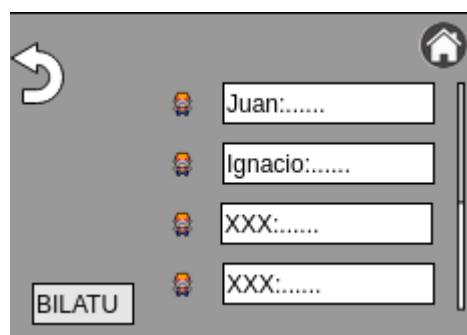
1.3. Changelog

1.3.1 Jakinarazpen menua



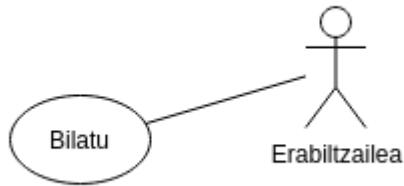
Irudia 56: Jakinarazpenak konsultatzeko erabilpen kasuaren diagrama

Izena:	Jakinarazpen kudeaketa
Deskripzioa:	Erabiltzaileari iristen zaizkion jakinarazpenak kudeatzeko eta ikusteko aukera
Aktoreak:	Erabiltzailea
Aurrebaldintzak:	Sistemarekin identifikatuta egotea.
Baldintza ez funtzionala:	Jakinarazpen kopuru batera iritsitakoan, zaharrenak ezabatzen hasiko dira
Gertaeren Fluxua:	<ol style="list-style-type: none">[Erabiltzailea menutik jakinarazpenak konsultatu botoia sakatzerakoan] (Irudia 57)<ol style="list-style-type: none">1. Iritsi zaion jakinarazpenak begiratuko ditu.[Bilatu botoia sakatzen badu] (Irudia 57)<ol style="list-style-type: none">2.1 EXTENDS Bilatu.
Postbaldintzak:	Erabiltzaileak bere notifikazioak ikusi eta kudeatu ditu
Interfaze grafikoa:	



Irudia 57: Jakinarazpenak kudeatzeko pantaila

1.3.1.1 Bilatu



Irudia 58: Erabiltzaile baten jakinarazpenak bilatzeko erabilpen kasuaren diagrama

Izena: Bilatu

Deskripzioa: Erabiltzailea beste erabiltzaile konkretu baten jakinarazpenak ikusteko aukera izango du erabilpen kasu honekin.

Aktoreak: Erabiltzailea

Aurrebaldintzak: Jakinarazpen mezuaren menuan egotea.

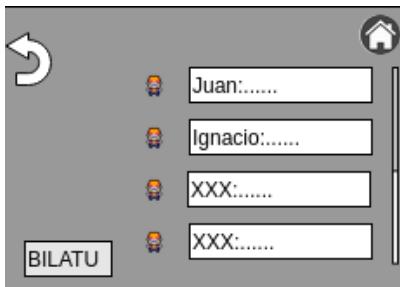
Baldintza ez funtzionala: –

Gertaeren Fluxua:

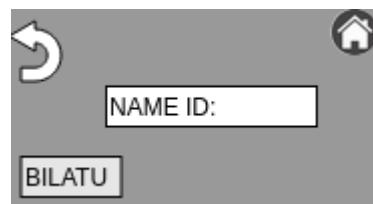
1. [Erabiltzailea Bilatu botoia sakatzen badu] (Irudia 59)
 - 1.1 Beste pantaila bat agertuko da non bilatu nahi duzun erabiltzailearen jakinarazpenak ID-a sar beharko du. (Irudia 60)
 - 1.2 [Sistemak aurkitzen badu erabiltzailea sartutako ID-arekin]
 - 1.2.1 Erabiltzaile horren jakinarazpenak zerrendatuko dira.
 - 1.3 [Sistemak aurkitzen ez badu erabiltzailea sartutako ID-arekin]
 - 1.3.1 Sistemak ID-a berriro sartzeko eskatuko dio erabiltzaileari.

Postbaldintzak: Erabiltzaileak beste erabiltzaile nahi duen notifikazioak ikusi ditu.

Interfaze grafikoa:



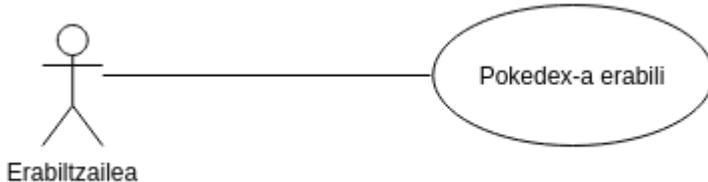
Irudia 59: Jakinarazpenak kudeatzeko pantaila



Irudia 60: Erabiltzaile bilatzeko pantaila

1.4. Pokemonen lista osoa eta bilaketa filtroekin

1.4.1 Pokedex-a



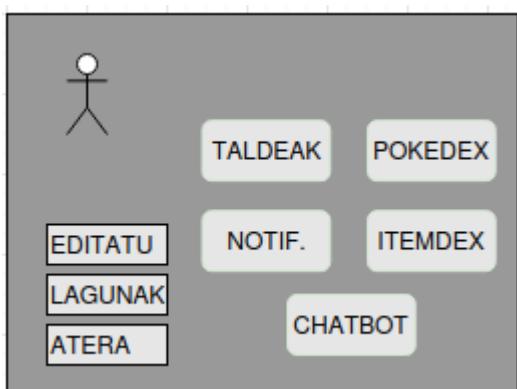
Irudia 61: Pokedex-a erabiltzeko erabilpen kasuaren diagrama

Izena:	Pokedex-a erabili
Deskripzioa: Pokedex-a kontsultatu, pokemon zerrenda guztia ikusiz eta filtroak erabiliz pokemonak bilatu.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: Erabiltzailea saioa hasita euki behar du.	
Baldintza ez funtzionala: –	
Gertaeren Fluxua:	
1. Erabiltzaileak menu nagusiko POKEDEX botoia sakatzen du.(Irudia 62)	
2. Interfazera sartzean, sistemak Pokemon zerrenda guztia erakutsiko du. (Irudia 63 eta Irudia 64)	
2.1. Erabiltzaileak BILATZAILEA erabiltzen du.	
2.1.1. Erabiltzaileak nahi duen Pokemonaren izena sartzen du. (Zenbakiak eta karaktere bereziak onartuta daude)	
2.1.1.1. [Pokemon-a existitzen bada]	
2.1.1.1.1. Pokemon-a agertuko da.	
2.1.1.1.2. [Erabiltzaileak Pokemon-a klikatzean]	
2.1.1.1.2.1. Bere estatistikak eta kate eboluzioa (existitzen bada) agertuko dira. (Irudia 65)	
2.1.1.1.2.2. [Erabiltzaileak “Itzuli” botoian klik egiten badu]	
2.1.1.1.2.2.1. Pokemon zerrendara bueltatuko da. (Irudia 63 eta Irudia 64)	
2.1.1.1.2.3. [Erabiltzaileak “Home” botoian klik egiten badu]	
2.1.1.1.2.3.1. Menu nagusira bueltatu. (Irudia 62)	
2.1.1.2. [Pokemon-a ez bada existitzen]	
2.1.1.2.1. Errore mezua agertuko da eta Lista osoa	

- erakutsiko da. (Irudia 63 eta Irudia 66)
- 2.1.1.2.2. [Erabiltzailea “Itzuli” botoian klik egiten badu]
- 2.1.1.2.2.1. Menu nagusira bueltatuko da. (Irudia 62)
- 2.1.1.2.3. [Erabiltzailea “Home” botoian klik egiten badu]
- 2.1.1.2.3.1. Menu nagusira bueltatuko da. (Irudia 62)
- 2.1.1.3. [Hutsik utzi badu bilatzailea]
- 2.1.1.3.1. Pokemon lista osoa erakutsiko zaio (Irudia 63)
- 2.1.1.4. [Hitz partziala sartu badu]
- 2.1.1.4.1. Hitz partzial horrekin bat egiten duten Pokemon guztiak agertuko dira zerrendan.
- 2.2. Erabiltzaileak IRAGAZKIAK erabiltzen ditu. (Irudia 64)
- 2.2.1. Sistemak erabiltzaileak sartutako iragazkiak betetzen dituzten pokemon zerrenda bat erakutsiko du.
- 2.3. [Erabiltzaileak “Itzuli” botoian klik egiten badu]
- 2.3.1. Menu nagusira bueltatuko da. (Irudia 62)
- 2.4. [Erabiltzaileak “Home” botoian klik egiten badu]
- 2.4.1. Menu nagusira bueltatuko da. (Irudia 62)

Postbaldintzak: Erabiltzaileak sartutako pokemonak aterako d(ir)a (edo ez).

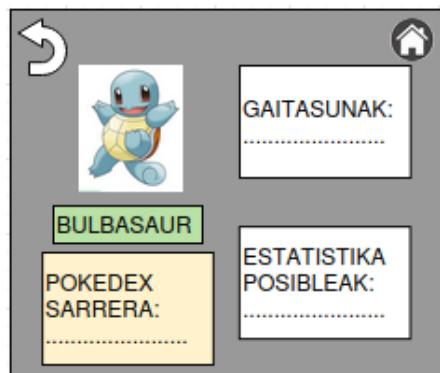
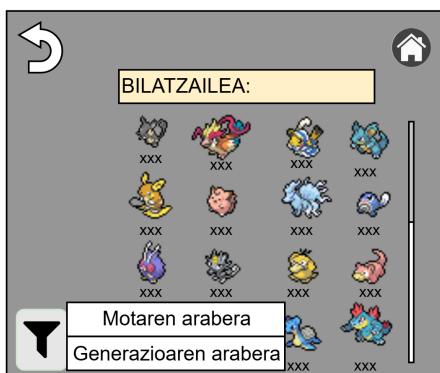
Interfaze grafikoa:



Irudia 62: Menu nagusiaren pantaila



Irudia 63: Pokédex pantaila



Irudia 64: Pokedex pantailaren iragazketak

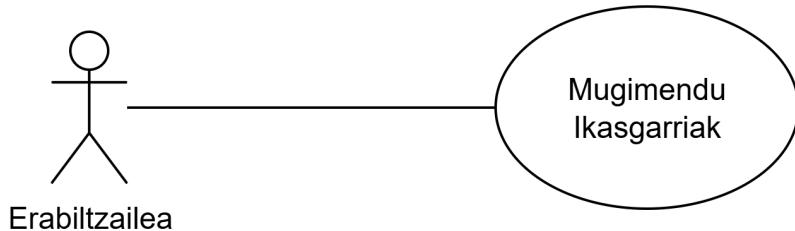
Irudia 65: Pokemon baten gaitasunen pantaila

KONTUZ! EZ DA EXISTITZEN
SARTUTAKO POKEMONA!

Irudia 66: Pokemon-a ez dela existzen errore mezua

1.5. Chatbot-a

1.5.1 Mugimendu ikasgarriak



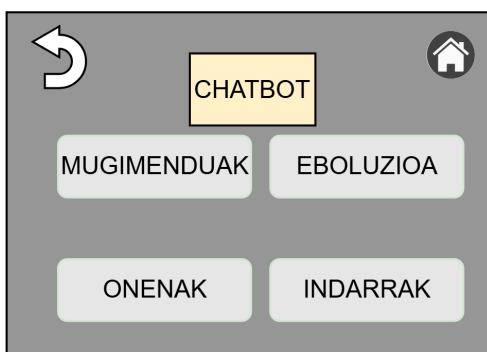
Irudia 67: Pokemon batek dituen mugimendu irakasgarriak erakusteko erabilpen kasuaren diagrama

Izena:	Pokemon baten mugimendu ikasgarriak
Deskripzioa: Erabiltzaileak pokemon bat aukeratzean, sistema honek pokemon horrek naturalki ikasi ahal dituen mugimendu posibileen lista erakusten ditu.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: ChatBot-en menuan egotea.	
Baldintza ez funtzionala: –	
Gertaeren Fluxua:	
1. Erabiltzaileak ChatBot-eko menu nagusian dagoen "MUGIMENDUAK" botoia sakatzen du. (Irudia 68)	
2. Sistemak Pokedex-a erakusten du. (Irudia 69 eta Irudia 70)	
2.1. Erabiltzaileak pokemon baten izena idazten du edo listatik hautatzen du.	
2.1.1. [Erabiltzaileak "Bilatu" botoian klik egiten badu]	
2.1.1.1. [Bilaketak existitzen den Pokemon batekin edo gehiagorekin kointziditzen du eta "Enter" sakatzen badu]	
2.1.1.1.1. Pokemon bat edo gehiago agertuko dira bilatzailearen emaitzetan.	
2.1.1.2. [Bilaketak ez du existitzen den Pokemon batekin ere kointziditzen eta "Enter" sakatzen badu]	
2.1.1.2.1. Pokemon-a ez dela existitzen errore mezua pantailaratuko da. (Irudia 71)	
2.1.1.3. [Bilaketa hutsik uzten da eta "Enter" sakatzen badu]	
2.1.1.3.1. Zerrenda osoa erakutsiko da. (Irudia 69)	

- 2.1.2. [Erabiltzaileak "Iragazketa" botoian klik egiten badu]
- 2.1.2.1. Aplika daitezkeen iragazketa guztien menu zabalgarria agertuko da. (Irudia 70)
- 2.1.2.1.1. [Erabiltzaileak iragazketa bat hautatzen badu]
- 2.1.2.1.1.1. Lista filtratuko da.
- 2.2. Sistema pantaila berri bat erakutsiko du, non hautatutako pokemonaren mugimendu posible guztiak agertzen diren. (Irudia 72)
- 2.3. [Erabiltzaileak "Itzuli" botoian klik egiten badu]
- 2.3.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 69 eta Irudia 70)
3. [Erabiltzaileak "Itzuli" botoian klik egiten badu]
- 3.1. Sistemak ChatBot-aren bilatzailearen pantailara bueltatzen da. (Irudia 68)
4. [Erabiltzaileak "Home" botoian klik egiten badu]
- 4.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 73)

Postbaldintzak: Erabiltzaileak pokemon baten mugimendu natural posibleak ikusi ditu.

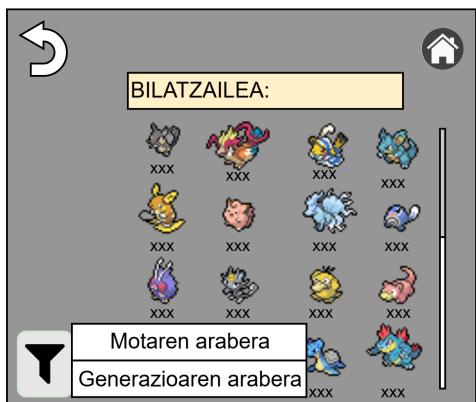
Interfaze grafikoa:



Irudia 68: Chatbot pantaila



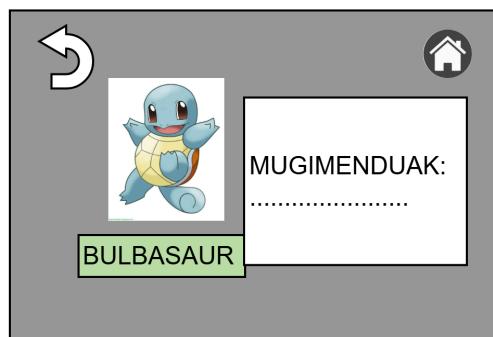
Irudia 69: Pokedex pantaila



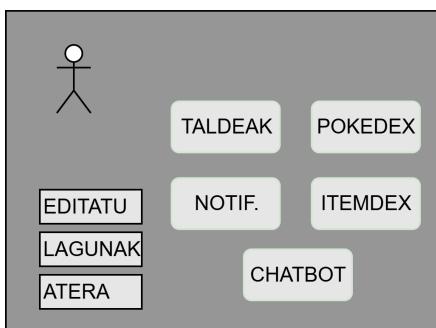
Irudia 70: Pokédex pantailaren iragazketak

KONTUZ! EZ DA EXISTITZEN SARTUTAKO POKEMONA!

Irudia 71: Pokemon-a ez existitzen ez delaren errore mezua

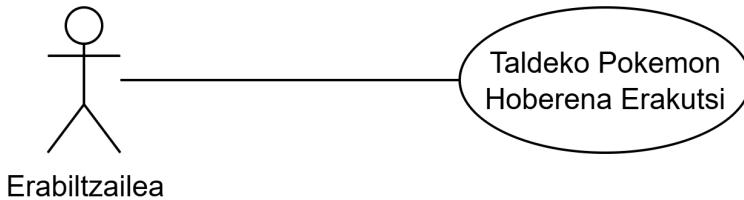


Irudia 72: Pokemon baten mugimendu ikasgarriak



Irudia 73: Menu nagusiaren pantaila

1.5.2 Taldeko pokemon hoberena



Irudia 74: Pokemon talde baten Pokemon hoberena erakusteko erabilpen kasuaren diagrama

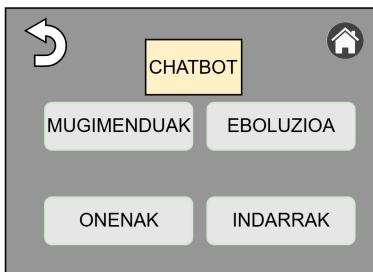
Izena:	Taldeko pokemon hoberena erakutsi
Deskripzioa: Sistema honek erabiltzailearen talde bateko pokemonik hoberena identifikatzen eta erakusten du, puntuazio-sistema baten arabera.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: ChatBot-en menuan egotea eta gutxienez talde bat sortuta egotea.	
Baldintza ez funtzionala: –	
Gertaeren Fluxua:	
<ol style="list-style-type: none">1. Erabiltzaileak ChatBot-eko menu nagusian dagoen "ONENAK" botoia sakatzen du. (Irudi 75)2. [Taldea(k) bad(it)u]<ol style="list-style-type: none">2.1. Sistemak taldeen listak erakusten ditu. (Irudi 76)2.2. Erabiltzaileak talde bat hautatzen du.2.3. Sistemak pantaila berri bat erakusten du, non taldeko pokemon hoberena nabarmentzen den, haren puntuazioa erakutsiz. (Irudi 77)2.4. [Erabiltzaileak "Itzuli" botoian klik egiten badu]<ol style="list-style-type: none">2.4.1. Sistemak "Taldeen lista" pantailara bueltatzen da. (Irudi 76)3. [Ez du talderik]<ol style="list-style-type: none">3.1. Sistema mezu bat erakutsiko du, "Lehenik, taldea sortu" esanez. (Irudi 78)4. [Erabiltzaileak "Itzuli" botoian klik egiten badu]<ol style="list-style-type: none">4.1. Sistemak "ChatBot-eko" menu nagusia pantailara bueltatzen da. (Irudi 78)	

5. [Erabiltzaileak "Home" botoian klik egiten badu]

5.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 79)

Postbaldintzak: Erabiltzaileak bere taldeko pokemonik indartsuena ikusi du, puntuazio-sistema baten arabera.

Interfaze grafikoa:



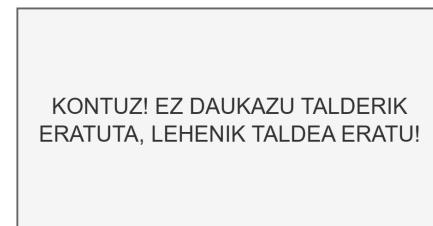
Irudia 75: Chatbot pantaila



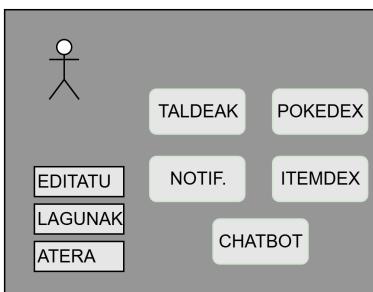
Irudia 76: Erabiltzailearen taldeen zerrenda



Irudia 77: Pokemon baten gaitasunen pantaila, hoherena dela erakutsiz

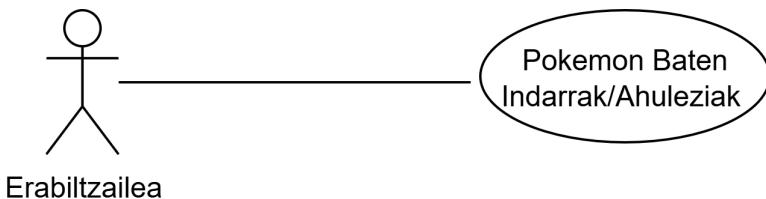


Irudia 78: Talde eratuta ez dagoelaren errore mezua



Irudia 79: Menu nagusiaren pantaila

1.5.3 Ahuleziak eta indarrak



Irudia 80: Pokemon baten indarrak/ahuleziak erakusteko erabilpen kasuaren diagrama

Izena: Pokemon baten ahuleziak eta indarrak erakutsi

Deskripzioa: Erabiltzaileak pokemon bat aukeratzen duenean, sistemak erakutsiko dio zein motaren aurka indartsua den eta zeinen aurka ahulagoa den. Gainera, erabiltzaileak Pokedex-eko bilatzailea erabili dezake nahi duen pokemonen topatzeko.

Aktoreak: Erabiltzailea

Aurrebaldintzak: ChatBot-en menuan egotea.

Baldintza ez funtzionala: –

Gertaeren Fluxua:

1. Erabiltzaileak ChatBot-eko menu nagusian dagoen "Indarrak" botoia sakatzen du. (Irudia 81)
2. Sistemak Pokedex-aren bilatzailea erakusten du. (Irudia 82 eta Irudia 83)
 - 2.1. [Erabiltzaileak "Bilatu" botoian klik egiten badu]
 - 2.1.1. [Bilaketak existitzen den Pokemon batekin edo gehiagorekin kointziditzen du eta "Enter" sakatzen badu]
 - 2.1.1.1. Pokemon bat edo gehiago agertuko dira bilatzailearen emaitzetan.
 - 2.1.2. [Bilaketak ez du existitzen den Pokemon batekin ere kointziditzen eta "Enter" sakatzen badu]
 - 2.1.2.1. Pokemon-a ez dela existitzen errore mezua pantailaratuko da. (Irudia 84)
 - 2.1.3. [Bilaketa hutsik uzten da eta "Enter" sakatzen badu]
 - 2.1.3.1. Zerrenda osoa erakutsiko da. (Irudia 82)
 - 2.2. [Erabiltzaileak "Iragazketa" botoian klik egiten badu]
 - 2.2.1. Aplika daitezkeen iragazketa guztien menu zabalgarria

agertuko da. (Irudia 83)

2.2.1.1. [Erabiltzaileak iragazketa bat hautatzen badu]

2.2.1.1.1. Lista filtratuko da.

3. Erabiltzaileak pokemon baten izena idazten du edo listatik hautatzen du.

3.1. Sistemak pantaila berri bat erakusten du, non pokemon hautatuaren mota, zein motatako aurkariei egiten dien eraso gehiago eta zeini gutxiago erakusten diren. (Irudia 85)

3.2. [Erabiltzaileak "Itzuli" botoian klik egiten badu]

3.2.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 82 eta Irudia 83)

4. [Erabiltzaileak "Itzuli" botoian klik egiten badu]

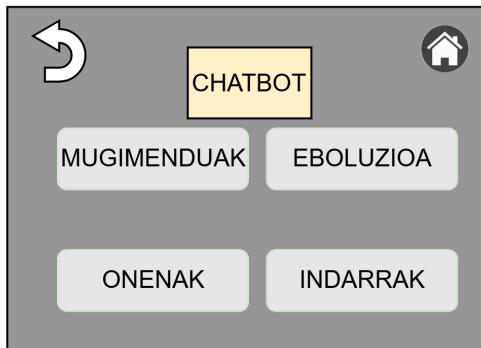
4.1. Sistemak Chatbot-aren bilatzailearen pantailara bueltatzen da. (Irudia 81)

5. [Erabiltzaileak "Home" botoian klik egiten badu]

5.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 86)

Postbaldintzak: Erabiltzaileak pokemon baten indar eta ahulezien informazioa ikusi du.

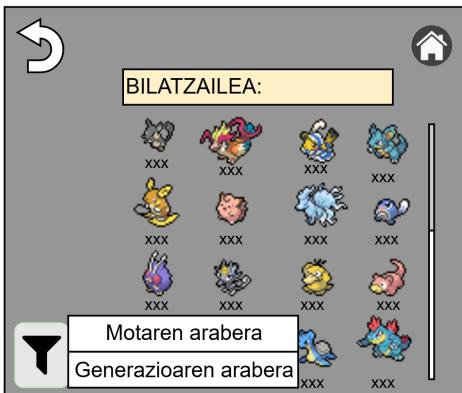
Interfaze grafikoa:



Irudia 81: Chatbot pantaila



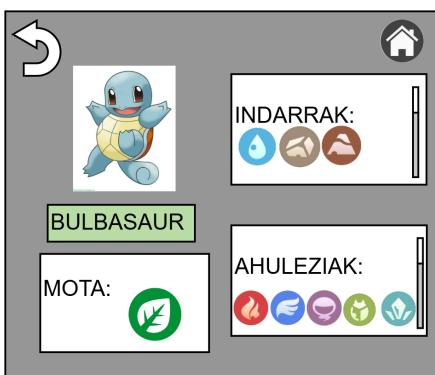
Irudia 82: Pokedex pantaila



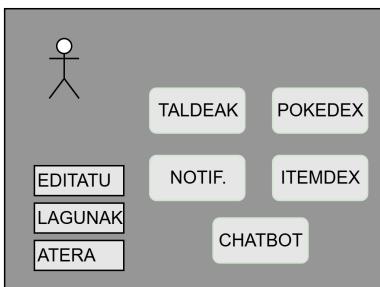
Irudia 83: Pokédex pantailaren iragazketak

KONTUZ! EZ DA EXISTITZEN SARTUTAKO POKEMONA!

Irudia 84: Pokemon-a existitzen ez delaren errore mezua

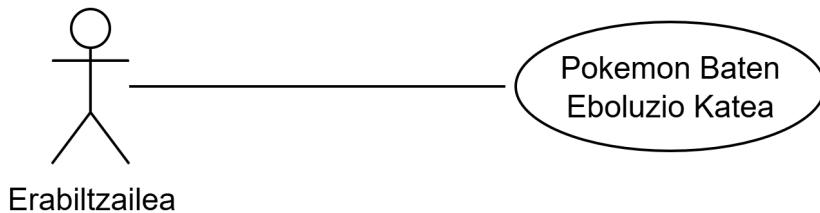


Irudia 85: Pokemon baten ahuleziak eta indarguneak



Irudia 86: Menu nagusiaren pantaila

1.5.4 Eboluzio katea



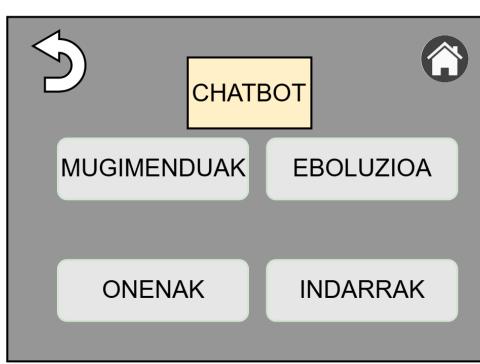
Irudia 87: Pokemon baten eboluzio katea erakusteko erabilpen kasuaren diagrama

Izena:	Pokemon baten eboluzio katea erakutsi
<p>Deskripzioa: Erabiltzaileak pokemon bat aukeratzen duenean, sistemak haren eboluzio kate osoa erakusten du. Gainera, erabiltzaileak Pokedex-eko bilatzailea erabili dezake nahi duen pokemonen topatzeko.</p>	
<p>Aktoreak: Erabiltzailea</p>	
<p>Aurrebaldintzak: -</p>	
<p>Baldintza ez funtzionala: -</p>	
<p>Gertaeren Fluxua:</p> <ol style="list-style-type: none">1. Erabiltzaileak ChatBot-eko menu nagusian dagoen "Eboluzioa" botoia sakatzen du. (Irudia 88)2. Sistemak Pokedex-aren bilatzailea erakusten du. (Irudia 89 eta Irudia 90)<ol style="list-style-type: none">2.1. [Erabiltzaileak "Bilatu" botoian klik egiten badu]<ol style="list-style-type: none">2.1.1. [Bilaketak existitzen den Pokemon batekin edo gehiagorekin kointziditzen du eta "Enter" sakatzen badu]<ol style="list-style-type: none">2.1.1.1. Pokemon bat edo gehiago agertuko dira bilatzailearen emaitzetan.2.1.2. [Bilaketak ez du existitzen den Pokemon batekin ere kointziditzen eta "Enter" sakatzen badu]<ol style="list-style-type: none">2.1.2.1. Pokemon-a ez dela existitzen errore mezua pantailaratuko da. (Irudia 91)2.1.3. [Bilaketa hutsik uzten da eta "Enter" sakatzen badu]<ol style="list-style-type: none">2.1.3.1. Zerreenda osoa erakutsiko da. (irudia 89)2.2. [Erabiltzaileak "Iragazketa" botoian klik egiten badu]<ol style="list-style-type: none">2.2.1. Aplika daitezkeen iragazketa guztien menu zabalarria agertuko da.<ol style="list-style-type: none">2.2.1.1. [Erabiltzaileak iragazketa bat hautatzen badu]<ol style="list-style-type: none">2.2.1.1.1. Lista filtratuko da.	

- 2.3.1. Sistema pantaila berri bat erakusten du, non pokemonaren eboluzio-katea grafiko moduan agertzen den, eboluzio guztiak eta haien arteko loturak erakutsiz. (Irudia 92)
- 2.4. [Eboluziorik ez badu]
- 2.4.1. Mezu bat agertzen da "Eboluziorik ez" esanez. (Irudia 93)
- 2.5. [Erabiltzaileak "Itzuli" botoian klik egiten badu]
- 2.5.1. Sistemak Pokedex-aren bilatzailearen pantailara bueltatzen da. (Irudia 89 eta Irudia 90)
3. [Erabiltzaileak "Itzuli" botoian klik egiten badu]
- 3.1. Sistemak Chatbot-aren bilatzailearen pantailara bueltatzen da. (Irudia 88)
4. [Erabiltzaileak "Home" botoian klik egiten badu]
- 4.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 94)

Postbaldintzak: Erabiltzaileak pokemon baten indar eta ahulezien informazioa ikusi du.

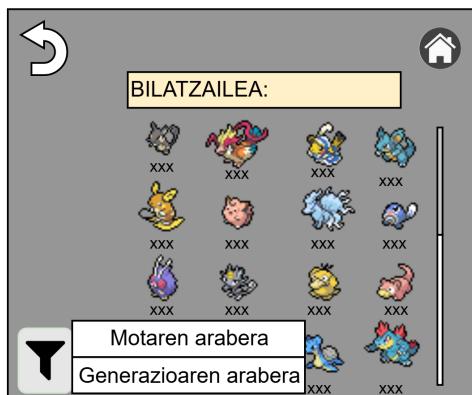
Interfaze grafikoa:



Irudia 88: Chatbot pantaila



Irudia 89: Pokedex pantaila



Irudia 90: Pokédex pantailaren iragazketak

KONTUZ! EZ DA EXISTITZEN
SARTUTAKO POKEMONA!

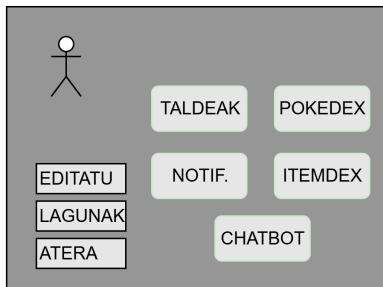
Irudia 91: Pokemon ez dela existitzearen errore mezua



92: Pokemon baten eboluzio katea

KONTUZ! HAUTATUTAKO POKEMONA
EZ DU EBOLUZIORIK!

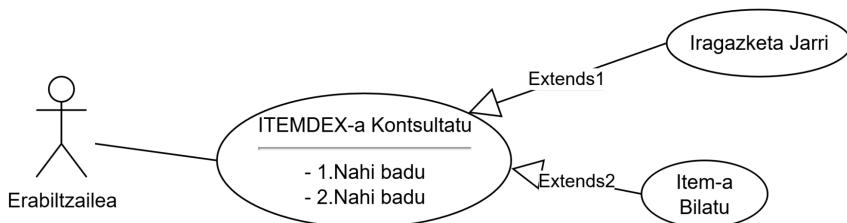
Irudia 93: Aukeratutako Pokemon-a eboluziorik ez duelaren errore mezua



Irudia 94: Menu nagusiaren pantaila

1.6. Hautazko funtzionalitatea

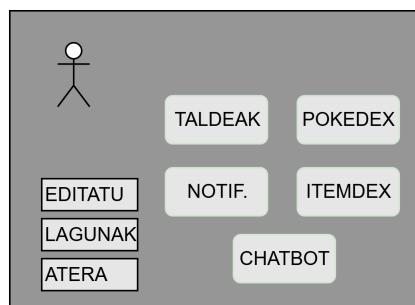
1.6.1 Itemdex



Irudia 95: Idemdex-aren erabilpen kasuen diagrama

Izena:	Itemdex
Deskripzioa: Erabiltzaileak item guztien lista ikus eta ordena ditzake nahi duen bezala (CDen arabera, pikorren arabera, edabeen arabera...). Bestalde, filtroekin bila dezake item jakin bat. Baten bat sakatzen bada, itemaren informazioa agertzen da beste leihoko batean. Adibidez, edabe bat sakatzen bada, honen deskripzioa, ikonoa eta zenbat HP berreskuratzen den.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: –	
Baldintza ez funtzionala: –	
Gertaerentzako Fluxua:	
1. Erabiltzileak menu nagusian dagoen "ITEMDEX" botoia sakatzen du. (Irudia 96)	
2. [Erabiltzaileak "BILATZAILEA" erabiltzen badu] (Irudia 97)	
2.1. EXTENDS Item-a bilatu	
3. [Erabiltzaileak "Iragazketa" botoian klik egiten badu] (Irudia 98)	
3.1. EXTENDS Iragazketa jarri	
4. [Erabiltzaileak "Itzuli" botoian klik egiten badu]	
4.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 96)	
5. [Erabiltzaileak "Home" botoian klik egiten badu]	
5.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 96)	
Postbaldintzak: Erabiltzaileak bere item-a aurkitu du (ala ez).	

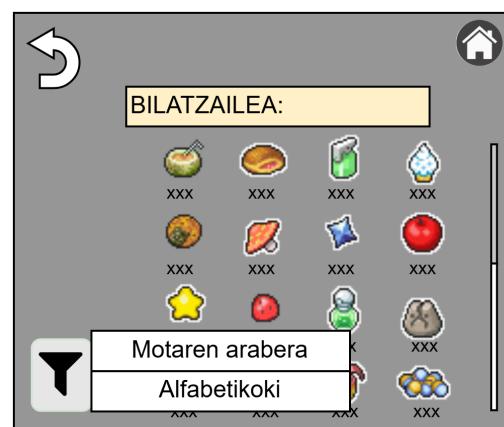
Interfaze grafikoa:



Irudia 96: Menu nagusiaren pantaila

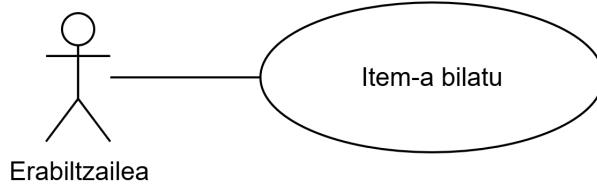


Irudia 97: Item zerrenda pantaila



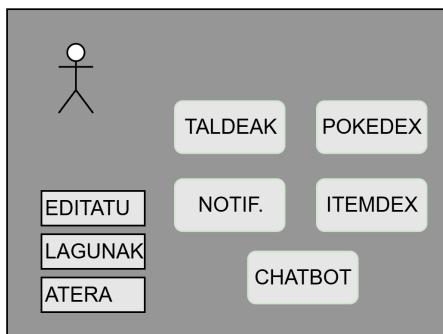
Irudia 98: Item zerrenda iragazketak

1.6.1.1 Item-a bilatu



Irudia 99: Item-a bilatu erabilpen kasuaren diagrama

Izena:	Item-a bilatu
<p>Deskripzioa: Erabiltzaileak item guztien lista ikus eta ordena ditzake nahi duen bezala (CDen arabera, pikorren arabera, edabeen arabera...). Baten bat sakatzen bada, item-aren informazioa agertzen da beste leihoa batean. Adibidez, edabe bat sakatzen bada, honen deskripzioa, ikonoa eta zenbat HP berreskuratzten den.</p>	
<p>Aktoreak: Erabiltzailea</p>	
<p>Aurrebaldintzak: –</p>	
<p>Baldintza ez funtzionala: –</p>	
<p>Gertaeraren Fluxua:</p> <ol style="list-style-type: none">1. Erabiltzaileak, item bat bilatu dezake bere izena jarriz.<ol style="list-style-type: none">1.1. [Itema existitzen bada]<ol style="list-style-type: none">1.1.1. Pantailan erakutsiko du eta itema sakatzean bere deskripzio laburra, ikonoa eta erabilpena erakutsiko ditu beste leihoa batean. (Irudia 102)1.2. [Itema ez bada existitzen]<ol style="list-style-type: none">1.2.1. Errore mezu bat agertuko da ez dela existitzen esanik. (Irudia 103)2. [Erabiltzaileak "Itzuli" botoian klik egiten badu]<ol style="list-style-type: none">2.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 100)3. [Erabiltzaileak "Home" botoian klik egiten badu]<ol style="list-style-type: none">3.1. Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 100)	
<p>Postbaldintzak: Erabiltzaileak item-a aurkitu du (ala ez).</p>	
<p>Interfaze grafikoa:</p>	



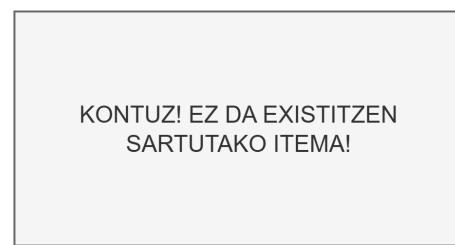
Irudia 100: Menu nagusiaren pantaila



Irudia 101: Item zerrenda pantaila

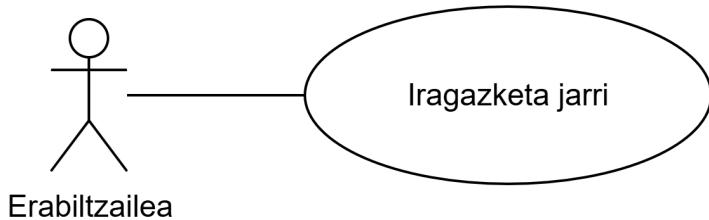


Irudia 102: Item baten informazio pantaila



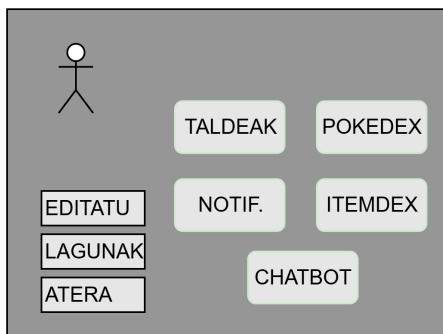
**Irudia 103: Item-a ez dela
aurkitu errore mezua**

1.6.1.2 Iragazketa jarri



Irudia 104: Iragazketa jarri erabilpen kasuaren diagrama

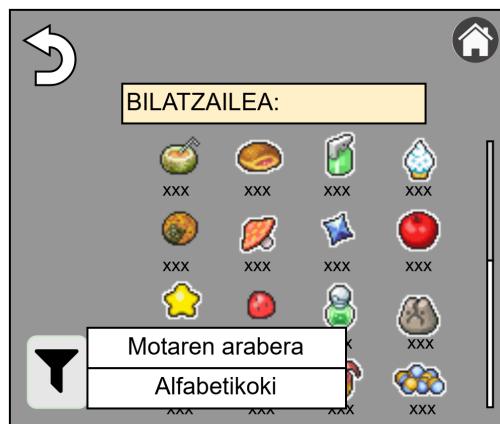
Izena:	Iragazketa jarri
Deskripzioa: Erabiltzaileak item guztien lista irakagizki ahal dezake eta filtroekin bila dezake item jakin bat.	
Aktoreak: Erabiltzailea	
Aurrebaldintzak: –	
Baldintza ez funtzionala: –	
Gertaeren Fluxua: <ol style="list-style-type: none">[Erabiltzaileak "Iragazketa" botoian klik egiten badu] (Irudia 107)<ol style="list-style-type: none">[“Motaren arabera” klik egiten badu]<ol style="list-style-type: none">Erabiltzaileak mota bat aukeratuko du, eta zerrenda mota horren arabera filtratuko da.[“Alfabetikoki” klik egiten badu]<ol style="list-style-type: none">Pokemon zerrenda A-tik Z-ra ordenatuko da.[Erabiltzaileak "Itzuli" botoian klik egiten badu]<ol style="list-style-type: none">Sistemak ItemDex pantailara bueltatzen da. (Irudia 106)[Erabiltzaileak "Home" botoian klik egiten badu]<ol style="list-style-type: none">Sistemak menu nagusiaren pantailara bueltatzen da. (Irudia 105)	
Postbaldintzak: Erabiltzaileak objektuak iragazki ditu.	
Interfaze grafikoa:	



Irudia 105: Menu nagusiaren pantaila



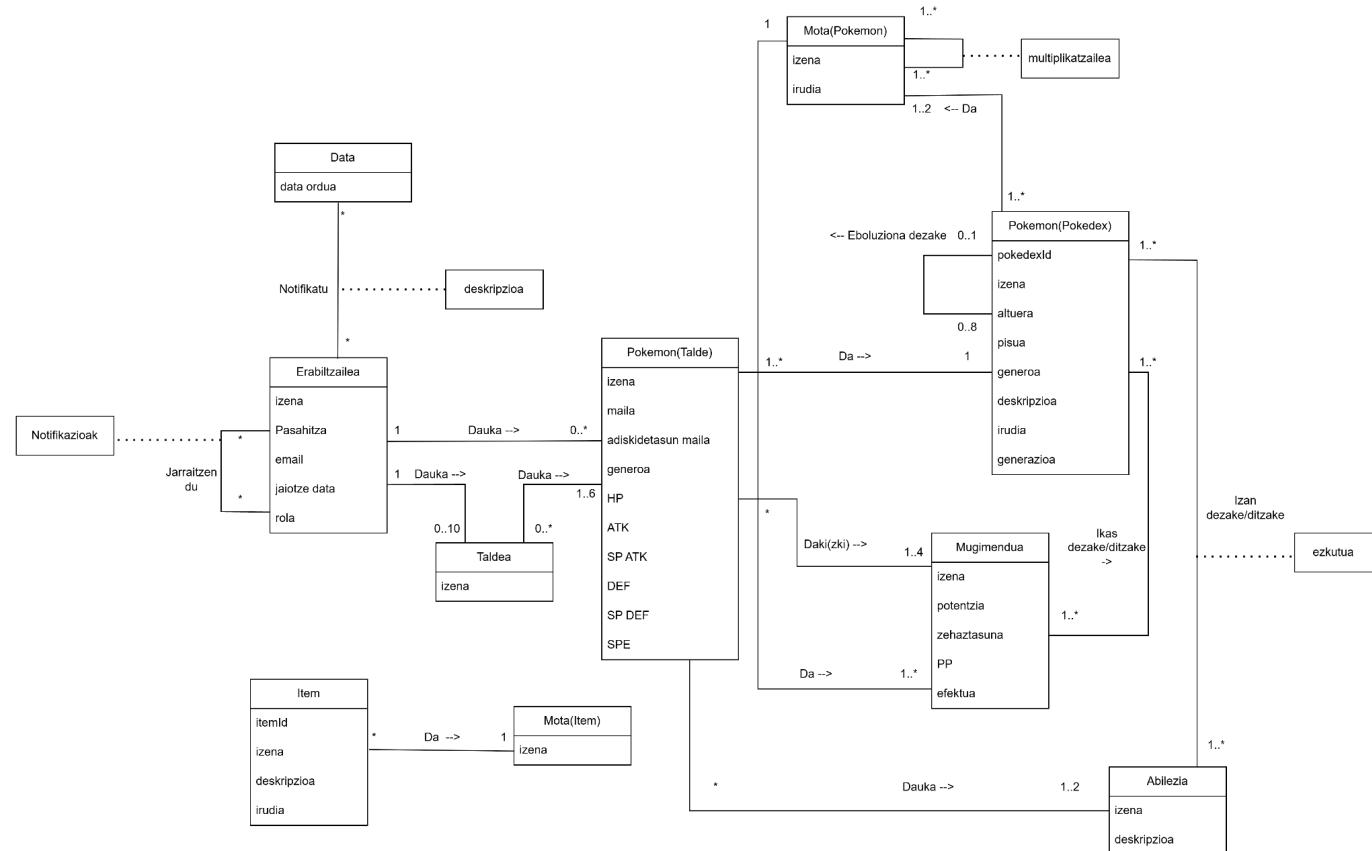
Irudia 106: Item bilatzailea pantaila



Irudia 107: Item zerrendaren iragazketak

2. Entregagarria (Domeinuaren eredua)

2.1 Domeinuaren eredua osoa



Irudia 108: Domeinu ereduaren diagrama osoa

2.1.1 Azalpen gehigarriak

Txarto ikustekotan, link honen bidez hobeto ikusi daiteke:

<https://drive.google.com/file/d/1KaY3N88ZsDggosTPkswuYDAqZZbzkO7Y/view?usp=sharing>

2.2 Entitateak

2.2.1. Erabiltzailea

Erabiltzailearen informazioa gordetzen duen entitatea. Kontuan izan behar dugu erregistratuta egon behar dela informazio hau sistemaren gordetzeko. Hauek dira entitatearen atributuak:

- Izena: Erabiltzailearen izena.
- Pasahitza: Saioa hasteko sartu behar duen pasahitza.
- Email: Erabiltzailearen email helbidea non sistemak batzutan mezuak bidaliko du.
- Jaiotze-data: Erabiltzailearen jaiotze-data.
- Rola: atributu honek zehazten du identifikatutako erabiltzailea erabiltzaile arrunta edo administratzalea den.

2.2.2. Taldea

Erabiltzaile batek bere Pokemon-en arteko 6 pokemon multzoa.

- Izena: Pokemon-taldearen izena.

2.2.3. Pokemon (Talde)

Entitate honek harrapatutako pokemon-a gordetzen du. Pokemon bakoitzak bere ezaugarri pertsonalak eta entrenamendu-datuak ditu. Hauek dira entitatearen atributuak:

- Izena: Pokemonaren izena.
- Maila: Pokemonaren maila.
- Adiskidetasun maila: Pokemonaren adiskidetasun maila.
- Generoa: Pokemonaren generoa(arra, hemea, edo ezer ez).
- HP: Pokemonaren sasun puntuak.
- ATK: Pokemonaren eraso indarra.
- SP ATK: Pokemonaren eraso berezi indarra.
- DEF: Pokemonaren defentsa.
- SP DEF: Pokemonaren defentsa berezia.
- SEP: Pokemonaren abiadura.

2.2.4. Pokemon (Pokedex)

Aurki daitezkeen pokemonei buruz konsulta daitekeen oinarrizko informazioa. Hona hemen entitatearen atributuak:

- PokedexID: Pokemon bakoitza pokedex-ean identifikatzen duen zenbakia.
- Izena: Pokemonaren izena.
- Altuera: Pokemonaren altuera estandarra.
- Pisua: Pokemonaren pisu estandarra.
- Generoa: Pokemonaren generoa(arra, hemea, edo ezer ez).
- Deskripzioa: Pokemonari buruzko deskripzio txiki bat.
- Irudia: Argazkia gordetzen duen path-a
- Generazioa: Pokemon-a lehenengo aldiz agertu zen generazioa.

2.2.5. Mugimendua

Pokemon-ak ikas edo izan dezaketen borroka teknikak dira.

- Izena: Mugimendua duen izena.
- Potentzia: Mugimenduaren indarra zehazten du.
- Zehaztasuna: Mugimenduak aurkariari jotzeko duen probabilitatea da (Ehunekotan).
- PP: Mugimendua bera zenbat aldiz erabil daiteke adierazten du (Berreskuratu daitezke).
- Efektua: Mugimendua erabili ostean gertatuko dena edo gertatu ahal dena. Adibidez, “Espora” izeneko mugimendua aurkaria lotan utziko du, baina “Danza Espada” izenekoa zuri ATK estatistika igoko ditzu.

2.2.6. Abilezia

Pokemon-ak daukaten edo izan dezaketen abileziak dira.

- Izena: Abileziak duen izena.
- Deskripzioa: Abileziaren zehaztasunak.

2.2.7. Mota (Pokemon)

Pokemon bakoitzak izan dezakeen mota edo motak.

- Izena: Motaren izena.
- Irudia: Argazkia gordetzen duen path-a

2.2.8. Item

Erabiltzaileka Pokemon-engan erabiltzen duten elementuak dira arrazoi ezberdinetarako, harrapatzeko (Poke Ball), bitzitzera berreskuratzeko (Pozioak), borroketan erabitzeako, besteak beste.

- Itemid: Item-aren id zehazten du
- Izena: Item-a duen izena.
- Deskripzioa: Item-ak zertarako balio duen azalpena.
- Irudia: Argazkia gordetzen duen path-a

2.2.9. Mota (Item)

Item-en iragazketa egiterakoan, multzoak egiteko erabiltzen den irizpidea da.

- Izena: Item multzo motak duen izena.

2.2.10 Data

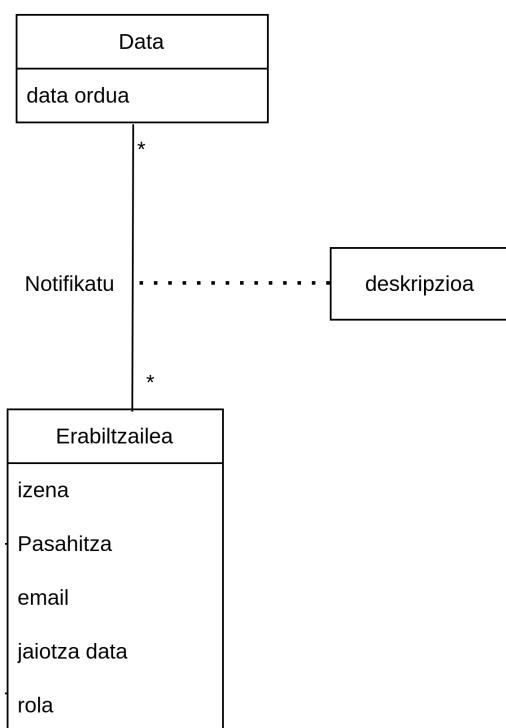
Erabiltzaileari noiz iritsi zaion jakinarazpen bat gordetzen duen entitate bat da.

- Data: Jakinarazpenaren data gordetzen duen atributua.
- Ordua: Jakinarazpenaren iritsi den ordua gordetzen duen atributua.

2.3. Erlazioak

2.3.1. Notifikatu

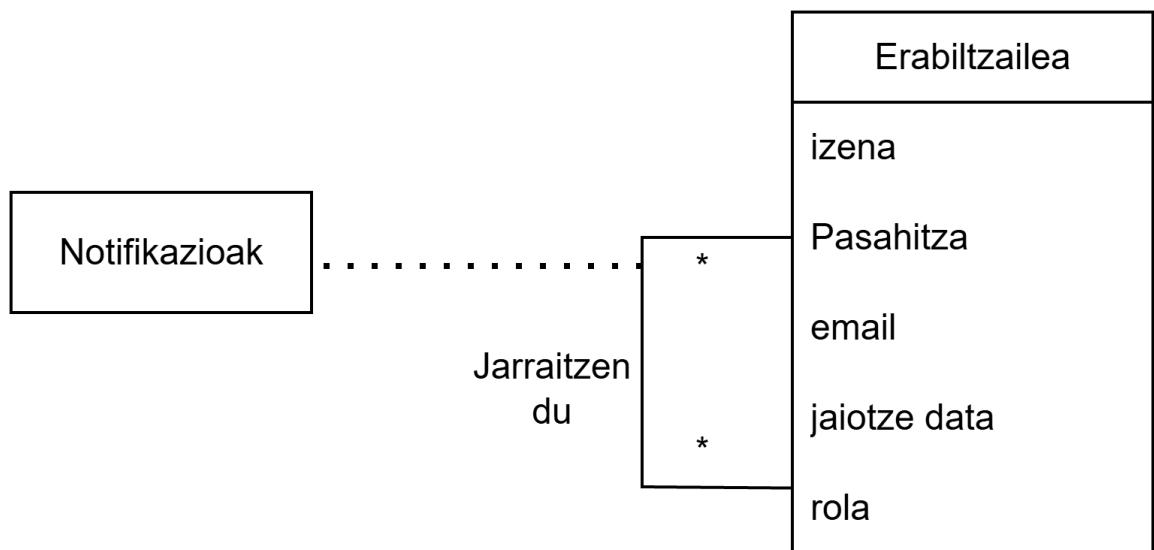
Data eta erabiltzaile entitateen arteko erlazio bat da. Erlazio honek erabiltzailearen eta sistemaren arteko notifikazio-sistemarako balio du. Alde batetik, notifikazio hori erabiltzaileari iritsi zaion eguna eta ordua gordetzen dira. Bestalde, notifikazioaren deskribapena gordetzen duen elkartze-entitatea du, hau da, notifikazioaren edukia. Amaitzeko, erlazio honen kardinalitatea *-koa da erlazioaren bi aldeetan erabiltzaile bakoitzak hainbat datetan jaso ditzake notifikazio bera, eta data berean erabiltzaile bati baino gehiagorri notifikazio bera bidal dokiekeelako.



Irudia 109: Notifikatu erlazioaren diagrama

2.3.2. Jarraitzen du (Erabiltzaile -> Erabiltzaile)

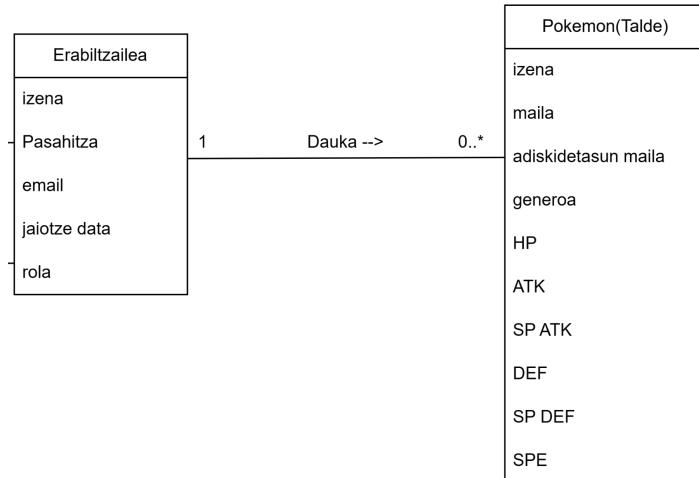
Erlazio honi esker, erabiltzaile baten jarraitzaileak eta jarraitzen dituen erabiltzaileak kudeatzen dira. Notifikazioak, atributu boolearra izango du Erabiltzaile horrek beste erabiltzailearen notifikazioak ezartzeko edo desgaitzeko. Erabiltzaile entitatea, berekiko erlazionatzen da. Erabiltzaile bakotzak, hainbat erabiltzaile jarraitu (0..**) eta era berean, hainbat jarraitzaile izan ditzake.



Irudia 110: Jarraitzen du (Erabiltzaile -> Erabiltzaile) erlazioaren diagrama

2.3.3. Dauka (Erabiltzaile -> Pokemon (Talde))

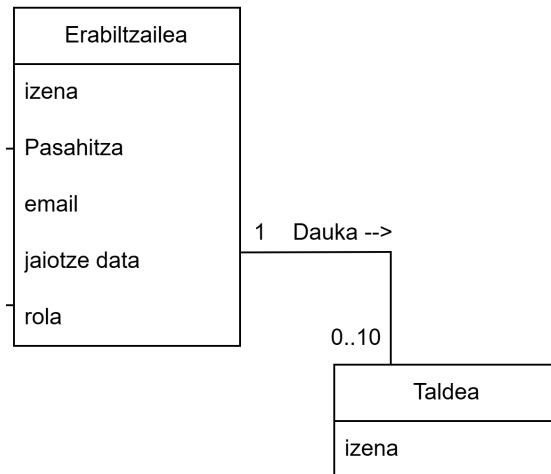
Erlazio honek erabiltzailea eta hark harrapatutako pokemonak kudeatzeko balio du. Erabiltzaile bakoitzak hainbat pokemon harrapa ditzake (0..*), baina pokemon bakoitza erabiltzaile bakar batek harrapa dezake (1). Gezia irakurketaren noranzkoa adierazten du: erabiltzaileak pokemonak dauzka, eta ez alderantziz.



Irudia 111: Dauka (Erabiltzaile -> Pokemon (Talde)) erlazioaren diagrama

2.3.4. Dauka (Erabiltzaile -> Talde)

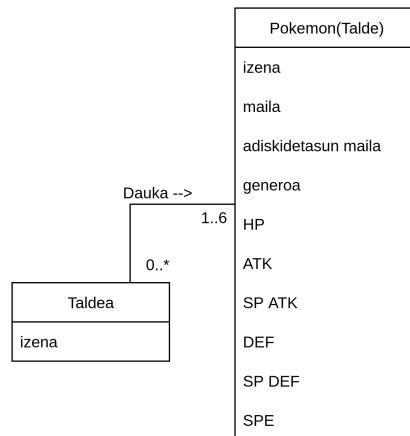
Erabiltzailea eta pokemon taldeen arteko erlazioa da, bertan adierazten dira erabiltzaileak dituen Pokemon taldeak. Gure aplikazioan, erabiltzaile batek ez du zertain Pokemon talderik izan, baina gehienez 10 talde izan ahalko ditu. Talde bakoitza bakarrik erabiltzaile batekoa izango da.



Irudia 112: Dauka (Erabiltzaile -> Talde) erlazioaren diagrama

2.3.5. Dauka (Talde -> Pokemon (Talde))

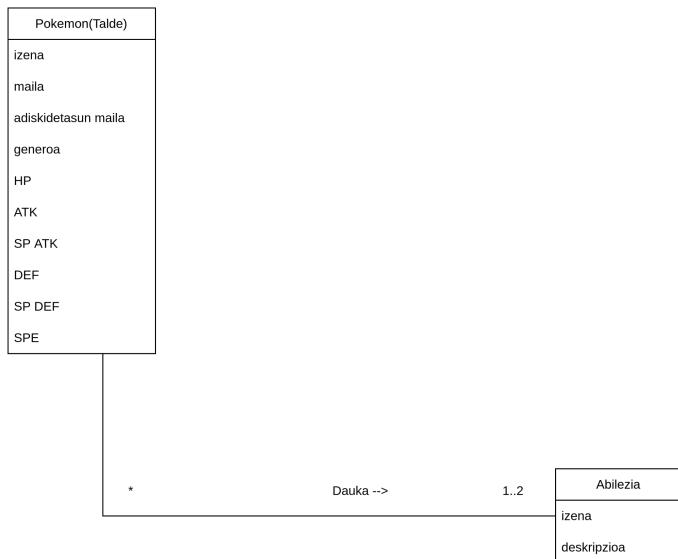
Pokemon entitatea “Dauka” atributuaren bitartez Taldearekin lotuta dago. Honek adierazten du harrapatutako Pokemon batek edo 0 aldiz (ez da sartu talde batean, Pokemon kutxan dago) edo n aldiz (n taldetan dago) egon daitezkela esan nahi du. Bestalde, talde batek, sortuta baldin badago, gutxienez Pokemon bat izan dezake, eta gehienez sei (talde osoa).



Irudia 113: Dauka (Talde -> Pokemon (Talde)) erlazioaren diagrama

2.3.6. Dauka (Pokemon (Talde) -> Abilezia)

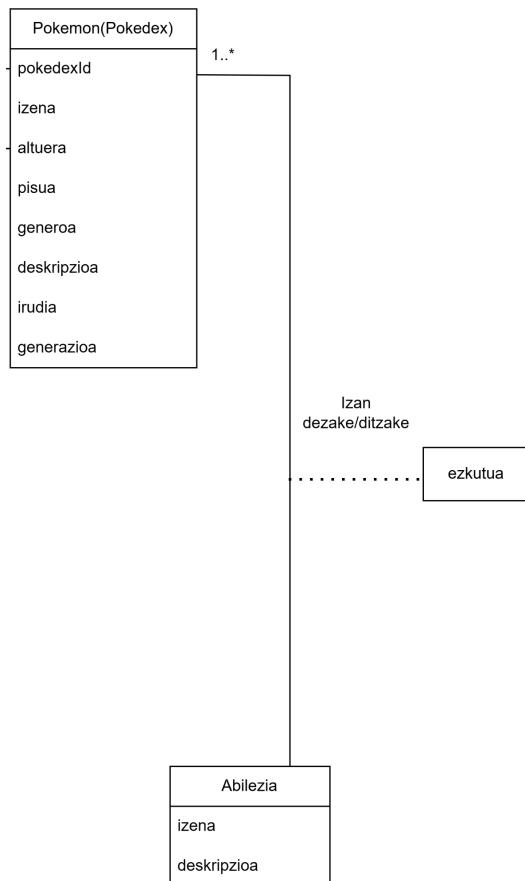
Pokemon (talde) eta abilezia entitateen arteko erlazio bat da. Bi entitate hauek eralzionatzen dira taldeko pokemon bakoitzak 1 edo 2 abilezia izan ahal duelako. Gainera, abilezi berbera hainbat pokemon (talde) desberdinetan aurki ditzake, horren ondorioz * aurkitu dezakegu pokemon (talde) entitatearen aldean. Erlazio hau abilezia horren informazioa eta zeri pokemon dagokio gordetzeko balio du.



Irudia 114: Dauka (Pokemon (Talde) -> Abilezia) erlazioaren diagrama

2.3.7. Izan dezake/ditzake

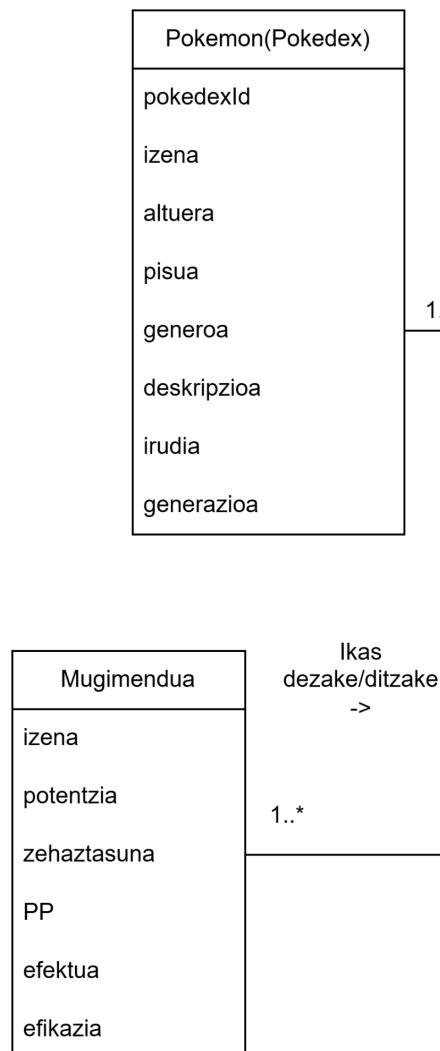
Pokemon espezie bakoitzak izan ditzakeen abilezien informazioa gordetzen duen erlazioa. Pokemon eta Abilezia entitateak erlazionatzen dira. Pokemon batek, abilezi bat edo asko (1..*) izan ahal ditu, eta abilezi bakoitzak, Pokemon askok euki ditzakete edo bat erez (1..*). Gainera zehazten da Pokemon espezie horrentzat abilezi ezkutua denentz.



Irudia 115: Izan dezake/ditzake erlazioaren diagrama

2.3.8. Ikas dezake/ditzake

Pokedexeko pokemonak mugimenduak ikas ditzakeenez, erlazio honek haien informazioa gordetzeko balio du. Pokemon batek mugimendu asko ikas ditzake eta gutxienez bat jakingo du (1..*), eta mugimendu batek pokemon batek ikasi dezake, baina gutxienez bat ikasiko du (1..*).

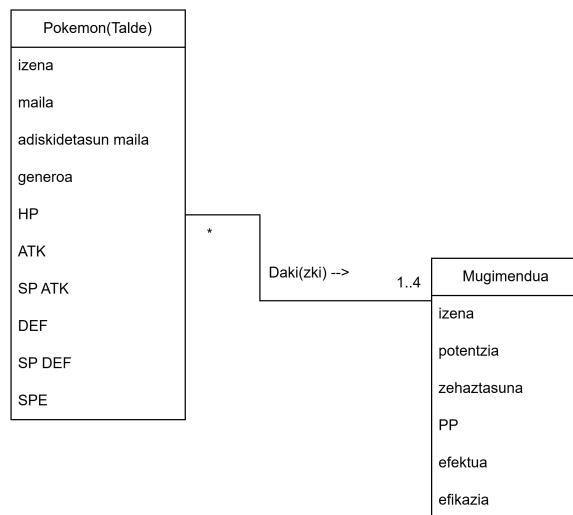


Irudia 116: Ikas dezake/ditzake erlazioaren diagrama

2.3.9. Dak(izk)i

Erlazio honen bitartez, taldeko pokemon bakoitzak ze mugimendu dakizkien gordeko ditu.

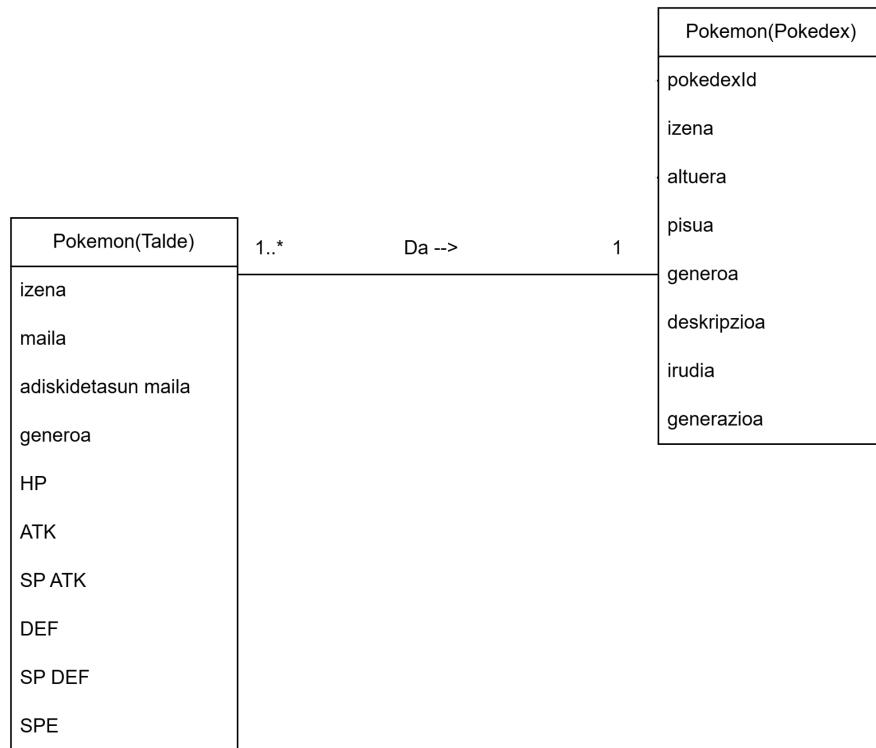
Kardinalitateari dagokionez, pokemon batek gutxienez mugimendu bat izango du eta gehienez 4, eta pokemon batek ez du zertain mugimendu hori eduki behar, edo bat baino gehiago izan ditzake.



Irudia 117: Dak(izk)i erlazioaren diagrama

2.3.10. Da (Pokemon (Talde) -> Pokemon (Pokedex))

Erlazio honen bidez talde batean pokemon bakoitzarekiko bere Pokedex-aren sarrerarekin erlazionatuko da. Horrela, pokemon talde bakoitzeko informazio gehiago izan dezakegu (mota, eboluzioa...)

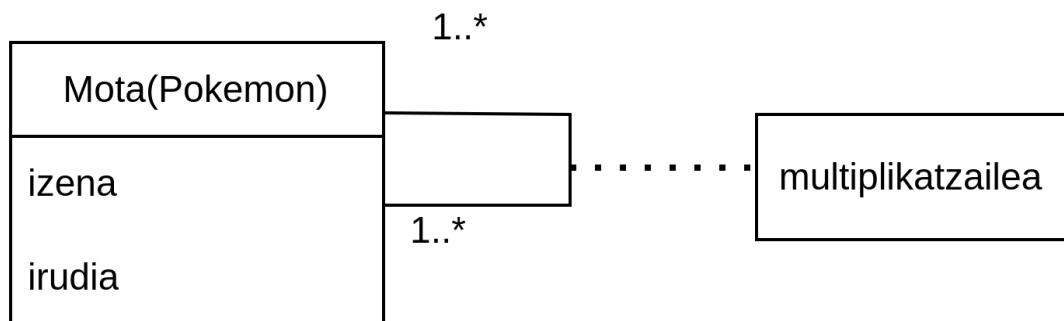


Irudia 118: Da (Pokemon (Talde) -> Pokemon (Pokedex)) erlazioaren diagrama

2.3.11. Eragina

Erlazio honen bitartez, mota bakoitza beste motekiko ze eragina duen gordetzeko erabiltzen da, horretarako pokemonaren eraso motaren arabera eta beste pokemonaren motaren araberako multiplikatzaile bat egongo da.

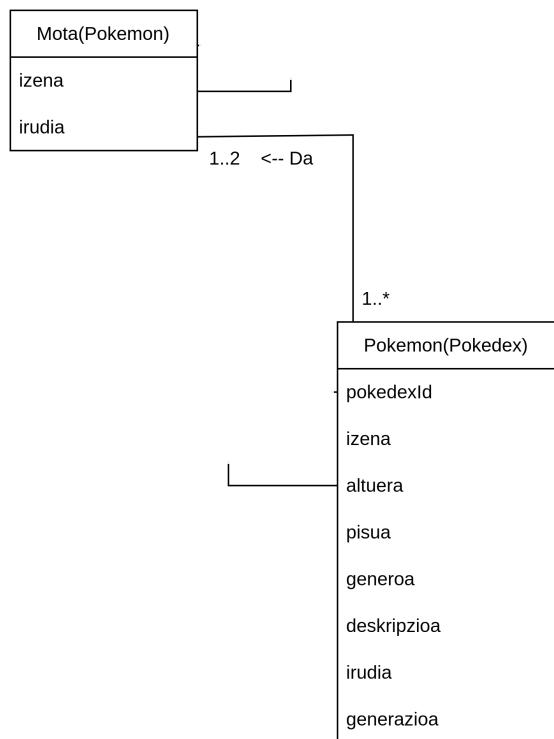
Kardinalitateari dagokionez, mota bakoitza eragina egiten dio pokemon bati gutxienez.



Irudia 119: Eragina erlazioaren diagrama

2.3.12. Da (Pokemon (Pokedex) -> Mota)

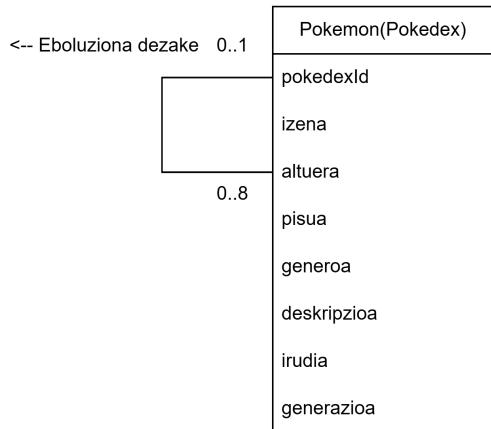
Pokedex-eko Pokemon-ak eta mota biltzen duten erlazioa da, honeta zehazten da Pokedex-eko Pokemon bakoitzak izango duen mota (Ura, Sua, Landare...). Pokemon bakoitzak gutxienez mota batekoa izango da eta gehienez 2 motakoa. Motaren kasuan, hainbat Pokemon ezberdinetan egon daiteke baina gutxienez bat.



Irudia 120: Da (Pokemon (Pokedex) -> Mota) erlazioaren diagrama

2.3.13. Eboluziona dezake

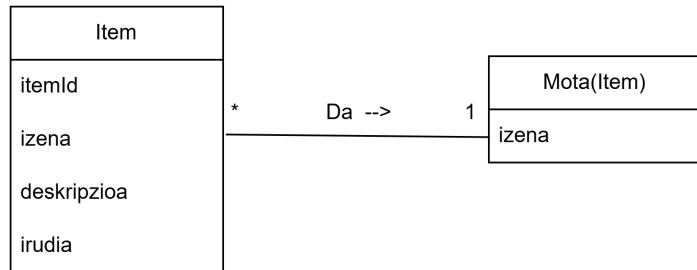
Pokemon espezie baten eboluzioen informazioa gordetzeko balio duen erlazioa. Pokemon(Pokedex) entitatea berekiko erlazonatzen da. Pokemon batek, eboluziorik ez izatea gerta daiteke (Kondairazko Pokemon-en kasuan adibidez) edo hainbat eboluzio euki ditzake (8 maximo, Eevee-ren kasuan).



Irudia 121: Eboluziona dezake erlazioaren diagrama

2.3.14. Da (Item -> Mota (Item))

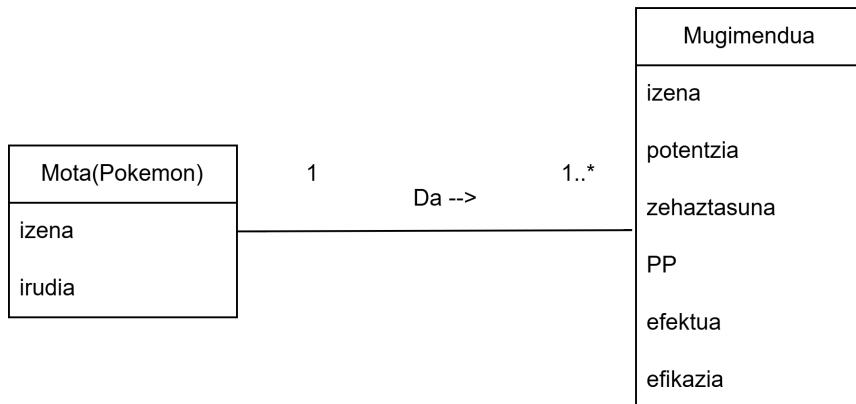
Item eta mota (item) entitateen arteko erlazio bat da non item bakoitza ze motatakoa den gordeko da. Kasu honetan, item bakoitza mota bakar bateko izango da bai ala bai (1 kardinalitatea). Aldi berean, existituko dira hainbat item mota berekoak (* kardinalitatea).



Irudia 122: Da (Item -> Mota (Item)) erlazioaren diagrama

2.3.15 Da (Mota (Pokemon) -> Mugimendua)

Erlazio honen bidez, mota (Pokemon) eta mugimendua entitateak erlazionatzen direnez, jakin dezakegu zein mugimendu bakoitza zein motatakoa den eta zeri egindo dion zehaztasun handiena. Haren kardinalitatea mugimendiaren aldetik (1..*) da zeren eta mota bakoitzeko hainbar mugimendu egon daitezke, eta motaren aldetik, 1 da mugimendu bakoitzak mota bakarra duelako.



Irudia 123: Da (Mota (Pokemon) -> Mugimendua) erlazioaren diagrama

3. Entregagarria (Software-aren probak)

3.1. Erabiltzaileen kudeaketa (Yassin Mehani)

Proba Kodea	Deskribapena	Espero den emaitza
3.1.1	Erabiltzaileak baliozko kredentzialak sartzen ditu eta " HASI SAIOA " botoiari eman.	Menu nagusira sartzen da.
3.1.2	Erabiltzaileak baliozko Izen bat ipini, baina pasahitza ez da egokia.	Ez du saioa hasiko.
3.1.3	Erabiltzaileak existitzen ez diren kredentzialak erabiltzen ditu.	Ez du saioa hasiko.
3.1.4	Erabiltzaileak hutsik utzi du eremu bat eta " HASI SAIOA " botoia sakatzen du.	Ez du saioa hasiko.
3.1.5	Erabiltzailea " ERREGISTRATU " botoiari ematen dio.	Erregistratzeko menura sartzen da.
3.1.6	Erabiltzaileak, karaktere espezialak dituen pasahitza ipintzen ditu eta " ERREGISTRATU " botoiari ematen dio.	"Pasahitza" eremuaren azpian errorea agertuko da, pasahitza karaktere bereziak dituela esanez.
3.1.7	Erabiltzaileak, pasahitz ezberdinak ipintzen ditu eta " ERREGISTRATU " botoiari ematen dio.	Errore bat agertuko da, pasahitzak ezberdinak direla esanez.
3.1.8	Erabiltzaileak balio ez duen posta elektroniko bat ipintzen du eta " ERREGISTRATU " botoiari ematen dio.	Errore bat agertuko da, posta baliozkoa ez dena esanez.
3.1.9	Erabiltzaileak balio ez duen jaiotze data bat ipintzen du eta " ERREGISTRATU " botoiari ematen dio.	"Jaiotze-data" eremuaren azpian, errore bat agertuko da, jaiotze-data baliozkoa ez dena esanez
3.1.10	Erabiltzaileak, eremu bat hutsik uzten du eta " ERREGISTRATU " botoiari ematen dio.	Eremu horren azpian errore mezua agertuko da.
3.1.11	Erabiltzaileak eremu guztietan baliozko datuak sartzen ditu eta " ERREGISTRATU " botoiari ematen dio.	Erabiltzailea erregistratuko da eta saioa hasteko pantailara itzuliko da.
3.1.12	Behin menu nagusian, Erabiltzaileak	Lagunak kudeatzeko menura

	“LAGUNAK” botoia sakatzen du.	sartu.
3.1.13	Erabiltzaileak “GEHITU” botoia saketzen du.	Entrenatzaile bat ID-aren bidez gehitzeko pantaila agertuko da.
3.1.14	Erabiltzaileak izena eremua hutsik uzten du eta “BILATU” botoiari ematen dio.	izena berriro sartzeko eskatuko dio erabiltzaileari.
3.1.15	Erabiltzaileak existitzen ez den baliozko izen bat sartzen du eta “BILATU” botoiari ematen dio.	“Entrenatzaile hori ez da existitzen” mezua agertuko da.
3.1.16	Erabiltzaileak bere lagun zerrenda jadanik dagoen entrenatzaile baten izena sartzen du eta “BILATU” botoiari ematen dio.	Entrenatzaile hori jadanik jarraitzen duzula agertuko da.
3.1.17	Erabiltzaileak bere lagun zerrenda ez dagoen entrenatzaile baten ID-a sartzen du eta “BILATU” botoiari ematen dio.	Entrenatzaile hori bere lagun zerrendan gehituko da.
3.1.18	Erabiltzaileak “HOME” botoiari ematen dio.	Menu nagusira bueltatuko da.
3.1.19	Erabiltzaileak “EZABATU” botoia sakatzen du.	Lagun bat ID-aren bidez ezabatzeko pantaila agertuko da.
3.1.20	Erabiltzaileak bere lagun zerrendako isilarazita ez dagoen entrenatzaile baten notifikazioak “ISILARAZI” botoiari ematen dio.	Entrenatzailearen notifikazioak isilaraziko ditu.
3.1.21	Erabiltzaileak bere lagun zerrendako isilarazita dagoen entrenatzaile baten notifikazioak “ISILARAZI” botoiari ematen dio.	Entrenatzailearen notifikazioak aktibatuko ditu.
3.1.22	Erabiltzaileak “HOME” botoiari ematen dio.	Menu nagusira bueltatuko da
3.1.23	Behin menu nagusian, Administratzaile den erabiltzaile bat “KUDEATU” botoia sakatzen du.	Erabiltzaileak kudeatzeko menura sartuko da.
3.1.24	Administratzaileak kontu baten “EDITATU” botoia sakatzen du.	Erabiltzaile horren datuak editatzeko pantailara sartuko da.
3.1.25	Administratzaileak eremuren bat	Eremu hori sartzeko eskatuko

	hutsik uzten du eta “ GORDE ” botoia sakatzen du.	dio administratzaileari
3.1.26	Administratzaileak “Pasahitza” eremuan karaktere bereziak ipintzen ditu eta “ GORDE ” botoia sakatzen du.	Pasahitza berriro sartzeko eskatuko dio.
3.1.27	Administratzaileak pasahitz ezberdinak ipintzen ditu eta “ GORDE ” botoia sakatzen du.	Pasahitzak ondo sartzeko eskatuko dio.
3.1.28	Administratzaileak existitzen den izen bat sartzen du (uneko erabiltzailearena ez dena) eta “ GORDE ” botoia sakatzen du.	“izena erabilgarri dago jada” mezua agertuko da.
3.1.29	Administratzaileak eremu guztiak betetezen ditu, posta elektroniko eta pasahitz egokiak sartzen ditu eta “ GORDE ” botoia sakatzen du.	Erabiltzailearen datuak eguneratuko dira.
3.1.30	Erabiltzaileak “ HOME ” botoiari ematen dio.	Menu nagusira bueltatuko da.
3.1.31	Administratzaileak, kontua ezabatzeko “ EZABATU ” botoia sakatzen du.	Kontua ezabatuko da.
3.1.32	Administratzaileak “ BILATZAILE ” eremua hutsik uzten du eta “ BILATU ” botoia sakatzen du.	Izen bat berriro sartzeko eskatuko dio.
3.1.33	Administratzaileak “ BILATZAILE ” eremuan existitzen ez den entrenatzaile baten izena ipintzen du eta “ BILATU ” botoia sakatzen du.	Erabiltzaileen zerrenda hutsik agertuko da.
3.1.34	Administratzaileak “ BILATZAILE ” eremuan existitzen den entrenatzaile baten izena ipintzen du eta “ BILATU ” botoia sakatzen du.	Erabiltzaileen zerrendan, izen hori duten entrenatzaileak agertuko dira.
3.1.35	Administratzaileak erabiltzaile bat administratzaile bihurtzeko botoia sakatzen du.	Erabiltzaile horrek administratzaile bihurtzen da.
3.1.36	Erabiltzaileak “ HOME ” botoiari ematen dio.	Menu nagusira bueltatuko da.
3.1.37	Erabiltzaileak “ EDITATU ” botoia sakatzen du.	Profiloa eguneratzeko pantaila aterako da.
3.1.38	Erabiltzaileak eremuren bat hutsik	Eremu hori sartzeko eskatuko

	uzten du eta “GORDE” botoia sakatzen du.	dio erabiltzaileari.
3.1.39	Erabiltzaileak “Pasahitza” eremuan karaktere bereziak ipintzen ditu eta “GORDE” botoia sakatzen du.	Pasahitza berriro sartzeko eskatuko dio.
3.1.40	Erabiltzaileak pasahitz ezberdinak ipintzen ditu eta “GORDE” botoia sakatzen du.	Pasahitzak ondo sartzeko eskatuko dio.
3.1.41	Erabiltzaileak existitzen den izen bat sartzen du (uneko erabiltzailearena ez dena) eta “GORDE” botoia sakatzen du.	“izena erabilgarri dago jada” mezua agertuko da.
3.1.42	Erabiltzaileak eremu guztiak betetzeen ditu, posta elektroniko eta pasahitz egokik sartzen ditu eta “GORDE” botoia sakatzen du.	Erabiltzailearen datuak eguneratuko dira.
3.1.43	Erabiltzaileak “HOME” botoiari ematen dio.	Menu nagusira bueltatuko da.

3.2. Pokemon taldeak sortzea (Unai Rodriguez)

Proba Kodea	Deskribapena	Espero den emaitza
3.2.1	Erabiltzailea “ EDIT ” botoia sakatzen du.	Hautatutako taldearen partaideak aldatzeko (gehitu edo ezabatu) menu berri bat agertuko zaio. (3.2.4-3.2.8)
3.2.2	Erabiltzailea “+” botoia sakatzen du eta 10 talde baino gutxiago ditu sortuta.	EDIT botoia sakatzerakoan menu berdina agertuko da baina kasu honetan taldea hutsik agertuko da. (talde berri bat sortzen ari zarelako) (3.2.4-3.2.8)
3.2.3	Erabiltzailea “+” botoia sakatzen du eta 10 talde ditu sortuta.	Errore mezua agerutko da erabiltzaile bakoitzak gehienez 10 talde izan ahal dituelako sortuta aldi berean.
3.2.4	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) “ PARTAIDE BAT ”-en (pokemon baten) argazkia sakatzen du.	Hautatutako pokemonarekin zer egin nahi duen erabiltzailea beste menu bat agertuko da. (3.2.9-3.2.10)
3.2.5	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) “+” botoia sakatzen du eta 6 pokemon baino gutxiago daude taldean.	Pokedex-aren menua agertuko da erabiltzaileak taldean sartu nahi duen pokemon-a hautatzeko (3.2.11-3.2.15)
3.2.6	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) “+” botoia sakatzen du eta 6 pokemon daude taldean.	Errore mezua agertuko da talde pokemon bakoitza gehienez 6 pokemon izan ahal dituelako.
3.2.7	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) “ EZABATU ” botoia sakatzen du.	Aukeratutako taldea sistematik ezabatuko da eta taldeen menu zerrendara bueltatuko da. (3.2.1-3.2.3)
3.2.8	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) “ GORDE ” botoia sakatzen du.	Aukeratutako taldea gordeko da aldaketa berriekin (ala ez) eta taldeen menu zerrendara bueltatuko da. (3.2.1-3.2.3)

3.2.9	Erabiltzaile batek pokemon bat aukeratzerakoan “ EZABATU ” botoia sakatzen du.	Hautatutako pokemon-a taldeik ezabatuko da eta taldeko menura bueltatuko da. (3.2.5-3.2.12)
3.2.10	Erabiltzaile batek pokemon bat aukeratzerakoan “ ALDATU ” botoia sakatzen du.	Pokedex-aren menua agertuko da pokemon hori ordezko bat bilatzeko. (3.2.11-3.2.15)
3.2.11	Pokedex menuaren barruan erabiltzailea “ POKEMON BAT ”-en argazkia sakatzen du.	Hautatutako pokemonaren laburpen txiki bat eta aukeratzeko aukera agertuko da beste menu batean. (3.2.19-3.2.20)
3.2.12	Pokedex menuaren barruan erabiltzailea bilatzailean pokemon baten izena idazten du eta “ ENTER ” botoia sakaten du.	Pokedex-aren zerrenda filtratuko da pokemon-aren izenaren arabera.
3.2.13	Pokedex menuaren barruan erabiltzailea bilatzailean pokemon baten izena idazten du (baina ez osorik) eta “ ENTER ” botoia sakatzen du.	Pokedex-aren zerrenda filtratuko da dagoeneko idatziakoaren arabera, kointzidentziarik ez badaude zerrenda hutsik agertuko da
3.2.14	Pokedex menuaren barruan erabiltzailea bilatzailean ez du ezer idatzen eta “ ENTER ” botoia sakatzen du.	Pokedex-aren zerrenda dagoen bezala geratuko da ez dagoelako ezer ez idatzita
3.2.15	Pokedex menuaren barruan erabiltzailea “ IRAGAZKETA ” botoia sakatzen du.	Pokedex-ean agertzen diren pokemon zerrenda filtratzeko aukerak agertuko dira. (3.2.16-3.2.18)
3.2.16	Iragazketa menuaren barruan erabiltzailea “ MOTAREN ARABERA ” botoia sakatzen du eta mota bat aukeratzen badu (kontuan izanda ez duela generaziorik aukeratu).	Pokedex-aren barruko pokemon zerrenda soilik motaren arabera filtratuko da.
3.2.17	Iragazketa menuaren barruan erabiltzailea “ GENERAZIOAREN ARABERA ” botoia sakatzen du eta generazio bat aukeratzen du (kontuan izanda ez duela motarik aukeratu).	Pokedex-aren barruko pokemon zerrenda soilik generazioaren arabera ordenatuko dira.
3.2.18	Iragazketa menuaren arabeta erabiltzaileak “ GENERAZIO BAT ETA	Pokedex-aren zerrenda generazio horretako moten

	MOTA BAT " aukeratzen du.	arabera filtratuko da. (Ez badaude kointzidentziarik zerrenda hutsik agertuko da)
3.2.19	Pokemon informazioaren barruan erabiltzailea " AUKERATU " botoia sakatzen du eta hautatutako pokemon-a ez dago taldean.	Hautatutako pokemon-a taldera sartuko da (ordezkapena edo berria) eta taldeko menura bueltatuko da (3.2.4-3.2.8)
3.2.20	Pokemon informazioaren barruan erabiltzailea " AUKERATU " botoia sakatzen du eta hautatutako pokemon-a taldean dago.	Errore mezua agertuko da erabiltzailea aukeratu nahi duen pokemon-a taldean dagoela jada esanez.
3.2.21	Erabiltzailea " HOME " botoia sakatzen du.	Erabiltzailea menu nagusira bueltatuko da. (3.1 puntuak)

3.3. ChangeLog (Eneko Puente)

Proba Kodea	Deskribapena	Espero den emaitza
3.3.1	Erabiltzaileak menuko “ JAKINARAZPENAK ” botoia sakatzen du.	Jakinazpenak pantailara aldatzea eta bertan dauden jakinazpenak ikusi ahal izatea.
3.3.2	Erabiltzaileak bilatzailean bilatu nahi duen erabiltzailearen izena erabiliz bilatzea. (Existitzen den izena)	Izen horri dagokion erabiltzailearen jakinazpenak pantailartuko dira.
3.3.3	Erabiltzaileak bilatzailean bilatu nahi duen erabiltzailearen izena erabiliz bilatzea. (Existitzen den izena, baina erabiltzaile bat baino gehiago agertzea)	Erabiltzaile guzti hauen jakinarazpenak agertzea dataren arabera ordenatuta.
3.3.4	Erabiltzaileak bilatzailean bilatu nahi duen erabiltzailearen izena erabiliz bilatzea. (Existitzen ez den izena)	ID-a berriz sartzeko eskatuko dio.
3.3.5	Bilatzailean letra bat edo letra para bat sartztea, hau da, izen partziala eta bilatu botoiari eman. (P edo Pa, Patxi izena bilatzekoan)	Letra horrekin edo izenaren parte partzialarekin dauden erabiltzaileak agertu behar dira.
3.3.6	Erabiltzaileak bilatzailea hutsik uzten du eta “ Bilatu ” botoia sakatzen du.	ID-a berriz sartzeko eskatuko dio.
3.3.7	Erabiltzaileak “ Menua ” botoia sakatzen du. (Jakinarazpenak barruan)	Menu nagusira bideratuko dio. (3.3.1)

3.4. Pokemon lista eta bilaketa (Asier Barrio)

Proba Kodea	Deskribapena	Espero den emaitza
3.4.1	Erabiltzaileak Pokemon baten argazkian sakatzen du.	Hautatutako Pokemon-aren gaitasunak eta eboluzioa erakutsiko dira.
3.4.2	Erabiltzaileak bilatzailean Pokemon baten izena sartzen du eta " Enter " sakatzen du (Pokemon-a existitzen da).	Idatzitako Pokemon-aren argazkia agertzen da zerrandan.
3.4.3	Erabiltzaileak bilatzailean Pokemon baten izena sartzen du eta " Enter " sakatzen du (Pokemon-a ez da existitzen).	Errore mezua agertzen da, Pokemon-a existitzen ez dela adieraziz.
3.4.4	Erabiltzaileak bilatzailea hutsik uzten du eta " Enter " sakatzen du (Ez du bilatzailearen gainean klik egiten, edo klik egiten du baina ez du ezer idatzi).	Pokemon guztien zerrenda agertzen da.
3.4.5	Erabiltzaileak bilatzailea hutsik uzten du eta " Enter " sakatzen du (Bilatzailean klik egin du eta hutsune bat jarri du).	Errore mezua agertuko da eta Pokemon lista osoa erakutsiko da.
3.4.6	Erabiltzaileak Pokemon baten hitz partziala sartzen du eta " Enter " sakatzen du (Char adibidez).	Idatzitako hitz partzialarekin bat datozen Pokemon-ak erakusten ditu. (Charmander , Charmeleon , Charizard , Charcadet... erakutsiko ditu)
3.4.7	Erabiltzaileak letra xehez edo letra larriz izen osoa idazten du bilatzailean eta Enter sakatzen du (CHARMANDER edo charmander , Charmander idatzi beharrean).	Sartutako Pokemon-aren izena erakutsiko du, nahiz eta letra larri edo letra xehe idatzi, emaitza bera agertuko da.
3.4.8	Erabiltzaileak Iragazki botoian sakatzen du.	Pokemon-ak filtrazeko aukerak erakutsiko dira. (3.4.9-3.4.15)
3.4.9	Erabiltzaileak Motaren Arabera iragazkia aukeratzen du eta bertan mota bat aukeratzen du (Generazio iragazketa ez du erabili).	Aukeratutako mota duten Pokemon-ak agertuko dira zerrandan.

3.4.10	Erabiltzaileak Generazioaren Arabera iragazkia aukeratzen du eta bertan generazio bat aukeratzen du (Mota iragazketa ez du erabili).	Aukeratutako generazioan parte diren Pokemon-ak agertuko dira zerrendan.
3.4.11	Erabiltzaileak Generazioaren Arabera eta Motaren Arabera iragazkian erabiltzen ditu, generazio konkretu bat eta mota konkretu bat aukeratzeko	Aukeratutako generazio parten diren eta mota hori duten Pokemon-ak agertuko dira zerrendan.
3.4.12	Erabiltzaileak iragazkiren bat (edo biak) erabiltzen d(it)u eta Pokemon baten izena idazten du bilatzailean eta “ Enter ” sakatzen du (Pokemon-a existitzen da).	Pokemon-a aukeratutako iragazkia betetzen badu zerrendan agertuko da, bestela hutsik erakutsiko da zerrenda.
3.4.13	Erabiltzaileak iragazkiren bat (edo biak) erabiltzen d(it)u eta Pokemon baten izena idazten du bilatzailean eta “ Enter ” sakatzen du (Pokemon-a ez da existitzen).	Errore mezua agertzen da, Pokemon-a existitzen ez dela adieraziz.
3.4.14	Erabiltzaileak iragazkiren bat (edo biak) erabiltzen d(it)u eta Pokemon baten hitz partziala idazten du bilatzailean eta “ Enter ” sakatzen du.	Idatzitako hitz partzialarekin bat datozten Pokemon-ak iragazkia betetzen badute erakusten ditu.
3.4.15	Erabiltzaileak Motaren Arabera iragazkia aukeratzen du eta bertan mota bat aukeratzen du (Generazio iragazketa ez du erabili).	Aukeratutako mota duten Pokemon-ak agertuko dira zerrendan.
3.4.16	Edozein momentutan erabiltzaileak “ Home ” botoian sakatzen du.	Menu nagusira itzultzen da.

3.5. ChatBot (Yoel Justel)

Proba Kodea	Deskribapena	Espero den emaitza
3.5.1	Erabiltzaileak “ CHATBOT ” botoia sakatzen du menuan.	ChatBot pantaila irekitzen da.
3.5.2	Erabiltzaileak “ MUGUMENDUAK ” botoia sakatzen du.	Pokemon guztien zerrenda agertuko da. (3.5.2.X)
3.5.3	Erabiltzaileak “ ONENAK ” botoia sakatzen du eta talde bat gutxienez lehendik sortuta dauka.	Pokemon taldeen zerrenda agertuko da. (3.5.3.X)
3.5.4	Erabiltzaileak “ INDARRAK ” botoia sakatzen du.	Pokemon guztien zerrenda agertuko da. (3.5.4.X)
3.5.5	Erabiltzaileak “ EBOLUZIOA ” botoia sakatzen du.	Pokemon guztien zerrenda agertuko da. (3.5.5.X)
3.5.6	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.
3.5.2.1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.
3.5.2.2	Erabiltzaileak Pokemon baten ikonoa sakatzen du.	Pokemon horrek ikasi dezakeen mugimendu posible guztiak, horren argazkia eta izena pantailaratuko dira.
3.5.2.2.1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.
3.5.3.1	Erabiltzaileak “ ONENAK ” botoia sakatzen du eta ez dauka gutxienez talde bat lehendik sortuta.	“Lehenik, taldea sortu” mezua agertuko da bistan.
3.5.3.2	Erabiltzaileak “ ONENAK ” botoia sakatzen du eta taldeak sortuta.	Taldeen zerrenda agertuko da.
3.5.3.2.1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.

3.5.3.3	Erabiltzaileak talde bat sakatzen badu eta talde honek ez du pokemonik.	“Talde honek ez du pokemonik” mezua agertuko da bistan.
3.5.3.3.1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.
3.5.3.4	Erabiltzaileak talde bat sakatzen badu eta talde honek badu pokemonia.	Taldeko Pokemon hoberenaren izena, gaitasunak, estatistikak eta argazkia pantailaratzen dira.
3.5.4.1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.
3.5.4.2	Erabiltzaileak Pokemon baten ikonoa sakatzen du. (Mota bakarrekoa)	Pokemon horrek noren aurka bere elementua den ahula eta noren aurka indartsua, horren argazkia eta izena pantailaratuko dira.
3.5.4.2.1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.
3.5.4.3	Erabiltzaileak Pokemon baten ikonoa sakatzen du. (Bi motakoa)	Pokemon horrek noren aurka bere elementua den ahula eta noren aurka indartsua, horren argazkia eta izena pantailaratuko dira.
3.5.5.1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.
3.5.5.2	Erabiltzaileak Pokemon baten ikonoa sakatzen du (Eboluzio katea duena).	Pokemon horrek daukan eboluzio katea agertuko da, Pokemon-en izena eta irudiarekin batera.
3.5.5.2.1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.
3.5.5.3	Erabiltzaileak Pokemon baten ikonoa sakatzen du (Eboluzio katea ez duena).	“Eboluziorik ez” mezua pantailatzen da.

3.6. ItemDex (Xinyan Wang)

Proba Kodea	Deskribapena	Espero den emaitza
3.6.1	Erabiltzaileak " ITEMDEX " botoia sakatzen du menuan.	ItemDex pantaila irekitzen da. (3.6.2-3.6.18)
3.6.1.1	Erabiltzaileak edozein momentutan " Home " botoia sakatzen du.	Menu nagusira itzultzen da.
3.6.2	Erabiltzaileak " Iragazketa " botoia sakatzen du.	Iragazketa aukerak agertzen dira. (Motaren arabera eta alfabetikoki).
3.6.3	Erabiltzaileak edozein momentutan " Home " botoia sakatzen du.	Menu nagusira itzultzen da.
3.6.4	Erabiltzaileak item baten izena idazten du eta " Enter " sakatzen du (itema ez da existitzen).	"Itema ez da existitzen" errore mezua agertzen da.
3.6.5	Erabiltzaileak hutsik utzi du bilaketa eremua eta "Enter" sakatzen du.	Zerrenda osoa erakusten du.
3.6.6	Erabiltzaileak zenbakiak sartzen ditu bilaketan.	<p>Item batzuek izen baten barruan zenbaki bat izango dute. Item horientzat, zenbaki hori duten itemen zerrenda erakutsiko da.</p> <p>Baina zenbaki luze bat bada, "Itema ez da existitzen" errore mezua agertzen da.</p>
3.6.7	Erabiltzaileak karaktere bereziak sartzen ditu bilaketan (adibidez: @, #, \$).	<p>Karaktere berezi batzuentzat, "Itema ez da existitzen" errore mezua agertzen da.</p> <p>Baina, adibidez "-", ager daiteke, izan ere, badira izenean karaktere hori duten itemak.</p>
3.6.8	Erabiltzaileak hitz partziala sartzen du bilaketan (adibidez: "ball" "Super Ball" bilatzeko).	Hitz partziala horrekin bat datorren item guztiak erakusten dira.

3.6.9	Erabiltzaileak maiuskulak /minuskulak erabiltzen ditu bilaketan (adibidez: "PASTA" vs "pasta").	Bilaketa maiuskula/minuskula sentikorra ez dela egiaztatzen (emaitza bera itzuli behar da).
3.6.10	Erabiltzaileak espazio gehiago sartzen ditu hasieran/amaieran bilaketan (adibidez: " pasta ").	Sistemak espazioak ezabatu behar ditu eta emaitza egokia erakutsi.
3.6.11	Erabiltzaileak luzeegia den bilaketa bat sartzen du (100 karaktere baino gehiago).	Sistemak ez du ezer agertuko.
3.6.12	Erabiltzaileak iragazkiak aplikatzen ditu. "Motaren arabera" klikatzen du, eta motaren bat aukeratzen du. Adibidez, <i>abono</i> edo <i>caja de bayas</i> aukeratuz.	Iragazketa horri dagokion elementu-zerrenda agertuko da.
3.6.13	Erabiltzaileak iragazkiak kentzen ditu.	Iragazketa kentzen da eta zerrenda osoa berriro erakusten da.
3.6.14	Erabiltzaileak ordenaketa aldaketa egiten du (alfabetikoki).	Uneko zerrenda ordenatuta agertzen da (A-tik Z-ra).
3.6.15	Erabiltzaileak mota bat aukeratzen du eta, gainera, item baten izena idazten du bilatzailean.	Mota horretan, izena betetzen duen item zerrenda agertuko da.
3.6.16	Erabiltzaileak item bat aukeratu eta gero "Itzuli" sakatzen du.	ItemDex pantailara itzultzen da.
3.6.17	Erabiltzaileak item bat bilatu eta gero beste bat bilatzen du.	Bigarren bilaketa egokia egiten da, lehenengoaren emaitzak ezabaturik.
3.6.18	Erabiltzaileak item baten gainean klik egiten du xehetasun pantailan.	Itemaren deskribapena, ikonoa eta erabilpena erakusten dituen leihoa berria irekitzen da.
3.6.18.1	Erabiltzaileak edozein momentutan " Home " botoia sakatzen du.	Menu nagusira itzultzen da.

3.7. AI-ren erabilpena

Xinyan:

<https://chat.deepseek.com/share/vnfbyscres3h89mznw>

Deepseek-i bidalitako informazioa honakoa da: probakasuetako eskakizunak, Egela-ko TEMA7 eta funtzionalitate horri dagozkion Gertaera fluxua. Ondoren, berari eskatu zitzzion proba-kasuak emateko. Galdera txineraz idatzita zegoenez, hemen dago haren euskarazko itzulpena:

“Goiko hori irakaslearen eskakizuna da; nire funtzionalitatea itemdex da. Irakaslearen eskakizunen eta PDFko txantiloien arabera, proba osatzen lagun diezadazun nahi dut, caja negra-koa bakarrik behar dut.

Hau da funtzionalitate honi buruzko gertaera-fluxua.”

Halaber, Deepseek-en lehen erantzunaren lehen paragrafoa ere txineraz dago, eta hori ere hemen dago itzulita:

“Zure emandako funtzionalitatearen deskribapena eta irakaslearen eskakizunak kontuan hartuta, ItemDex funtzionalitatearentzako caja negra probetarako plana (Proba Plana) sortu dut, euskaraz idatzita eta PDFeko txantiloien formatua jarraituz.”

Orokorrean, proba-kasu gehienak haren emandako txantiloien gainean daude idatziak; batzuk aldaketarik gabe daude, zuzenean kopiatuak. Adibidez: “ITEM_11: Erabiltzaileak hutsik uzten du bilaketa eremua eta ‘Enter’ sakatzen du.” Hori gertatu daitekeen egoera da, eta sistema horrelako kasuak onartzen ditu.

Hala ere, badaude aldaketa txiki batzuk. Adibidez, Deepseek-en “ITEM_08: Erabiltzaileak iragazkiak aplikatzen ditu (mota, alfabetikoa, bitxitasuna).” Kasu honetan, mota-aren arabera bakarrik sailkatzen da; bitxitasuna ez dago, eta alfabetikoa ez da sailkapena, zerrenda A-tik Z-raino ordenatzea baizik. Beste adibide bat da “ITEM_12: Erabiltzaileak karaktere bereziak sartzen ditu bilaketan (adibidez: @, #, \$).” Kasu honetan, sinbolo horiek ez dira onartzen; bakarra onartzen da: -, izan ere badira item izenetan hori duten kasuak, adibidez power-belt.

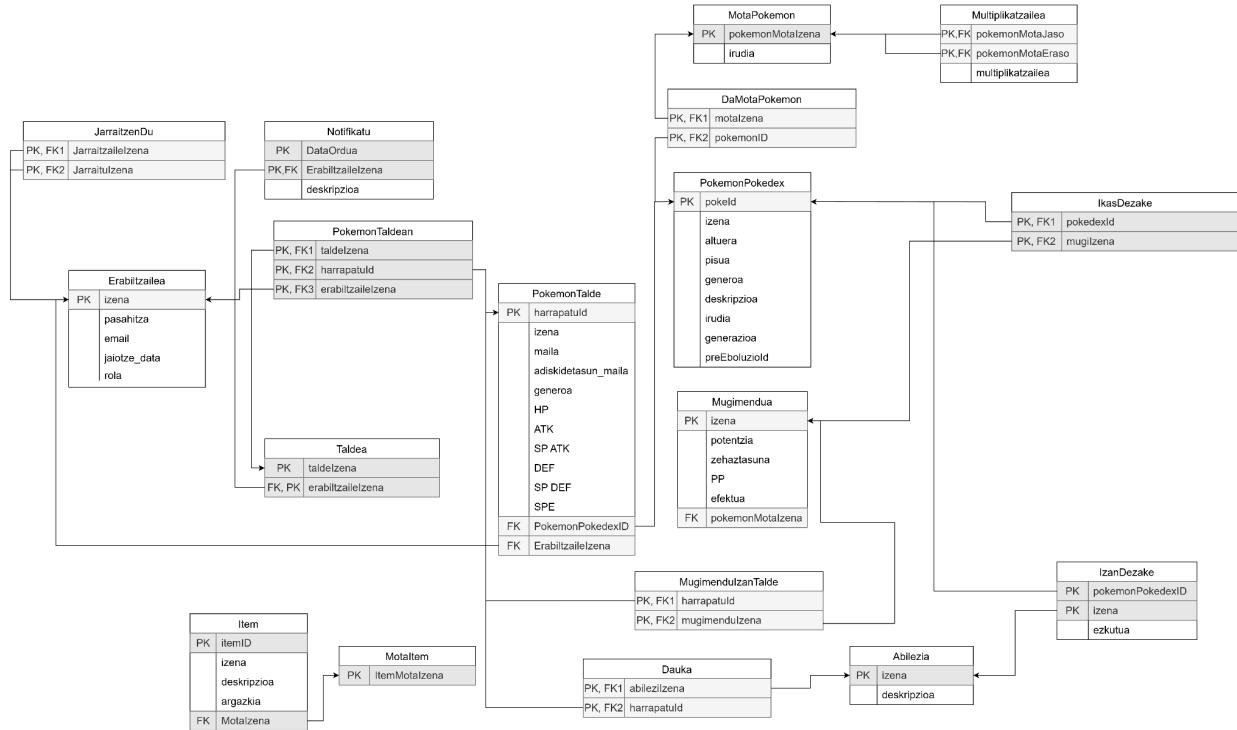
Badira baita sartu ez diren batzuk ere, adibidez “Item_34: Pantaila tamaina txikia duen gailu batean (mobile).” Hori ez da funtzionalitate honen barruan kontuan hartu beharrekoa.

Laburbilduz, Deepseek-en hasierako erantzuna logikoa da informazio zehatza jasotzen duenean, baina ondorengo zatieta logika falta daiteke. Horregatik, ez da gomendagarria erantzunak zuzenean kopiatzea; komeni da erantzun bakoitzak irakurtzea eta arretaz baloratzea ea logikoa eta beharrezkoa den.

4. Entregagarria (Klase diagrama, datu basea eta komunikazio diagrama)

4.1 Datu basea

Datu basea domeinu eredutik abiatuta sortu egin da. Eraldaketa hori egiteko, ikasgelan erakutsi dizkiguten urratsak jarraitu ditugu, alegia, domeinu-ereduaren diseinua datu-basera nola aldatu. Hona hemen emaitza:



Irudia 124: Datu base erlazionala

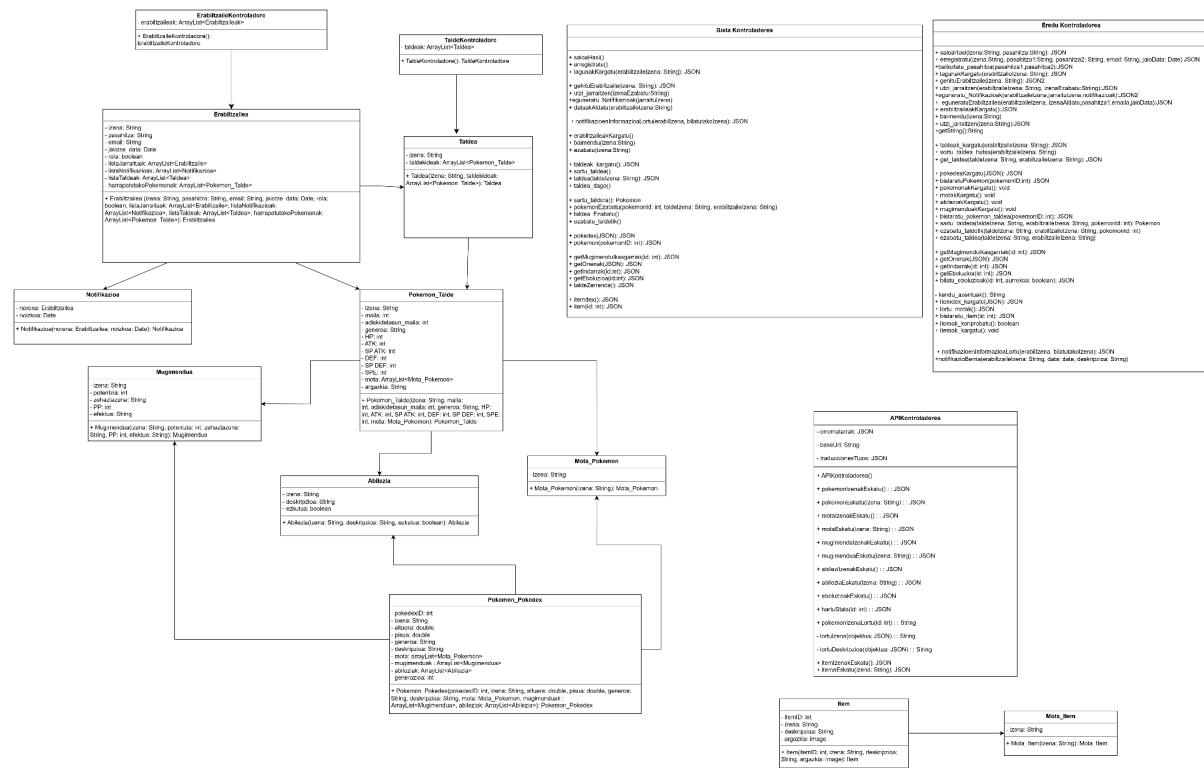
4.1.1 Azalpen gehigarriak

Txarto ikustekotan, link honen bidez hobeto ikusi daiteke:

https://drive.google.com/file/d/1uPf_9o_7SX3pYIIXeBq8n8lYO5NQLnR2/view?usp=ssharing

4.2 Klase diagrama

Klase-diagrama domeinu-eredutik abiatuta sortu da. Eraldaketa egitean, domeinu-ereduan zeuden entitate asko klase bihurtu dira, beren atributuekin eta funtziekin. Gainera, diseinu-logikari eusteko, klase gehigarri batzuk sortu behar izan dira. Hona hemen klase diagrama osoaren irudia:



4.2.1 Azalpen gehigarriak

Txarto ikustekotan, link honen bidez hobeto ikusi daiteke:

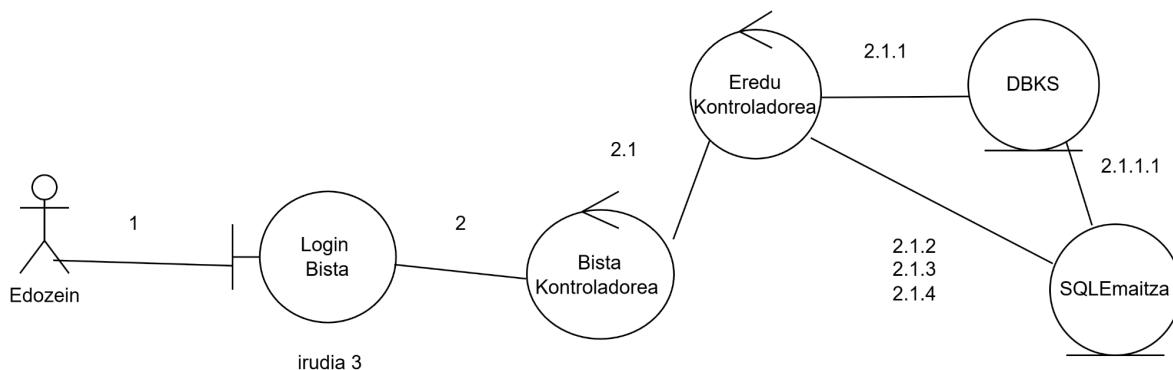
<https://drive.google.com/file/d/1oItAZOq-dn14YxGonwSU-ZmPwl92kLRP/view?usp=sharing>

4.3 Komunikazio diagrama

Komunikazio diagrama burutzeko, atal jakin bat erabiltzeko kasuak egitea zegokien pertsona berak arduratuko dira komunikazio-diagraman hori egiteaz. Hona hemen beharrezko den atal bakoitzaren komunikazio diagrama egilearen izenarekin batera.

4.3.1 Erabiltzaileen kudeaketa (Eneko Puente)

4.3.1.1 Saioa Hasi



Irudia 126: Saioa hasi funtzionalitatearen komunikazio diagrama

1: Erabiltzaileak bere izena eta pasahitza sartzen ditu eta "Saioa Hasi" botoia sakatzen du.

2:SaioaHasi()

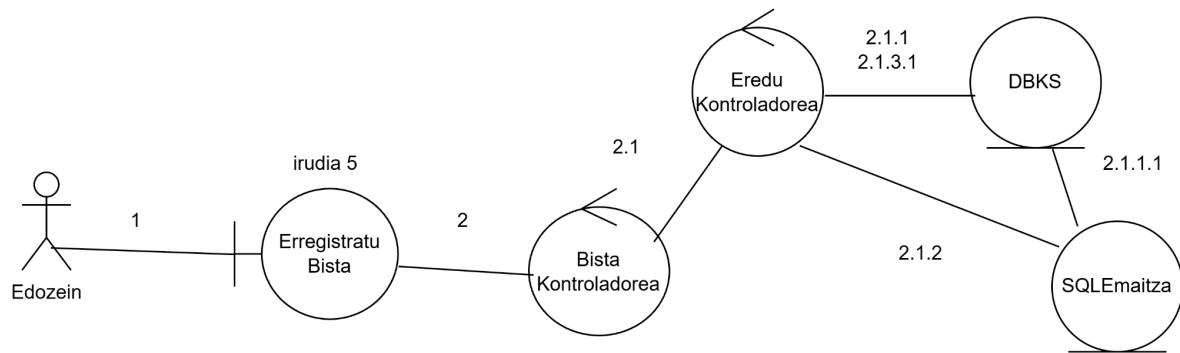
- 2.1: saioaHasi(izena, pasahitza):JSON1
 - 2.1.1: execSQL(SQL1): EmaitzSQL
 - 2.1.1.1: new EmaitzSQL
[Emaitz ez null bada]
 - 2.1.2: next()
 - 2.1.3: getString("pasahitza"):String
 - 2.1.4: getString("Administratzalea"):String

SQL1: SELECT izena,pasahitza, rol
FROM Erabiltzaile
WHERE E.izena=%izena%

JSON1:

[{Ondo:Boolean, ErabiltzaileIzena:String,Rola:String,mezua:String},]

4.3.1.2 Erregistroatu



Irudia 127: Erregistroatu funtzionalitatearen komunikazio diagrama

1: Erabiltzaileak bere datuak (Izena, pasahitza, Email-a, Jaiotza-Data) sartzen ditu eta "Erregistroatu" botoia sakatzen du.

2:Erregistroatu()

2.1:Erregistroatu(izena, pasahitza1, pasahitza2, Email, JaioData):JSON2

2.1.1: execSQL(SQL2): EmaitzaSQL

2.1.1.1: new EmaitzaSQL

2.1.2: next()

[Emaitza null]

Eedu Kontroladoreko balioztatu_pasahitza(pasahitza1, pasahitza2):JSON2

metodoari deitzen zaio

[True]

2.1.3.1a: execSQL(SQL3)

SQL2: SELECT E.izena

FROM Erabiltzaile E

WHERE E.izena=%izena%

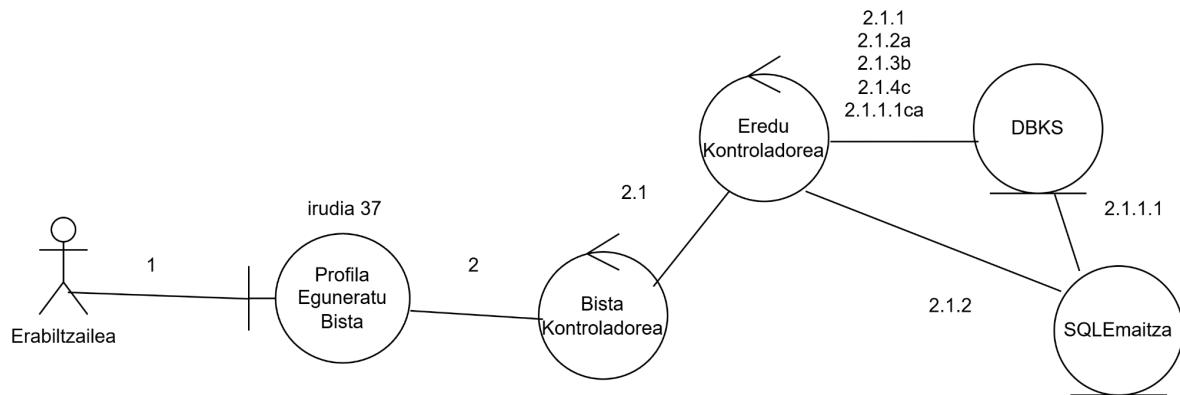
SQL3: INSERT INTO Erabiltzaile (Izena, Pasahitza, Email, JaioData, Rola)

VALUES (%Izena%, %Pasahitza%, %Emai%, %JaioData%,%rola%)

JSON2:

[{Ondo:Boolean, mezua:String},]

4.3.1.3 Profila Egumeratu



Irudia 128: Profila egueratu funtzionalitatearen komunikazio diagrama

1: "Editatu" botoia sakatzen du (bakarrik aldatzen dira sartutako datuak, besteak zeuden moduan gelditu dira)

2: datuakAldatu(erabiltzailezena)

2.1: eguneratuErabiltzailea(erabiltzailezena, izenaAldatu,pasahitzal, emaila,jaioData):JSON

2.1.1: execSQL(SQL4)
[izenaAldatu==null]

2.1.2a: execSQL(SQL5)
[email==null]

2.1.3b: execSQL(SQL6)
[jaioData==null]

2.1.4c: execSQL(SQL7)
[izenaAldatu!=erabiltzailezena]
2.1.1ca: execSQL(SQL8)

SQL4:SELECT izena, pasahitzza, rola FROM Erabiltzailea WHERE izena = %izena%

SQL5:SELECT email FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL6:SELECT jaiotze_data FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL7:SELECT izena FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL8: UPDATE Erabiltzaile

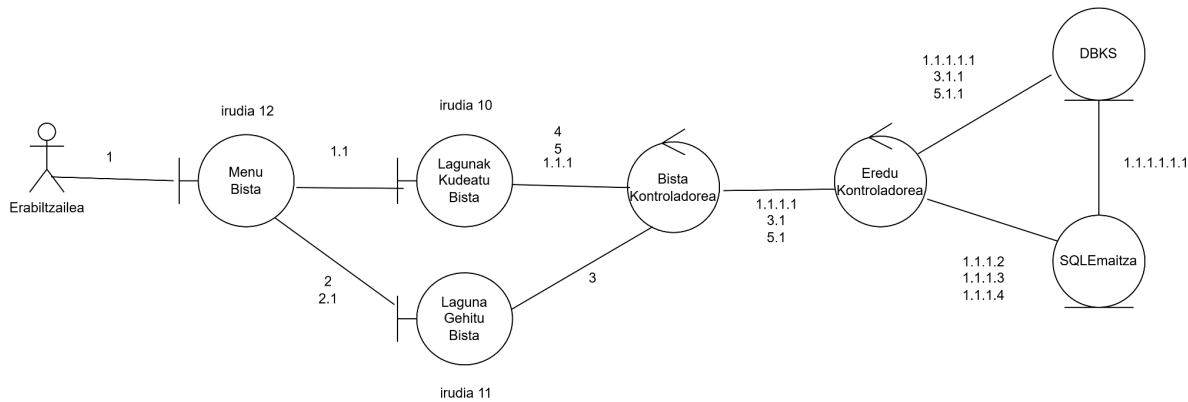
SET izena=%izenaAldatu%,
pasahitzza=%pasahitzal%,
email=%email%, jaiotze_data=%JaioData%

WHERE izena=%erabiltzailezena%

JSON: [{Ondo:Boolean, mezu:string},]

(JSON hau erabiltzen gero erabiltzailea behar den lekura bueltatzeko, izan ere, kudeaketa atalerako eta editatu atalerako atal bera erabiltzen.)

4.3.1.4 Lagun Zerrenda



Irudia 129: Lagun zerrenda funtzionalitatearen komunikazio diagrama

1: Menuko "LagunakKudeatu" botoia sakatzen du.

1.1: new LagunakKudeatu()

- 1.1.1: lagunakKargatu(erabiltzailezena: String):JSON1
- 1.1.1.1: lagunakKargatu(erabiltzailezena: String):JSON1
- 1.1.1.1.1: execSQL(SQL4): EmaitzaSQL
- 1.1.1.1.1.1: new EmaitzaSQL()
- 1.1.1.2: next()
- 1.1.1.3: getString("izena"):String
- 1.1.1.4: getBoolean("notifikazioak"):Boolean

2a:"Gehitu" botoia sakatzen du

2.1a: new LagunaGehitu()

3a:gehituErabiltzailea(plzena):JSON2

- 3.1a gehituErabiltzailea(erabiltzailezena: string, izena:String):JSON2
- [Erabiltzailea existitzne bada eta jarraitu botoiari ematen radio]
- 3.1.1ab:execSQL(SQL6)

4b: "Ezabatu" botoia sakatzen du

5b: utzi_jarraitzen(plzenaEzabatu): JSON3

- 5.1b: utzi_jarraitzen(erabiltzailezena:String, izenaEzabatu): JSON2
- 5.1.1b execSQL(SQL7)

6c: "Notifikazioak kendu" botoia sakatzen du

7c:eguneratu_Notifikazioak(jarraitulzena)

7.1c:eguneratu_Notifikazioak(erabiltzailezena,jarraitulzena,notifikazioak):JSON2

7.1.1c: execSQL(SQL8)

SQL4: SELECT Jarraitulzena, Notifikatu

FROM JarraitzenDu

WHERE Jarraitzailezena = %erabiltzailezena%

ORDER BY Jarraitulzena

SQL6: INSERT INTO JarraitzenDu (JarraitzaileZena, Jarraitulzena)
VALUES (%erabiltzaileZena%, %IzenaGehitu%)

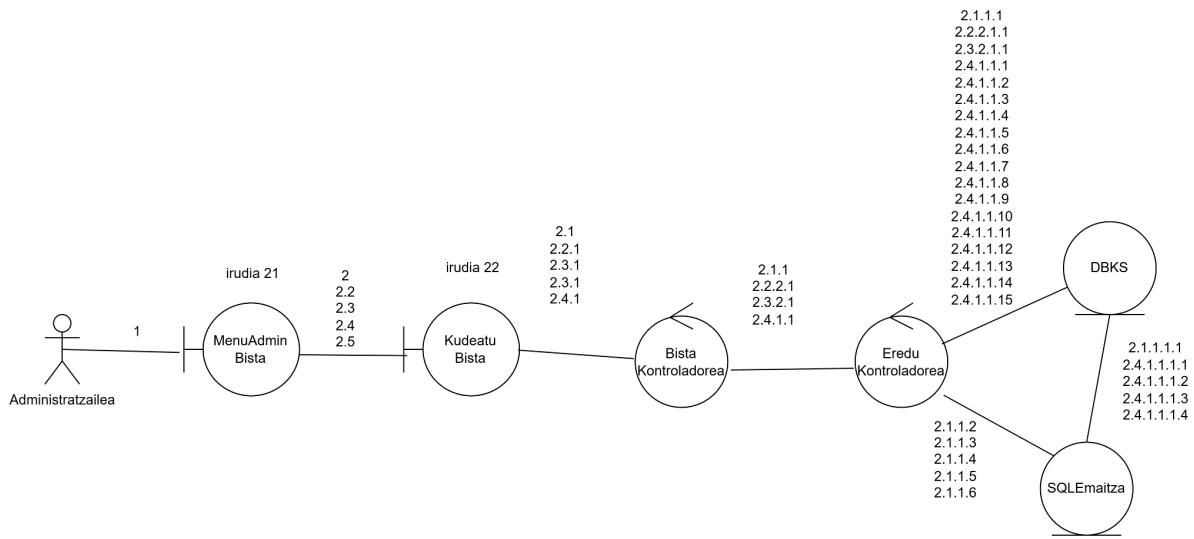
SQL7: DELETE FROM JarraitzenDu
WHERE JarraitzaileZena=%erabiltzaileZena% And
Jarraitulzena=%plzenaEzabau%

SQL8: UPDATE JarraitzenDu SET Notifikatu = %notifikazioak% WHERE
JarraitzaileZena = %erabiltzaileZena% AND Jarraitulzena = %jarraitulzena%

JSON1: {[{izena:string, notifikazioak:boolean},{}]}

JSON2: {[{ondo:boolean, mezua: string},{}]}

4.3.1.5 Menu Admin



Irudia 130: Menu Admin funtzionalitatearen komunikazio diagrama

1: Administratzaileak "Kudeatu" botoia sakatzen du

2: new Kudeatu()

2.1: erabiltzaileakKargatu()

2.1.1: erabiltzaileak Kargatu():JSON1

1.1.1: execSQL(SQL1): SQL Emaitza

2.1.1.1.

2.1.1.2: next()

2.1.1.3: `getString("izena")`: String

2.1.1.4: getString("email"): S

2.1.1.5: `getString("rola")`: String

2.1.1.6: getString("jaiotze")

Ezabatu" botoia sakat.

ezabatu(izena)

a: ezabatu(izena)

2.2.2.1.1a: execSQL

Baimendu" botoia sakat

baimendu(izena)

o: baimendu(izena)

2.3.2.1.1b: execSQL(SQL3)

ak zeuden moduan gelditu dira)

2.4.1c: eguneratuErabiltzailea(erabiltzailezena, erabiltzailearen izena)

```
        emaila,jaioData):JSON  
    2.4.1.1.1c: execSQL(SQL4)  
        2.4.1.1.1c: new SQLEmaitza()  
    2.4.1.1.2c:next()  
    2.4.1.1.3c:getString("izena"):String  
    2.4.1.1.4c:getString("pasahitza"):String  
    2.4.1.1.5c:getString("rola"):String  
    2.4.1.1.6c:execSQL(SQL5)  
        2.4.1.1.1.2c: new SQLEmaitza()  
    2.4.1.1.7c:next()  
    2.4.1.1.8c:getString("email"):String  
    2.4.1.1.9c: execSQL(SQL6)  
        2.4.1.1.1.3c: new SQLEmaitza()  
    2.4.1.1.10c:next()  
    2.4.1.1.11c:getString("jaioData"):String  
    2.4.1.1.12c: execSQL(SQL7)  
        2.4.1.1.1.4c: new SQLEmaitza()  
    2.4.1.1.13c:next()  
    2.4.1.1.14c:getString("izena"):String  
    2.4.1.1.15c: execSQL(SQL8)
```

2.5d: "Bilatzailean izen bat sartzen du"

(Hemen aurretik kargatutako erabiltzaileen gainean bilaketa bat egiten du, usersFilters() erabiliz.

SQL1: SELECT Izena, Rola FROM Erabiltzaile

SQL2: DELETE FROM Erabiltzaile WHERE izena=%izena%

SQL3: UPDATE Erabiltzaile

```
    SET rola="admin"  
    WHERE izena=%izena%
```

SQL4:SELECT izena, pasahitza, rola FROM Erabiltzailea WHERE izena = %izena%

SQL5:SELECT email FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL6:SELECT jaiotze_data FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL7:SELECT izena FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL8: UPDATE UPDATE Erabiltzaile

```
    SET izena=%izenaAldatu%,  
    pasahitza=%pasahitza1%,  
    email=%email%, jaiotze_data=%JaioData%  
    WHERE izena=%erabiltzailezena%
```

SQL8:

JSON1: {[erabiltzaileLista,]}

JSON: [{Ondo:Boolean, mezu:string},]

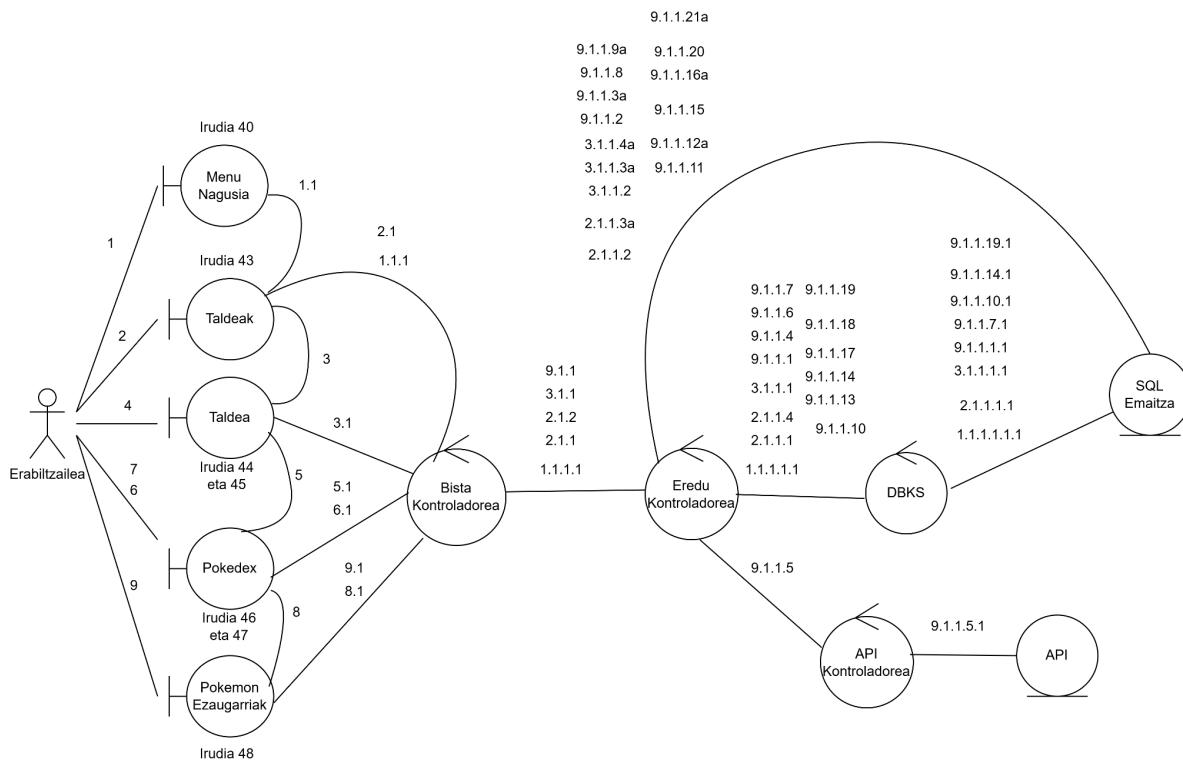
(JSON (JSON) hau erabiltzen gero erabiltzailea behar den lekura bueltatzeko, izan ere, kudeaketa atalerako eta editatu atalerako atal bera erabiltzen.)

2.5: Bilatzailean izen bat sartzen du

(Atal hau zuzenean egiten bilaketa lehenik igo diren erabiltzaileen gainean, filterUsers() funtzioa erabiliz)

4.3.2 Pokemon taldeak sortzea (Asier Barrio)

4.3.2.1 Talde berria sortu eta Pokemon sartu



Irudia 131: Talde berria sortu eta Pokemon sartu funtzionalitateen komunikazio diagrama

1- Erabiltzaileak Menu Nagusian "Taldeak" botoia sakatzen du

1.1- new Taldeak()

1.1.1- taldeak_kargatu(): JSON4

1.1.1.1- taldeak_kargatu(erabiltzailezena): JSON4

1.1.1.1.1- execSQL(sql_taldeak): SQL Ematiza

1.1.1.1.1.1- new SQL Emaitza()

1.1.1.1.2- next(): boolean

[next() true den bitartean]

1.1.1.1.3a- `getString("taldelzena")`: String

2- Erabiltzailea "+" botoia sakatzen du.

2.1- sortu_taldea()

2.1.1- sortu_taldea_hutsa(erabiltzailezena)

2.1.1.1- execSQL(sql_taldeak): SQL Emaitza

2.1.1.1.1- new SQL Emaitz()

2.1.1.2- next(): boolean

[next() true den bitartean]

2.1.1.3a- `getString("taldeIzena")`: String

2.1.1.4- execSQL(sql_sortu)

2.1.2- notifikazioBerria(erabiltzailelzena, data, deskripzioa)

[Erabiltzailea Gorde edo Ezabatu klikatzen ez duen bitartean hurrengoa errepikatu]

3- new Taldea()

3.1- taldea(taldezena): JSON3

3.1.1- get_taldea(taldezena, erabiltzailelzena): JSON3

3.1.1.1- execSQL(sql_poketalde): SQL Emaitza

3.1.1.1.1- new SQL Emaitza()

3.1.1.2- next(): boolean

[next() true den bitartean]

3.1.1.3a- getInt("pokeld"): Int

3.1.1.4a- getString("argazkia"): String

4- Erabiltzailea "+" botoia sakatzen du.

5- new Pokedex()

5.1- pokedexKargatu(JSON1): JSON2

[Erabiltzaileak filtroak ezartzen baditu]

6- Filtroa aukeratu eta bilaketa sakatzen du

6.1- pokedexKargatu(JSON1): JSON2

7- Erabiltzaileak Pokemon baten argazkian klik egiten du.

8- new PokemonEzaugarriak(pokemonID)

8.1- bistaratuPokemon(pokemonID): JSON7

9- Erabiltzaileak "Hautatu" botoia sakatzen du.

9.1- sartu_taldera()

9.1.1- sartu_taldera(taldezena, erabiltzailelzena, pokemon_id)

9.1.1.1- execSQL(sql_tamaina): SQL Emaitza

9.1.1.1.1- new SQL Emaitza()

9.1.1.2- next(): boolean

[next() true den bitartean]

9.1.1.3a- getInt("count"): Int

9.1.1.4- execSQL(sql_generoa)

9.1.1.5- hartu_stats(pokemonId): JSON9

9.1.1.5.1- pokemon(pokemonId): JSON9

9.1.1.6- execSQL(sql_harrapatu)

9.1.1.7- execSQL(sql_azkenharrapatu): SQL Emaitza

9.1.1.7.1- new SQL Emaitza()

9.1.1.8- next(): boolean

[next() true den bitartean]

9.1.1.9a- getInt("id"): Int

9.1.1.10- execSQL(sql_abilezi): SQL Emaitza

9.1.1.10.1- new SQL Emaitza()

9.1.1.11- next(): boolean

[next() true den bitartean]

- 9.1.1.12a- getString("izena"): string
- 9.1.1.13- execSQL(sql_dauka)
- 9.1.1.14- execSQL(sql_mugi): SQL Emaitza
- 9.1.1.14.1- new SQL Emaitza()
- 9.1.1.15- next(): boolean

[next() true den bitartean]

- 9.1.1.16a- getString("mugilzena"): string
- 9.1.1.17- execSQL(sql_mugimenduak)
- 9.1.1.18- execSQL(sql_taldera)
- 9.1.1.19- execSQL(sql_izena): SQL Emaitza
- 9.1.1.19.1- new SQL Emaitza()

9.1.1.20- next(): boolean

[next() true den bitartean]

- 9.1.1.21a- getString("izena"): string

sql_taldeak: SELECT taldezena
FROM Taldea
WHERE erabiltzailelzena = %erabiltzailelzena%

sql_sortu: INSERT INTO
Taldea (taldelzena, erabiltzailelzena)
VALUES (%taldelzena%,%erabiltzailelzena%)

sql_poketalde: SELECT PT.harrapatuld, PP.irudia
FROM Taldea T
JOIN PokemonTaldean PT ON T.taldelzena = PT.taldelzena
JOIN PokemonTalde PKT ON PT.harrapatuld = PKT.harrapatuld
JOIN PokemonPokedex PP ON PKT.PokemonPokedexID = PP.pokId
WHERE T.erabiltzailelzena = %erabiltzailelzena% AND T.taldelzena = %taldelzena%

sql_tamaina: SELECT COUNT(*) as count
FROM PokemonTaldean PT
JOIN Taldea T ON PT.taldelzena = T.taldelzena
WHERE T.erabiltzailelzena = %erabiltzailelzena% AND T.taldelzena = %taldelzena%

sql_generoa: SELECT generoa
FROM PokemonPokedex
WHERE pokId = %pokId%

sql_harrapatu: INSERT INTO
PokemonTalde (izena, maila, adiskidetasun_maila, generoa, HP, ATK, SPATK, DEF,
SPDEF, SPE, PokemonPokedexID, Erabiltzailelzena)
VALUES (%izena%, 50, 0, %generoa%, %hp%, %atk%, %spatk%, %def%, %spdef%,
%spe%, %pokId%, %erabiltzailelzena%)

```
sql_azkenharrapatu: SELECT MAX(harrapatuld) as id  
FROM PokemonTalde  
WHERE Erabiltzailelzena = %erabiltzailelzena% AND PokemonPokedexID =  
%pokeld%
```

```
sql_ablezi: SELECT izena  
FROM IzanDezake  
WHERE pokemonPokedexID = %pokeld%
```

```
sql_mugi: SELECT mugilzena  
FROM IkasDezake  
WHERE pokemonPokedexID = %pokeld%
```

```
sql_dauka: INSERT INTO  
Dauka (abilezilzena, harrapatuld)  
VALUES (%abilezilzena%, %harrapatuld%)
```

```
sql_mugimenduak: INSERT INTO  
MugimendulzanTalde (harrapatuld, mugimendulzena)  
VALUES (%harrapatuld%, %mugimendulzena%)
```

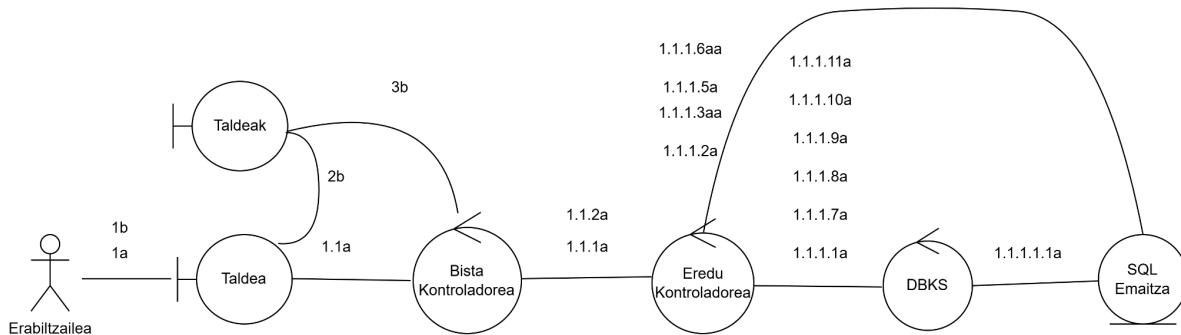
```
sql_taldera: INSERT INTO  
PokemonTaldean (taldezena, harrapatuld, erabiltzailelzena)  
VALUES (%taldezena%, %harrapatuld%, %erabiltzailelzena%)
```

```
sql_izena: SELECT izena  
FROM PokemonTalde  
WHERE harrapatuld = %harrapatuld%
```

```
JSON1: {[izena: string, argazkia: string, ID: int]}  
JSON2: {[izena: string, motak: string[], generazioak: int[]]}  
JSON3: {[pokelD: int, argazkia: string]}  
JSON4: {[{taldezena: String}]}  
JSON7 = {izena: string, argazkia, image, deskripzioa: string, ID: int, abileziak:  
string[]}  
JSON9 = {hp: int, atk: int, spatk: int, def: int, spdef: int, spe: int}
```

4.3.2.2 Taldea gorde eta ezabatu

Taldea gordetzeko eta ezabatzeko komunikazio diagrama aparte egin da komunikazio diagramek nahiko espazioa okupatzen zutelako eta zati hau gehitzea gehiegiz izango litzateke. Edozein momentutan, erabiltzailea Taldea bistean dagoen bitartean, egin ahal diren akzioak dira, bai taldea sortzean zein editatzean.



Irudia 132: Talde gorde eta ezabatu funtzionalitatearen komunikazio diagrama

[Erabiltzaileak taldea ezabatu nahi badu]

1a- Erabiltzaileak "Ezabatu" botoia sakatzen badu

1.1a- taldea_ezabatu()

- 1.1.1a- ezabatu_taldea(taldezena, erabiltzailezena)
- 1.1.1.1a- execSQL(sql_pokemonguztiak): SQL Emaitza
- 1.1.1.1.1a- new SQL Emaitza()
- 1.1.1.2a- next()
- [next() true den bitartean]
- 1.1.1.3aa- getInt("harrapatuld"): Int
- 1.1.1.4a- execSQL(sql_ezabatulzena): SQL Emaitza
- 1.1.1.4.1a- new SQL Emaitza()
- 1.1.1.5a- next()
- [next() true den bitartean]
- 1.1.1.6aa- getString("izena"): String
- 1.1.1.7a- execSQL(sql_mugiezabatu)
- 1.1.1.8a- execSQL(sql_abilezabatu)
- 1.1.1.9a- execSQL(sql_taldekanpo)
- 1.1.1.10a- execSQL(sql_askatu)
- 1.1.1.11a- execSQL(sql_taldeaezabatu)

1.1.2a- notigikazioBerria(erabiltzailezena, data, deskripzioa)

[Erabiltzaileak taldea gorde nahi badu]

1b- Erabiltzaileak "Gorde" botoia sakatzen badu

2b- new Taldeak()

3b- taldeak_kargatu():JSON4 (Taldea sortu implementatuta)

```
sql_pokemonguztiak = SELECT harrapatuld FROM PokemonTaldean  
WHERE taldelzena = %tadelzena% AND erabiltzailelzena = %erabiltzailelzena%
```

```
sql_ezabatulzena = SELECT izena FROM PokemonTalde  
WHERE harrapatuld = %harrapatuld%
```

```
sql_mugiezabatu = DELETE FROM MugimendulzanTalde  
WHERE harrapatuld = %harrapatuld%
```

```
sql_abilezabatu = DELETE FROM Dauka  
WHERE harrapatuld = %harrapatuld%
```

```
sql_taldekanpo = DELETE FROM PokemonTaldean  
WHERE harrapatuld = ? AND taldelzena = %tadelzena% AND erabiltzailelzena =  
%erabiltzailelzena%
```

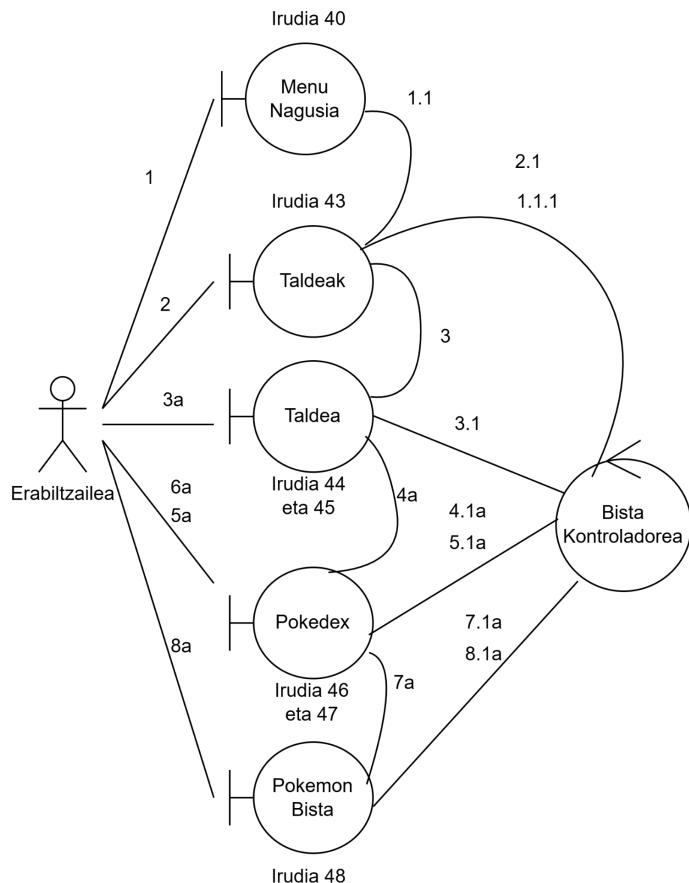
```
sql_askatu = DELETE FROM PokemonTalde  
WHERE harrapatuld = %harrapatuld%
```

```
sql_taldeaezabatu = DELETE FROM Taldea  
WHERE taldelzena = %tadelzena% AND erabiltzailelzena = %erabiltzailelzena%
```

JSON4: {[{tadelzena: String}]}]

4.3.2.3 Taldea editatu

Bi zatitan banatuko da diagrama hau, alde batetik Pokemon bat sartzeko, eta bestetik, ezabatzeko edo gordetzeko (Biak antzekoak baitira)



Irudia 133: Talde editatu funtzionalitatearen komunikazio diagrama (Pokemon gehitu)

1- Erabiltzaileak Menu Nagusian "Taldeak" botoia sakatzen du

1.1- new Taldeak()

1.1.1- taldeak_kargatu(): JSON4 (Taldea sortu implementatuta)

2- Erabiltzailea "EDIT" botoia sakatzen du.

2.1- new Taldea()

2.1.1- taldea(taldezena): JSON3 (Taldea sortu implementatuta)

2.1.2- notifikazioBerria(erabiltzailezena, data, deskripzioa)

[Erabiltzailea Gorde edo Ezabatu klikatzen ez duen bitartean hurrengoa errepikatu]

[Erabiltzailea "+" botoia sakatzen badu]

3a- Erabiltzailea "+" botoia sakatzen du.

4a- new Pokedex()

4.1a- pokedexKargatu(JSON1): JSON2

[Erabiltzaileak filtroak ezartzen baditu]

5aa- Filtroa aukeratu eta bilaketa sakatzen du

5.1aa- pokedexKargatu(JSON2): JSON1

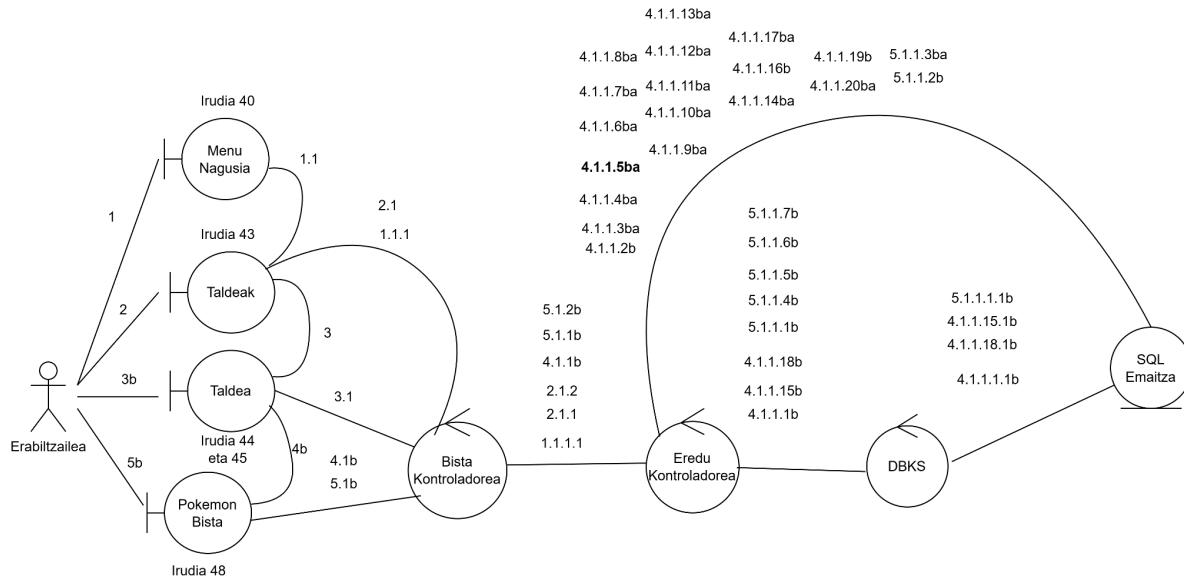
6a- Erabiltzaileak Pokemon baten argazkian klik egiten du.

7a- new PokemonBista(pokemonID)

7.1a- bistaratuPokemon(pokemonID): JSON7

8a- Erabiltzaileak "Hautatu" botoia sakatzen du.

8.1a- sartu_taldera()



**Irudia 134: Talde editatu funtzionalitatearen komunikazio diagrama
(Pokemon Aldatu edo Ezabatu)**

1- Erabiltzaileak Menu Nagusian "Taldeak" botoia sakatzen du

1.1- new Taldea()

1.1.1- taldeak_kargatu(): JSON4 (Taldea sortu implemenatuta)

2- Erabiltzailea "EDIT" botoia sakatzen du.

2.1- new Taldea()

2.1.1- taldea(taldezena): JSON3 (Taldea sortu implemenatuta)

2.1.2- notifikazioBerria(erabiltzailezena, data, deskripzioa)

[Erabiltzailea Gorde edo Ezabatu klikatzen ez duen bitartean hurrengoa errepikatu]

[Erabiltzaileak Pokemon baten argazkian sakatzen badu]

3b- Erabiltzaileak Pokemon baten argazkian sakatzen du

4b- new PokemonEzaugarriak(pokemonID)

4.1b- bistaratuPokemon(pokemonId): JSON6

4.1.1b- bistaratu_pokemon_taldea(pokemonID): JSON6

4.1.1.1b- execSQL(sqlPokeBista): SQL Emaitza

4.1.1.1.1b- new SQL Emaitza()

4.1.1.2b- next(): boolean

[next() true den bitartean]

4.1.1.3ba- getInt("harrapatuld"): Int
4.1.1.4ba- getString("izena"): String
4.1.1.5ba- getString("argazkia"): String
4.1.1.6ba- getInt("id"): Int
4.1.1.7ba- getfloat("pisua"): float
4.1.1.8ba- getfloat("altuera"): float
4.1.1.9ba- getInt("hp"): Int
4.1.1.10ba- getInt("atk"): Int
4.1.1.11ba- getInt("spatk"): Int
4.1.1.12ba- getInt("def"): Int
4.1.1.13ba- getInt("spdef"): Int
4.1.1.14ba- getInt("spe"): Int
4.1.1.15b- execSQL(sqlAbil): SQL Emaitza
 4.1.1.15.1b- new SQL Emaitza()
 4.1.1.16b- next(): boolean
 [next() true den bitartean]
 4.1.1.17ba- getString("abilezilzena"): String
 4.1.1.18b- execSQL(sqlMugi): SQL Emaitza
 4.1.1.18.1b- new SQL Emaitza()
 4.1.1.19b- next(): boolean
 [next() true den bitartean]
 4.1.1.20ba- getString("mugimendulzena"): String

[Erabiltzaileak "Ezabatu" edo "Aldatu" botoia sakatzen badu]

5b- Erabiltzaileak "Ezabatu" botoia sakatzen du

5.1b- ezabatu_taldetik()
 5.1.1b- ezabatu_taldetik(taldezena, erabiltzailezena, pokemonId): String
 5.1.1.1b- execSQL(sql_ezabatulzena): SQL Emaitza
 5.1.1.1.1b- new SQL Emaitza()
 5.1.1.2b- next()
 [next() true den bitartean]
 5.1.1.3ba- getString("izena"): String
 5.1.1.4b- execSQL(sql_mugiezabatu)
 5.1.1.5b- execSQL(sql_abilzabatu)
 5.1.1.6b- execSQL(sql_taldekanpo)
 5.1.1.7b- execSQL(sql_askatu)
 5.1.2b- notifikazioBerria(erabiltzailezena, data, deskripzioa)
[Ezabatu aukeratu bada taldera bueltatzen da]
[Gorde aukeratu bada 3a pausura bueltatzen gara]

sql_ezabatulzena = SELECT izena FROM PokemonTalde
WHERE harrapatuld = %harrapatuld%

```
sql_mugiezabatu = DELETE FROM MugimendulzanTalde  
WHERE harrapatuld = %harrapatuld%
```

```
sql_abilezabatu = DELETE FROM Dauka  
WHERE harrapatuld = %harrapatuld%
```

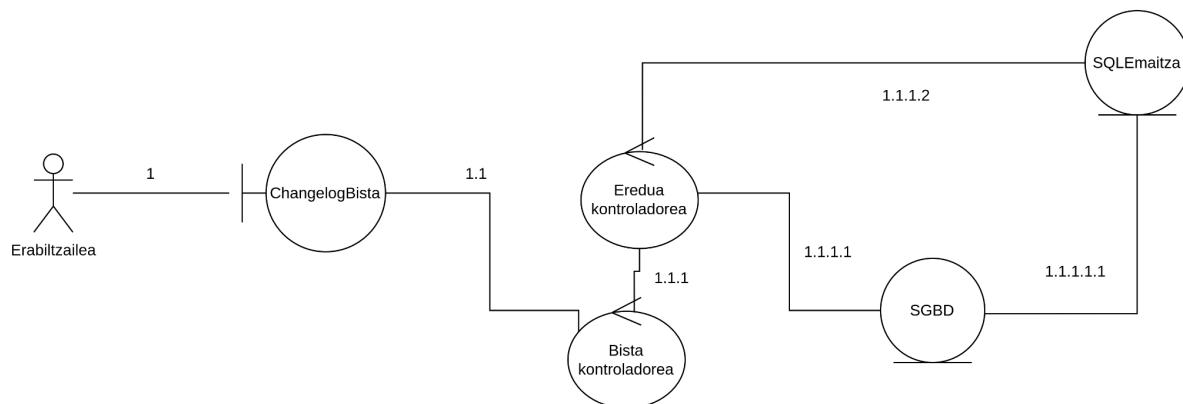
```
sql_taldekanpo = DELETE FROM PokemonTaldean  
WHERE harrapatuld = %harrapatuld% AND taldelzena = %tadelzena% AND  
erabiltzailelzena = %erabiltzailelzena%
```

```
sql_askatu = DELETE FROM PokemonTalde  
WHERE harrapatuld = %harrapatuld%
```

JSON1: {[izena: string, argazkia: string, ID: int]}
JSON2: {[izena: string, motak: string[], generazioak: int[]}]}
JSON3: {[pokelID: int, argazkia: string]}
JSON4: {[{tadelzena: String}]}
JSON6: {[{harrapatuld: int, izena: String, argazkia: String, id: int, pisua: float, altuera: float, hp: int, atk: int, spatk: int, def: int, spdef: int, spe: int, abileziak: string, mugilzenak: [String]}]}
JSON7 = {izena: string, argazkia, image, deskripzioa: string, ID: int, abileziak: string[]}

4.3.3 Changelog-a (Unai Rodriguez)

4.3.3.1 Notifikazioak kargatu izen baten menpe (Bilatu botoia)



Irudia 135: Changelog-a funtzionalitatearen komunikazio diagrama

1- Erabiltzailea ChangeLogBista-ra sartzen da, erabiltzaile baten izena idatzi eta bilatu botoia sakatzen du

1.1- notifikazioenInformazioaLortuerabiltzailea, bilatutakolzena): JSON;

1.1.1- notifikazioenInformazioaLortu(erabiltzailea, bilatutakolzena): JSON;

1.1.1.1- execSQL(SQL): SQLEmaitz

1.1.1.1.1- new SQLEmaitz()

1.1.1.2- next()

1.1.1.3- getDate("DataOrdua"): Date

1.1.1.4- getString("Erabiltzailezena"): String

1.1.1.5- getString("Deskripzioa"): String

SQL1: SELECT

J.Jarraitualzena,
N.DataOrdua,
N.deskripzioa

FROM

JarraitzenDu J

JOIN

Notifikatu N

ON J.Jarraitualzena = N.Erabiltzailezena

WHERE

J.Jarraitzailezena = ?

AND

J.Notifikatu= 1

AND

J.Jarraitulzena LIKE ?

ORDER BY

N.DataOrdua DESC;

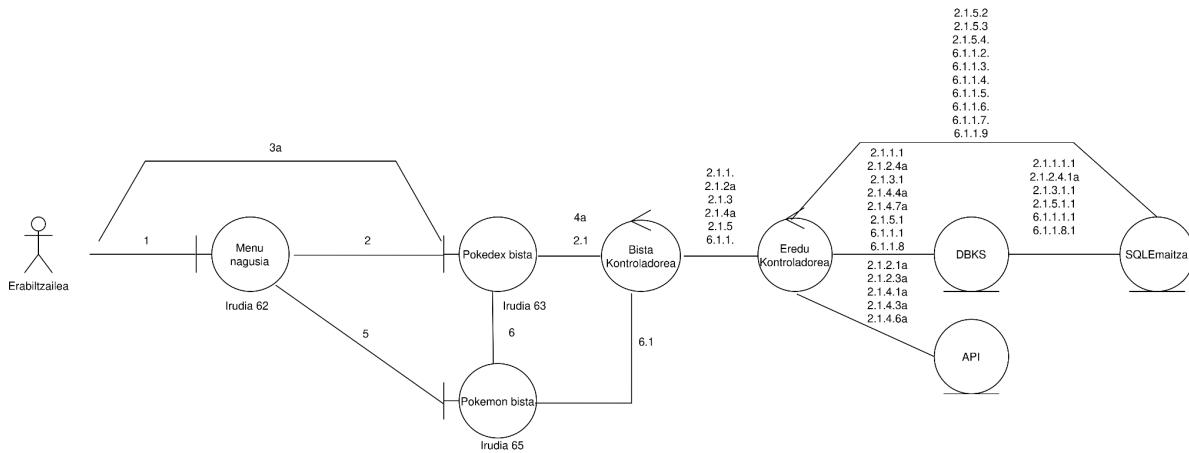
JSON1: {[{Jarraitulzena: String, DataOrdua: Date, deskripzioa:String}]}

Oharrak:

WHERE baldintzaren barruan SQLren konsultak parametro bat du (%izena), geroago saioaren erabiltzailearen izenarekin beteko dena. Datu-basetik notifikazioak ateratzeko prozesua aurrekoaren berdina baita. Aldatzen den gauza bakarra da SQLren konsultean parametro bat gehiago dagoela WHERE-n, bilatzen dituen notifikazioak bilaketa-barran idatzi den erabiltzaile zehatzarenak izan daitezen. Horretaz aparte, bistako kontroladorean isilduta dauden notifikazioak agertzea edo ez kudeatzen da, hau da, erabiltzaile batek beste bat sileziatuta badu, haren jakinarazpenak ez dira agertuko. Pertsona baten izena bilatzerakoan, ez da isilarazpen-sistema kontuan hartzen, ez baitu zentzurik eginkizun horretarako kontuan hartzea; izan ere, jakinarazpen jakin batzuk bilatzen ari zarenean, isilarazita egon behar dute edo ez.

Beltzez dagoen atala, if baten barruan dago non bilatzeko input-an ez badago ezer idatzita ez da query-an itsatsiko

4.3.4 Pokédex-a (Yassin Mehani)



Irudia 136: Pokedex funtzionalitatearen komunikazio diagrama

1. "Pokedex" botoia sakatzen du
 2. new Pokedex():
 - 2.1. pokedexKargatu(JSON1): JSON2
 - 2.1.1. pokemonakKonprobatu(): boolean
 - 2.1.1.1. execSQL(SQL1): SQLEmaitza
 - 2.1.1.1.1. new SQLEmaitza
 - [Pokemon-ak Datu Basean kargatuta ez badaude]
 - 2.1.2a pokemonakKargatu(): void
 - 2.1.2.1a pokemonIzenakEskatu(): JSON3
 - 2.1.2.2a next()
 - 2.1.2.3a pokemonakEskatu(izena): JSON4
 - 2.1.2.4a execSQL(SQL2): SQLEmaitza
 - 2.1.2.4.1a new SQLEmaitza
 - 2.1.3. datuakKonprobatu(): boolean
 - 2.1.3.1. execSQL(SQL3): SQLEmaitza
 - 2.1.3.1.1. new SQLEmaitza
 - [Mota/Abilezia/Mugimendua datu basean kargatuta ez badaude]
 - 2.1.4a datuaKargatu(): void
 - 2.1.4.1a datualzenakEskatu(): JSON5
 - 2.1.4.2a next()
 - 2.1.4.3a datuaEskatu(izena): JSON6
 - 2.1.4.4a execSQL(SQL4): SQLEmaitza
 - 2.1.4.5a next()
 - 2.1.4.6a pokemonakEskatu(izena): JSON4
 - 2.1.4.7a execSQL(SQL5): SQLEmaitza
 - 2.1.5. pokedexKargatu(JSON1): JSON2
 - 2.1.5.1. execSQL(SQL6): SQLEmaitza
 - 2.1.5.1.1. new SQLEmaitza()

- 2.1.5.2. next()
- 2.1.5.3. getIzena("izena"): String
- 2.1.5.4. getArgazkia("argazkia"): image

[Erabiltzaileak nahi badu]

3a. filtro edo bilaketa sakatzen du

4a. pokedexKargatu(JSON1): JSON2

5. Erabiltzaileak pokemon bat aukeratzen du.

6. new PokemonBista():

6.1 bistaratuPokemon(id): JSON7

6.1.1. bistaratuPokemon(id): JSON7

6.1.1.1. execSQL(SQL7): SQLEmaitza

6.1.1.1.1. new SQLEmaitza

6.1.1.2. getArgazkia("argazkia"): image

6.1.1.3. getIzena("izena"): String

6.1.1.4. getDeskripzioa("deskr"): String

6.1.1.5. getId("id"): int

6.1.1.6. getAltuera("altuera"): float

6.1.1.7. getPisua("pisua"): float

6.1.1.8. execSQL(SQL8): SQLEmaitza

6.1.1.8.1. new SQLEmaitza

6.1.1.9. getAbilezia("izena"): String[]

JSON1 = {izena: String, motak: String[], generazioak: []}

JSON2 = [{id: int, izena: String, irudia: Image}]

JSON3 = [{izena:string, id: int}]

JSON4 = {id: int, izena: String, altuera: float, pisua: float, generoa: String, deskripzioa: String, irudia: Image, generazioa: int, preEboluzioid: int}

JSON5 = [{izena:string, id: int}]

JSON6: JSON honek aldaketak ditu eskatzen den datu-motaren arabera

JSON6.1(Mota) = {izena: string, id: int, pokemonak: [{id: int, izena: string}]}]

JSON6.2(Abilezia) = {izena: string, id: int, deskripzioa: pokemonak = [{id: int, izena: string, ezkutua: boolean}]}]

JSON6.3(Mugimendua) = {izena: string, potentzia: int, pp: int, efektua: string, pokemonMotalzena: string, pokemonak: [{id: int, izena: string}]}]

JSON7 = {izena: string, argazkia, image, deskripzioa: string, ID: int, abileziak: string[]}

SQL1 = SELECT * FROM PokemonPokedex

SQL2 = INSERT INTO PokemonPokedex (pokeld, izena, altuera, pisua, generoa, deskripzioa, irudia, generazioa, preEboluzioid)

SQL3 = SQL honek aldaketak ditu insertazen den datu-motaren arabera

SQL3.1(Mota) = SELECT * FROM MotaPokemon

SQL3.2(Abilezia) = SELECT izena FROM Abilezia

SQL3.3(Mugimendua) = SELECT * FROM Mugimendua

SQL4 = SQL honek aldaketak ditu insertatzen den datu-motaren arabera

SQL4.1(Mota) = INSERT OR IGNORE INTO MotaPokemon (pokemonMotalzena, irudia)

SQL4.2(Abilezia) = INSERT OR IGNORE INTO Abilezia (izena, deskripzioa)

SQL4.3(Mugimendua) = INSERT OR IGNORE INTO Mugimendua (izena, potentzia, zehaztazuna, PP, efektua, pokemonMotalzena)

SQL5 = SQL honek aldaketak ditu insertatzen den datu-motaren arabera

SQL5.1(Mota) = INSERT OR IGNORE INTO DaMotaPokemon (motalzena, pokemonID)

SQL5.2(Abilezia) = INSERT OR IGNORE INTO IzanDezake (pokemonPokedexID, izena, ezkutua)

SQL5.3(Mugimendua) = INSERT OR IGNORE INTO IkasDezake (pokedexId, mugilzena)

SQL6 = SELECT izena, irudia, pokeld

FROM PokemonPokedex P

JOIN DaMotaPokemon DM ON P.pokeld = DM.pokemonID

JOIN MotaPokemon M on DM.motalzena = M.pokemonMotalzena

WHERE P.izena Like %izena% (% gehigarri bat dago, erabiltzaileak izen ez oso bat ipintzen badu)

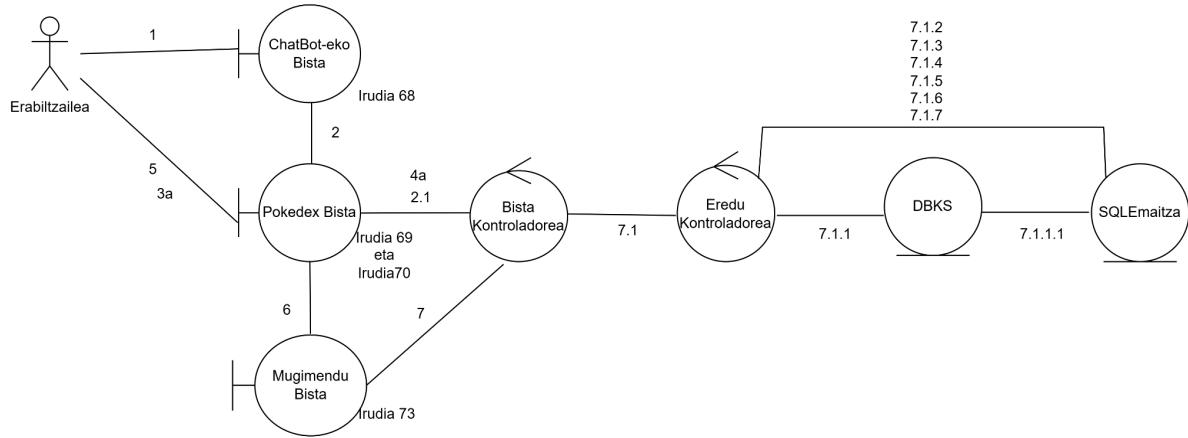
AND P.generazioa IN ({generazioak})

SQL7 = SELECT izena, pokeld, argazkia, deskripzioa FROM Pokemon(Pokedex) P
WHERE pokeld = %id%

SQL8 = SELECT izena FROM IzanDezake WHERE pokemonPokedexID = %id%

4.3.5 Chatbot-a (Xinyan Wang)

4.3.5.1 Mugimendu ikasgarriak



Irudia 137: Mugimendu ikasgarriak funtzionalitatearen komunikazio diagrama

- 1- "MUGIMENDUAK" botoia sakatzen du.
- 2- new Pokedex()
 - 2.1- pokedexKargatu(JSON2): JSON1

[Erabiltzaileak nahi badu]

 - 3a- filtro edo bilaketa sakatzen du.
 - 4a- pokedexKargatu(JSON2): JSON1
- 5- Erabiltzaileak pokemon bat aukeratzen du.
- 6- new MugimenduBista(id)
- 7- getMugimendulkasgarriak(id): pokemon_info
 - 7.1- getMugimendulkasgarriak(id): pokemon_info
 - 7.1.1- execSQL(SQL): SQLEmaitza
 - 7.1.1.1- new SQLEmaitza()
 - 7.1.2- next()
 - 7.1.3- getString("pokemon_izena"): string
 - 7.1.4- getString("pokemon_irudia"): string
 - 7.1.5- getString("mugimendu_izena"): string
 - 7.1.6- getInt("potentzia"): int
 - 7.1.7- getInt("zehaztasuna"): int

JSON1: [{id: int, izena: string, irudia:string}]

JSON2: {izena: string, motak: string[], generazioak: int[]}

pokemon_info: {izena: string, irudia: string, mugimenduak:[{mugimendulzena: string, potentzia: int, zehaztasuna: int}]]}

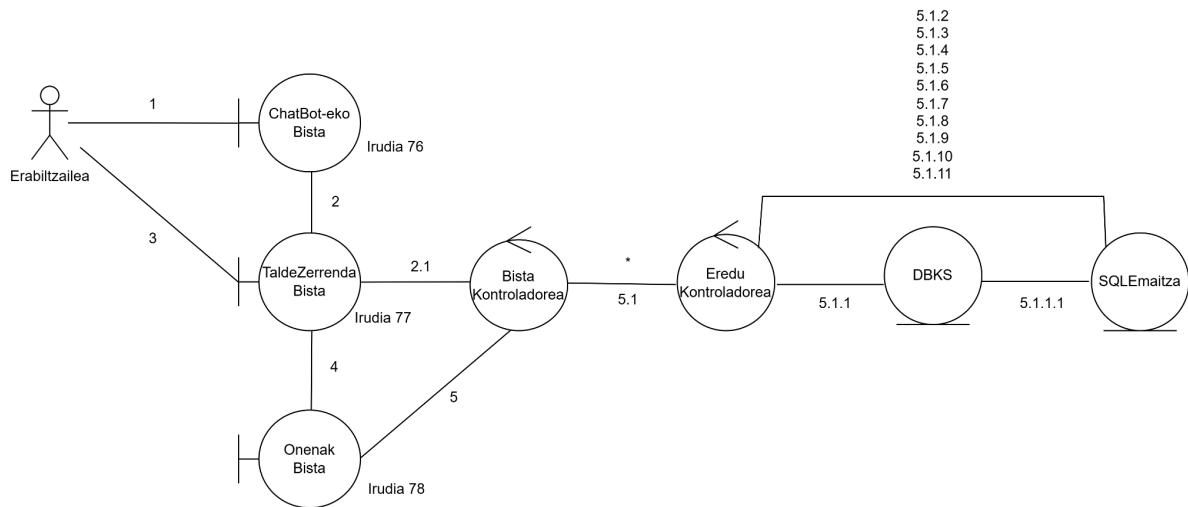
SQL:

```
SELECT p.izena AS pokemon_izena,
       p.irudia AS pokemon_irudia,
```

```

m.izena AS mugimendu_izena,
m.potenzia,
m.zehatztazuna
FROM PokemonPokedex p
    LEFT JOIN IkasDezake i ON p.pokeld = i.pokedexId
    LEFT JOIN Mugimendua m ON i.mugilzena = m.izena
WHERE p.pokeld = %pokemonID%;
```

4.3.5.2 Taldeko pokemon hoberena



Irudia 138: Taldeko pokemon hoberena funtzionalitatearen komunikazio diagrama

1. "ONENAK" botoia sakatzen du.
2. new TaldeZerrenda()
 - 2.1 taldeZerrenda(): JSON1
 - * taldeak_kargatu(erabiltzailelzena): JSON1
3. Erabiltzaileak talde bat aukeratzen du.
4. new Onenak(JSON2)
 5. getOnenak(JSON2): onena_info
 - 5.1 getOnenak(JSON2): onena_info
 - 5.1.1 execSQL(SQL1): SQLEmaitza
 - 5.1.1.1 new SQLEmaitza()
 - 5.1.2 next()
 - 5.1.3 getInt("PokemonPokedexID"): int
 - 5.1.4 getString("izena"): string
 - 5.1.5 getInt("HP"): int
 - 5.1.6 getInt("ATK"): int
 - 5.1.7 getInt("DEF"): int
 - 5.1.8 getInt("SPATK"): int

5.1.9 getInt("SPDEF"): int
5.1.10 getInt("SPE"): int
5.1.11 getString("irudia"): string

JSON1: {[{taldezena: String}]}

JSON2: {taldezena: string, erabiltzailezena: string}

onena_info: {PokemonPokedexID:int, izaña: string, irudia: string, HP: int, ATK: int, DEF: int, SPATK: int, SPDEF: int, SPE: int, puntuazioa: int, taldezena: string}

SQL1:

SELECT pt.PokemonPokedexID,

 pt.izaña,
 pt.HP,
 pt.ATK,
 pt.DEF,
 pt.SPATK,
 pt.SPDEF,
 pt.SPE,
 pd.irudia

FROM Taldea t

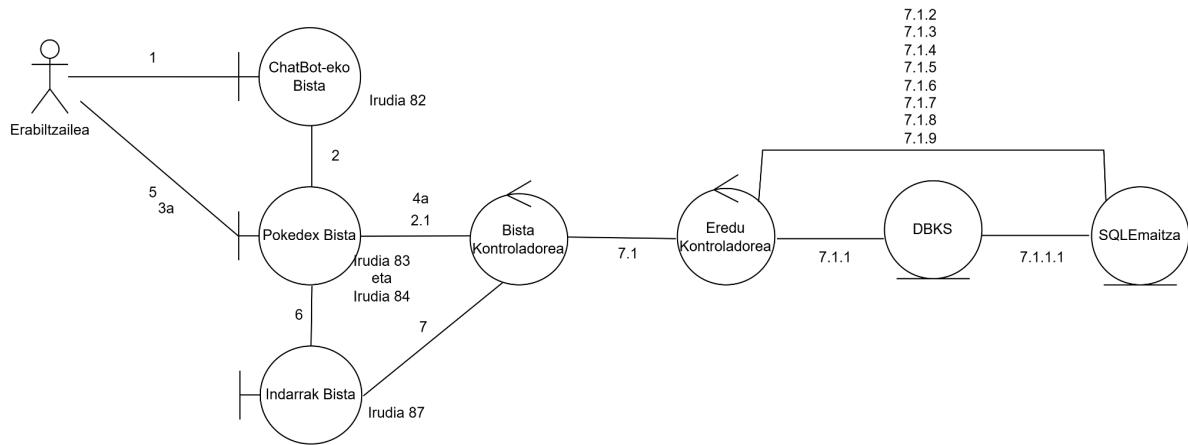
 INNER JOIN PokemonTaldean pta ON pta.taldezena = t.taldezena AND
 pta.erabiltzailezena = t.erabiltzailezena

 LEFT JOIN PokemonTalde pt ON pt.harrapatuld = pta.harrapatuld AND
 pt.erabiltzailezena = t.erabiltzailezena

 LEFT JOIN PokemonPokedex pd ON pt.PokemonPokedexID = pd.pokeld

WHERE t.taldezena = %taldezena% AND t.erabiltzailezena = %erabiltzailezena%;

4.3.5.3 Indarrak eta ahuleziak



Irudia 139: Indarrak eta ahuleziak funtzionalitatearen komunikazio diagrama

1. "INDARRAK" botoia sakatzen du
2. new Pokedex()
 - 2.1 pokedexKargatu(JSON2): JSON1
[Erabiltzaileak nahi badu]
 - 3a. filtro edo bilaketa sakatzen du.
 - 4a pokedexKargatu(JSON2): JSON1
5. Erabiltzaileak pokemon bat aukeratzen du.
6. new Indarrak(id)
7. getIndarrak(id): pokemon_info
 - 7.1 getIndarrak(id): pokemon_info
 - 7.1.1 execSQL(SQL1): SQL emaitza
 - 7.1.1.1 newSQLemaitza()
 - 7.1.2 next()
 - 7.1.3 getString("izena"): string
 - 7.1.4 getString("irudia"): string
 - 7.1.5 getString("mota_izena"): string
 - 7.1.6 getInt("mota_irudia"): string
 - 7.1.7 getString("pokemonMotaEraso"): string
 - 7.1.8 getString("multiplikatzalea"): int
 - 7.1.9 getString("erasoko_mota_irudia"): string

JSON1: [{id: int, izena: string, irudia:string}]

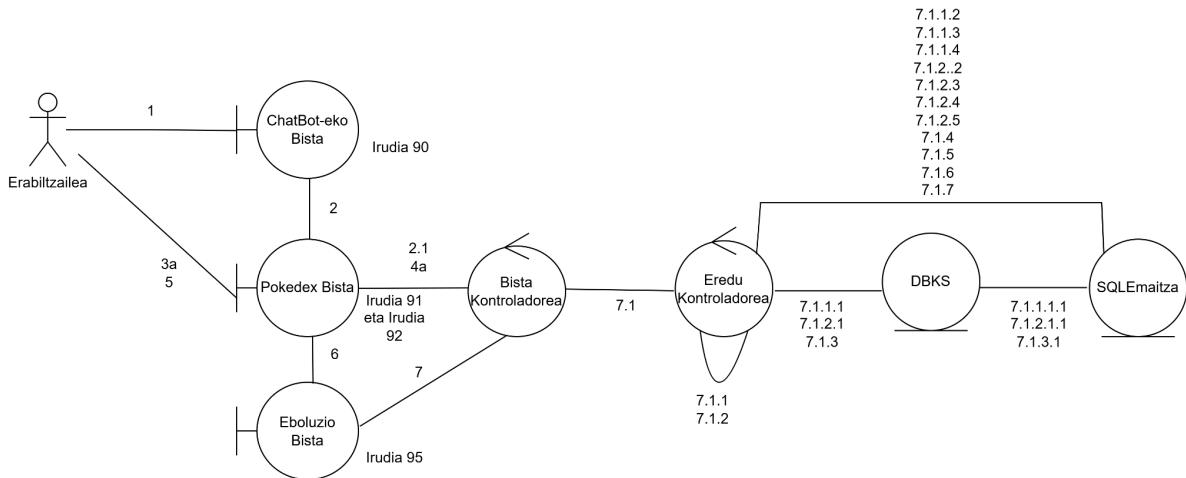
JSON2: {izena: string, motak: string[], generazioak: int[]}

pokemon_info: {izena: string, irudia: string, pokemon_motak:[{izena:string, irudia:string}]], Ahuleziak:[{izena: string, irudia: string}], Indarrak:[{izena: string, irudia: string}]]}

SQL1:

```
SELECT p.izena,
       p.irudia,
       t.pokemonMotalzena AS mota_izena,
       t.irudia AS mota_irudia,
       m.pokemonMotaEraso,
       m.multiplikatzalea,
       t2.irudia AS erasoko_mota_irudia
FROM PokemonPokedex p
      LEFT JOIN DaMotaPokemon d ON p.pokeld = d.pokemonID
      LEFT JOIN MotaPokemon t ON d.Motalzena = t.pokemonMotalzena
      LEFT JOIN Multiplikatzalea m ON t.pokemonMotalzena =
m.pokemonMotaJaso
      LEFT JOIN MotaPokemon t2 ON m.pokemonMotaEraso =
t2.pokemonMotalzena
WHERE p.pokeld = %pokemonId%;
```

4.3.5.4 Eboluzio katea



Irudia 140: Eboluzio katea funtzionalitatearen komunikazio diagrama

1. "EBOLUZIOA" botoia sakatzen du.
2. new Pokedex()
 - 2.1 pokedexKargatu(JSON2): JSON1

[Erabiltzaileak nahi badu]

3a filtro edo bilaketa sakatzen du.
 - 4a pokedexKargatu(JSON2): JSON1
5. Erabiltzaileak pokemon bat aukeratzen du.
6. new EboluzioBista(id)
7. getEboluzioa(id): eboluzio_info
 - 7.1 getEboluzioa(id):eboluzio_info
 - 7.1.1 bilatu_eboluzioak(id, aurrekooa=True): emaitza
 - 7.1.1.1 execSQL(SQL1): SQLEmailta
 - 7.1.1.1.1 new SQLEmailta()
 - 7.1.1.2 next()
 - 7.1.1.3 getInt("pokeld"): int
 - 7.1.1.4 getString("irudia"): string
 - 7.1.1.5 getString("izena"): string
 - 7.1.2 bilatu_eboluzioak(id, aurrekooa=False): emaitza
 - 7.1.2.1 execSQL(SQL2): SQLEmailta
 - 7.1.2.1.1 new SQLEmailta()
 - 7.1.2.2 next()
 - 7.1.2.3 getInt("pokeld"): int
 - 7.1.2.4 getString("irudia"): string
 - 7.1.2.5 getString("izena"): string
 - 7.1.3 execSQL(SQL3): SQLEmailta
 - 7.1.3.1 new SQLEmailta()
 - 7.1.4 next()
 - 7.1.5 getInt("pokeld"): int

7.1.6 getString("irudia"): string

7.1.7 getString("izena"): string

JSON1: [{id: int, izena: string, irudia:string}]

JSON2: {[izena: string, motak: string[], generazioak: int[]]}

emaitza: [{pokeld: int, izena: string, irudia:string}]

eboluzio_info: {izena: string, irudia: string, aurrekoak: emaitza, hurrengoak: emaitza}

SQL1:

```
SELECT P1.pokeld, P1.izena, P1.irudia  
FROM PokemonPokedex P1  
JOIN PokemonPokedex P2 ON P1.pokeld = P2.preEboluzioid  
WHERE P2.pokeld= %pokemonID%;
```

SQL2:

```
SELECT P2.pokeld, P2.izena, P2.irudia  
FROM PokemonPokedex P1  
JOIN PokemonPokedex P2 ON P1.pokeld = P2.preEboluzioid  
WHERE P1.pokeld = %pokemonID%;
```

SQL3:

```
SELECT izena, irudia FROM PokemonPokedex WHERE pokeld = %pokemonID%;
```

Oharra:

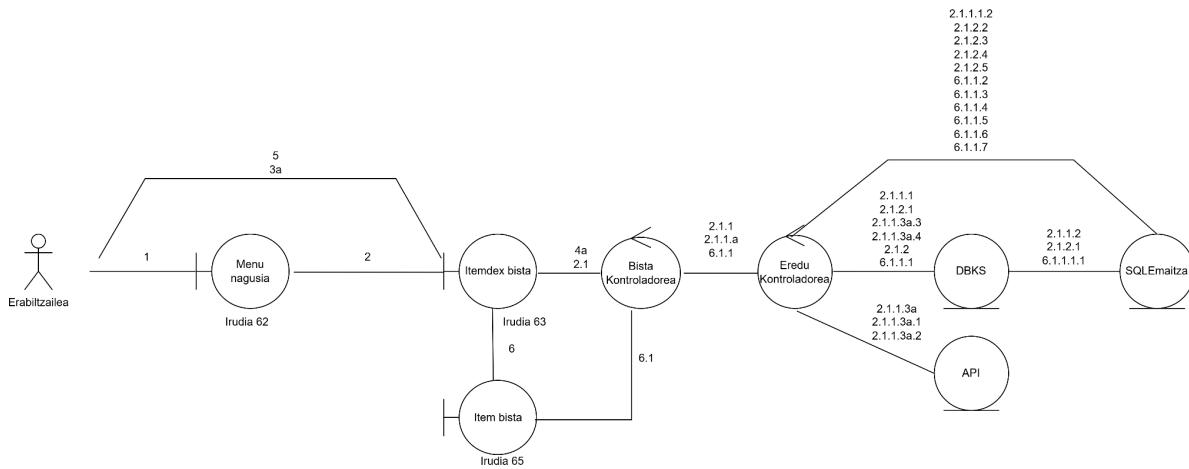
Pokemon askok hurrengo forma bat baino gehiago izan dezakete (adib. Eevee) eta, batzuetan, aurreko forma bat baino gehiago ere bai. Horregatik, emaitzak zerrenda moduan pilatu behar dira:

aurrekoak[]: aurreko formak

hurrengoak[]: hurrengo formak

SQL emaitzak ez dira beti erregistro bakarra; beraz beharrezkoa da iterazioa (while(next())). Horretarako BFS sistema erabiliko dugu aurreko formak eta hurrengo formak bilatzeko, errepiakenak saihesteko.

4.3.6 Itemdex (Yoel Justel)



Irudia 141: Itemdex funtzionalitatearen komunikazio diagrama

1. "Itemdex" botoia sakatzen du

2. new Itemdex()

 2.1. itemdex_kargatu(JSON2): JSON1

 2.1.1 itemdex_kargatu(JSON2): JSON1

 2.1.1.1. execSQL(SQL4): SQUEmailta

 2.1.1.1.1. new SQUEmailta

 2.1.1.2. next():boolean

 [Item-ak Datu Basean kargatuta ez badaude]

 2.1.1.3a. itemak_kargatu(): void

 2.1.1.3a.1. item_izenak_eskatu(): JSON4

 [Item bakoitzeko]

 2.1.1.3a.2. itema_eskatu(): JSON5

 2.1.1.3a.3. execSQL(SQL5): void

 2.1.1.3a.4. execSQL(SQL3): void

 2.1.2. execSQL(SQL1): SQUEmailta

 2.1.2.1. new SQUEmailta

 2.1.2.2. next()

 [Item bakoitzeko]

 2.1.2.3. getIzena(): String

 2.1.2.4. getArgazkia(): image

 2.1.2.5. getId(): int

[Erabiltzaileak nahi badu]

 3a. Filtro edo bilaketa sakatzen du

 4a. itemdex_kargatu(JSON2): JSON1

 (2.1. pausuak berriro exekutatzeten dira)

5. Erabiltzaileak item bat aukeratzen du.

6. new Item(id)

 6.1. bistaratu_item(id): JSON3

6.1.1. item(id): JSON3
 6.1.1.1. execSQL(SQL2): SQLEmaitza
 6.1.1.1.1. new SQLEmaitza
 6.1.1.2. next()
[Item bakoitzeko]
 6.1.1.3. getArgazkia(): image
 6.1.1.4. getIzena(): String
 6.1.1.5. getDeskripzioa(): String
 6.1.1.6. getId(): int
 6.1.1.7. getMotalzena(): String

JSON1: {[{izena: string, argazkia: string, ID: int}]}

JSON2: {izena: string, motak: string[], alfabetikokiAlderantziz: boolean}

JSON3: {izena: string, argazkia: image, deskripzioa: string, itemID: int, Motalzena: string}

JSON4: [{name: string}] (item izenak)

JSON5: {id: int, izena: string, deskripzioa: string, argazkia: string, mota: string}

SQL1: SELECT I.itemID as ID, I.izena, I.argazkia
 FROM Item I
 WHERE I=1
 AND I.izena LIKE '%izena%'
 AND I.Motalzena IN (...)
 ORDER BY I.izena ASC/DESC

SQL2: SELECT itemID, izena, deskripzioa, argazkia, Motalzena
 FROM Item
 WHERE itemID = ?
 LIMIT 1

SQL3: INSERT OR IGNORE INTO Item (itemID, izena, deskripzioa, argazkia,
Motalzena)
VALUES (?, ?, ?, ?, ?)

SQL4: SELECT * FROM Item LIMIT 1

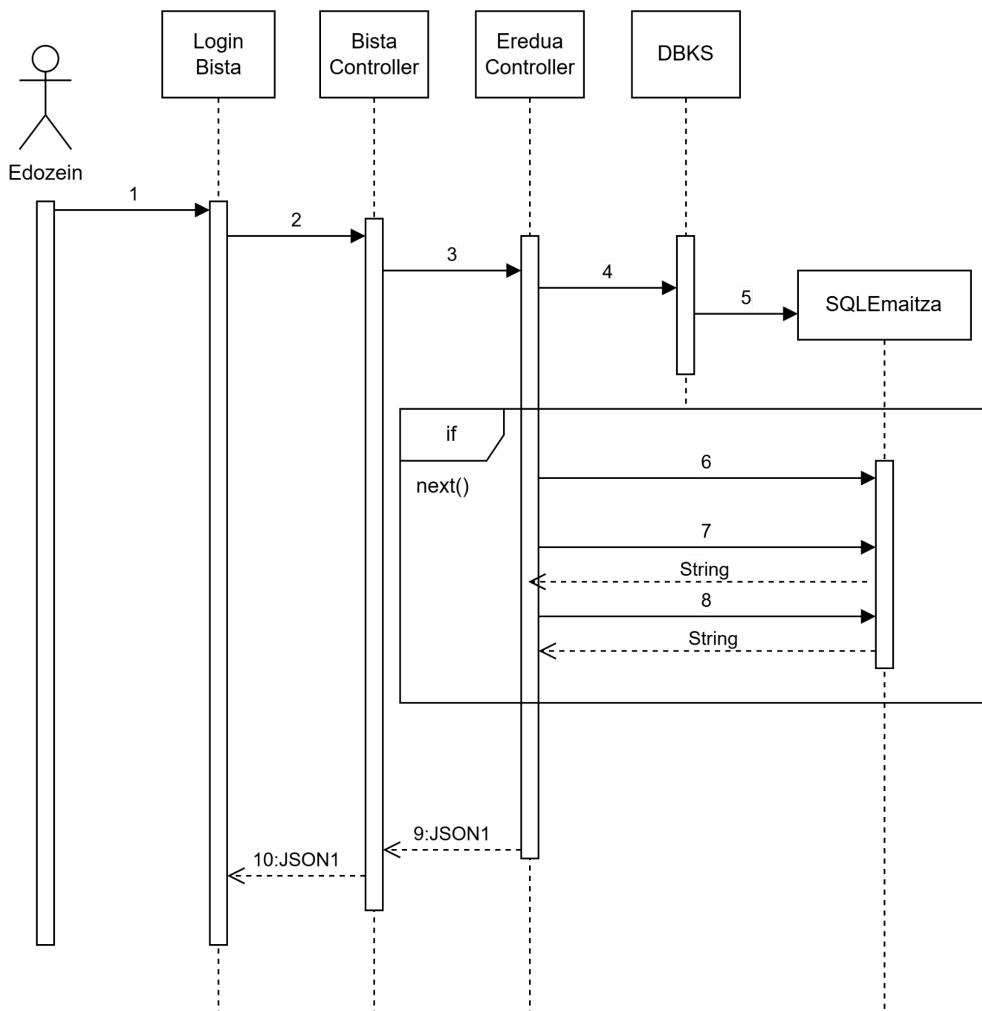
SQL5: INSERT OR IGNORE INTO MotalItem (ItemMotalzena) VALUES (?)

5. Entregagarria (Sekuentzia diagrama eta kodea)

5.1 Sekuentzia diagrama

5.1.1 Erabiltzaileen kudeaketa (Eneko Puente)

5.1.1.1 Saioa Hasi



Irudia 142: Saioa Hasi funtzionalitatearen sekuentzia diagrama

1: Erabiltzaileak bere izena eta pasahitza sartzen ditu eta "Saioa Hasi" botoia sakatzen du.

2:saioHasi()

3: saioHasi(izena, pasahitza):String

4: execSQL(SQL1): SQLEmaitza

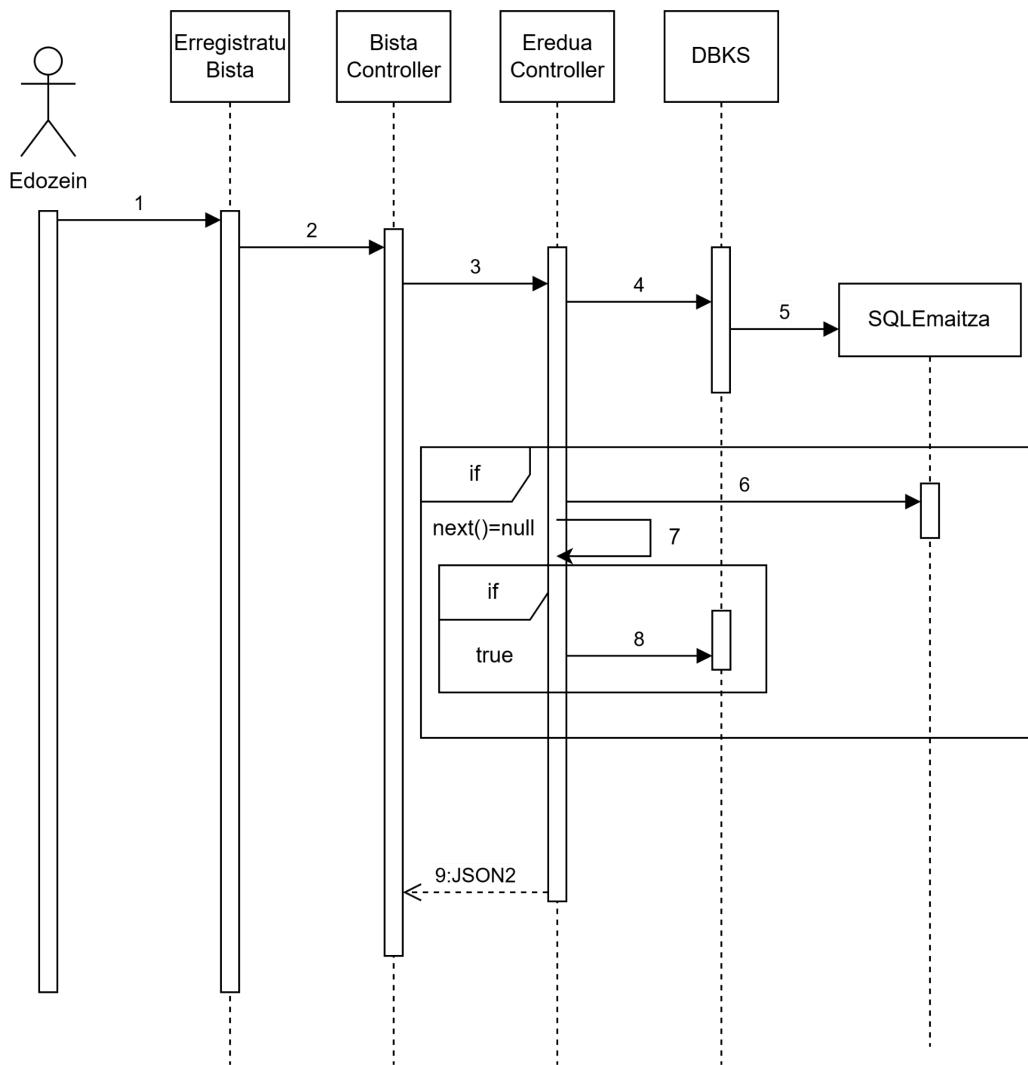
5: new SQLEmaitza

6: next()
[Emaitzaz ez null bada]
7a: getString("pasahitza"):String
8a: getString("Administratzaitzailea"):String

JSON1:

[{Ondo:Boolean, Erabiltzailezena:String,Rola:String,mezua:String},]

5.1.1.2 Erregistratu



Irudia 143: Erregistratu funtzionalitatearen sekuentzia diagrama

1: Erabiltzaileak bere datuak (Izena, pasahitza, Email, Jaiotza-Data, pasahitza, pasahitzaBerretsi) sartzen ditu eta "Erregistratu" botoia sakatzen du.

2: Erregistratu()

3: Erregistratu(izen, email, jaiotze_data, pasahitza1, pasahitza2): JSON2

4: execSQL(SQL2): SQLEmaitz

5: new SQLEmaitz

6 next()

7: balioztatu_pasahitza(pasahitza1, pasahitza2): JSON2

8: execSQL(SQL3)

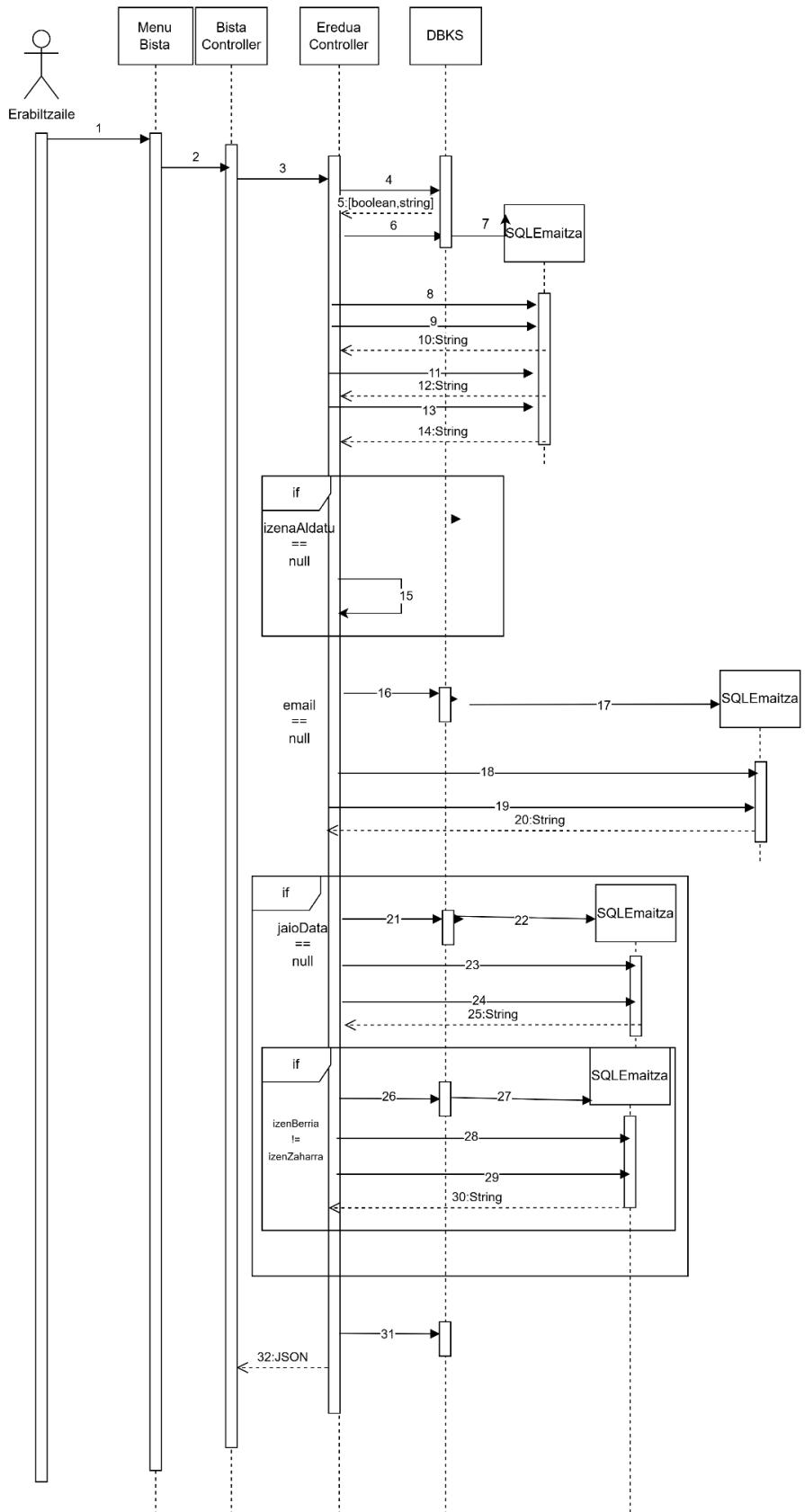
JSON2:

[{Ondo:Boolean, mezua:String},]

```
SQL2: SELECT izena  
      FROM Erabiltzaile  
      WHERE izena=%izena%
```

```
SQL3: INSERT INTO Erabiltzaile (Izena, Pasahitza, Email, JaioData,Rola)  
      VALUES (%Izena%, %Pasahitza%, %Emai%, %JaioData,%rola%)
```

5.1.1.3 Datuak Aldatu



Irudia 144: Datuak Aldatu funtzionalitatearen sekuentzia diagrama

1: "Editatu" botoia sakatzen du (bakarrik aldatzen dira sartutako datuak, besteak zeuden moduan gelditu dira)

2: datuakAldatu(erabiltzailezena)

3: eguneratuErabiltzailea(erabiltzailezena, izenaAldatu,pasahitzal,
emaila,jaioData):JSON

6: execSQL(SQL4)

7: new SQLEmaitza()

8:next()

9:getString("izena"):String

11:getString("pasahitza"):String

13:getString("rola"):String

15:izenBerria=izenZaharra

16:execSQL(SQL5)

17: new SQLEmaitza()

18:next()

19:getString("email"):String

21: execSQL(SQL6)

22: new SQLEmaitza()

23:next()

24:getString("jaioData"):String

26: execSQL(SQL7)

27: new SQLEmaitza()

28:next()

29:getString("izena"):String

31: execSQL(SQL8)

SQL4:SELECT izena, pasahitza, rola FROM Erabiltzailea WHERE izena = %izena%

SQL5:SELECT email FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL6:SELECT jaiotze_data FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL7:SELECT izena FROM Erabiltzailea WHERE izena = %erabiltzailezena%

SQL8: UPDATE UPDATE Erabiltzaile

SET izena=%izenaAldatu%,
pasahitza=%pasahitzal%,
email=%email%, jaiotze_data=%JaioData%

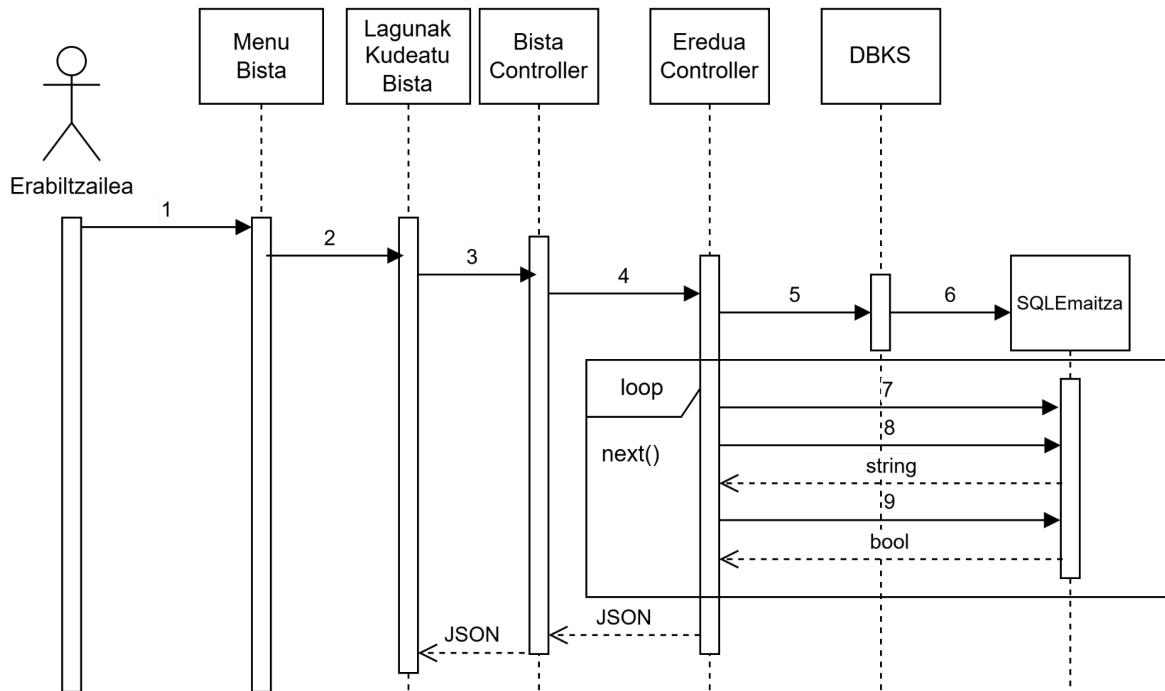
WHERE izena=%erabiltzailezena%

SQL8:

JSON: [{Ondo:Boolean, mezu:string},]

(JSON hau erabiltzen gero erabiltzailea behar den lekura bueltatzeko, izan ere, kudeaketa atalerako eta editatu atalerako atal bera erabiltzen da.)

5.1.1.4 (LAGUNAK) Lagunak Kargatu

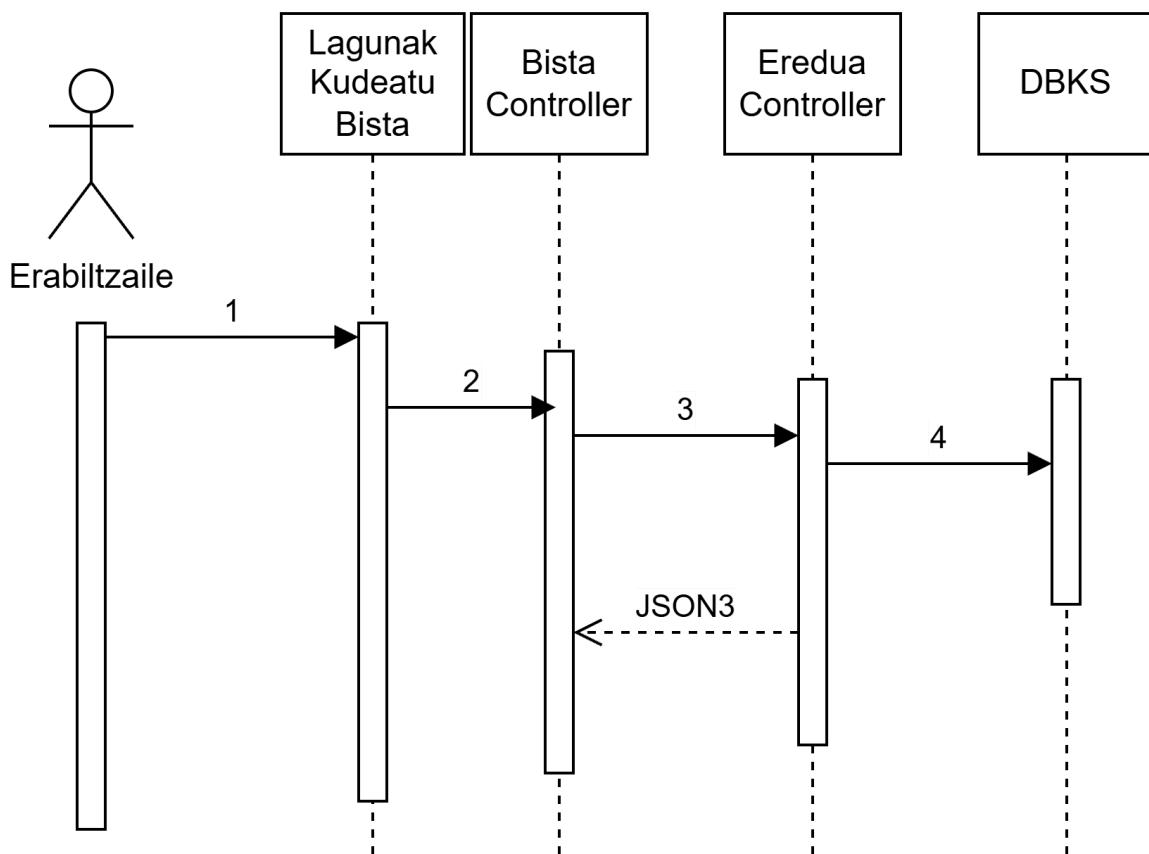


Irudia 145: Lagunak Kargatu funtzionalitatearen sekuentzia diagrama

- 1: Menuko "LagunakKudeatu" botoia sakatzen du.
- 2: new LagunakKudeatu()
- 3: lagunakKargatu(erabiltzailelzena: String):JSON1
- 4: lagunakKargatu(erabiltzailelzena: String):JSON1
- 5: execSQL(SQL4): EmaitzaSQL
- 6: new EmaitzaSQL()
- 7: next()
- 8: getString("izena"):String
- 9: getBoolean("notifikazioak"):Boolean

SQL4: SELECT Jarraitulzena, Notifikatu
 FROM JarraitzenDu
 WHERE Jarraitzailelzena = %erabiltzailelzena%
 ORDER BY Jarraitulzena
 JSON: [[[izena:string, notifikazioak:boolean],{}]]

5.1.1.5 (LAGUNAK) Utzi Jarraitzen

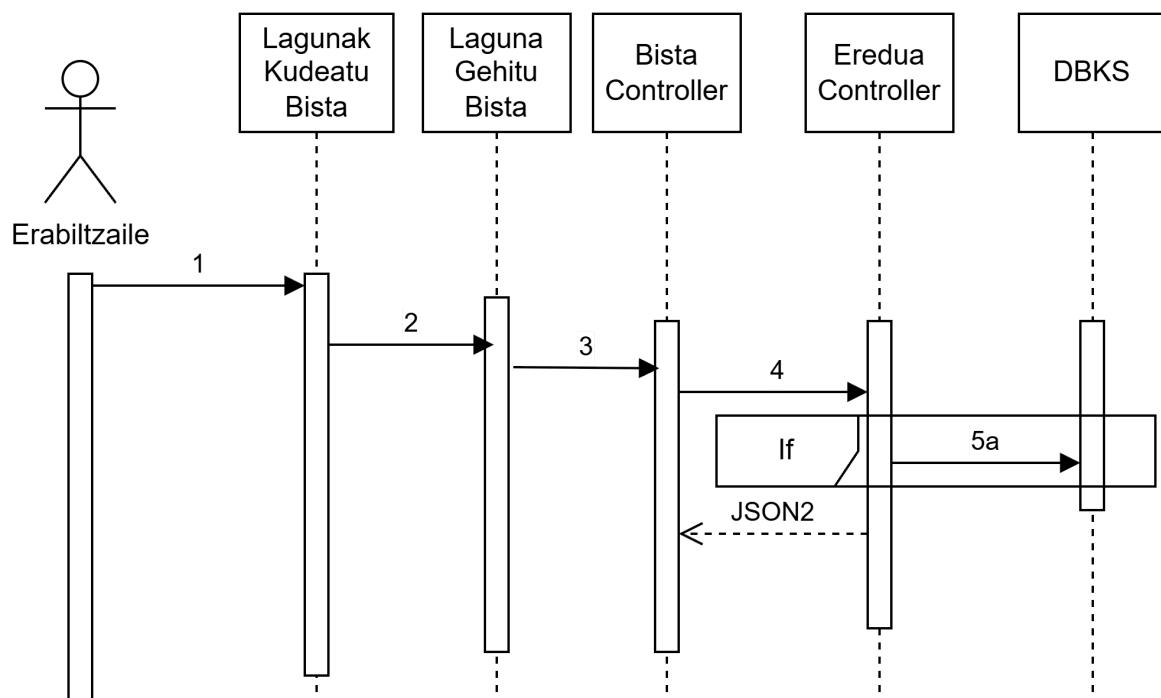


Irudia 146: Utzi Jarraitzen funtzionalitatearen sekuentzia diagrama

- 1: "Ezabatu" botoia sakatzen du
- 2: utzi_jarraitzen(plzenaEzabatu);
- 3: utzi_jarraitzen(erabiltzailezena:String, izenaEzabatu); JSON3
- 4: execSQL(SQL7)

SQL7: DELETE FROM JarraitzenDu
 WHERE Jarraitzailezena=%erabiltzailezena% And
 Jarraitulzena=%plzenaEzabatu%
 JSON3: [{ondoko:boolean, mezua: string}, {}]

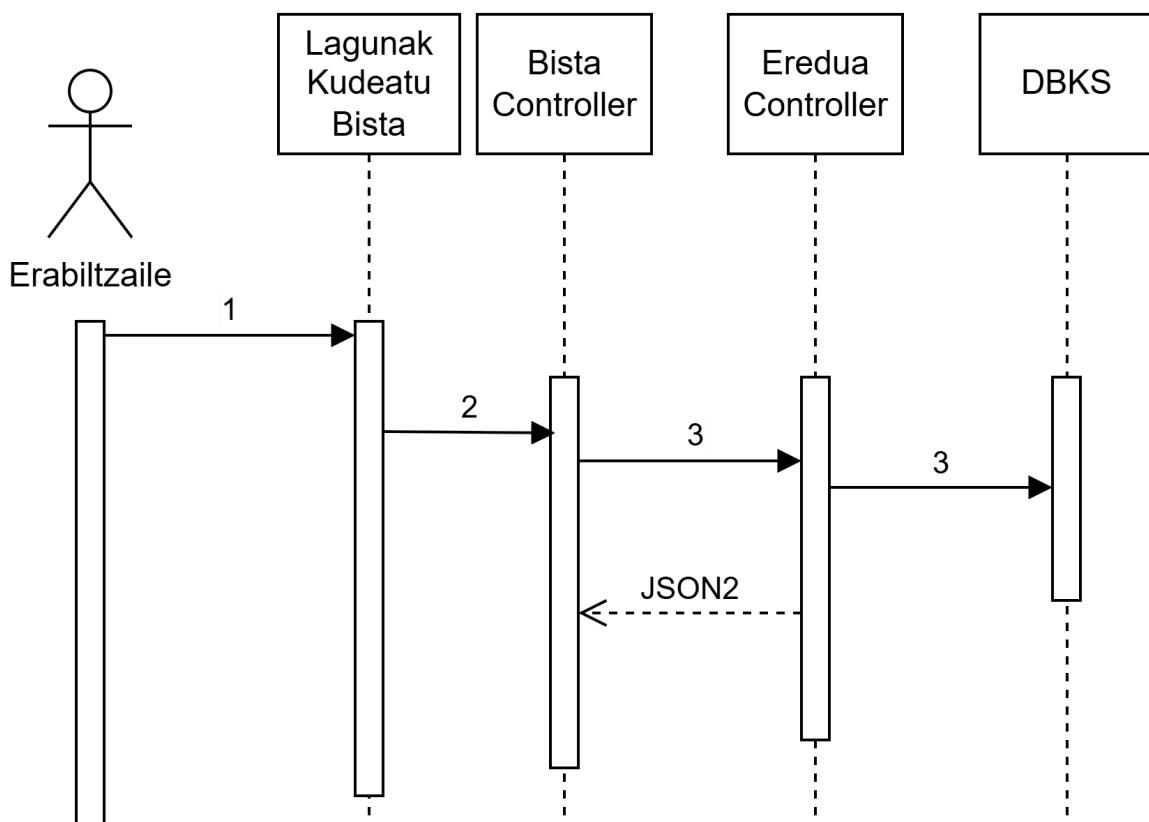
5.1.1.6 (LAGUNAK) Erabiltzailea Jarraitu



Irudia 147: Erabiltzailea Jarraitu funtzionalitatearen sekuentzia diagrama

- 1:"Gehitu" botoia sakatzen du
- 2: new LagunaGehitu()
- 3:gehituErabiltzailea(plzena:String)
- 4 gehituErabiltzailea(erabiltzailelzena: string, izena:String):JSON2
[Erabiltzailea existitzne bada eta jarraitu botoiari ematen badio]
- 5a: execSQL(SQL6)
SQL6:INSERT INTO JarraitzenDu (JarraitzailelZena, Jarraitulzena)
VALUES (%erabiltzailelzena%, %izena%)
- JSON2: [{ondo:boolean, mezua: string}, {}]]

5.1.1.7 (LAGUNAK) Notifikazioak Ezarri/Desgaitu



Irudia 148: Notifikazioak Ezarri/Desgaitu funtzionalitatearen sekuentzia diagrama

1: "Notifikazioak kendu" botoia sakatzen du

2:eguneratu_Notifikazioak(jarraitzulzena)

3:eguneratu_Notifikazioak(erabiltzailezena,jarraitzulzena,notifikazioak):JSON2

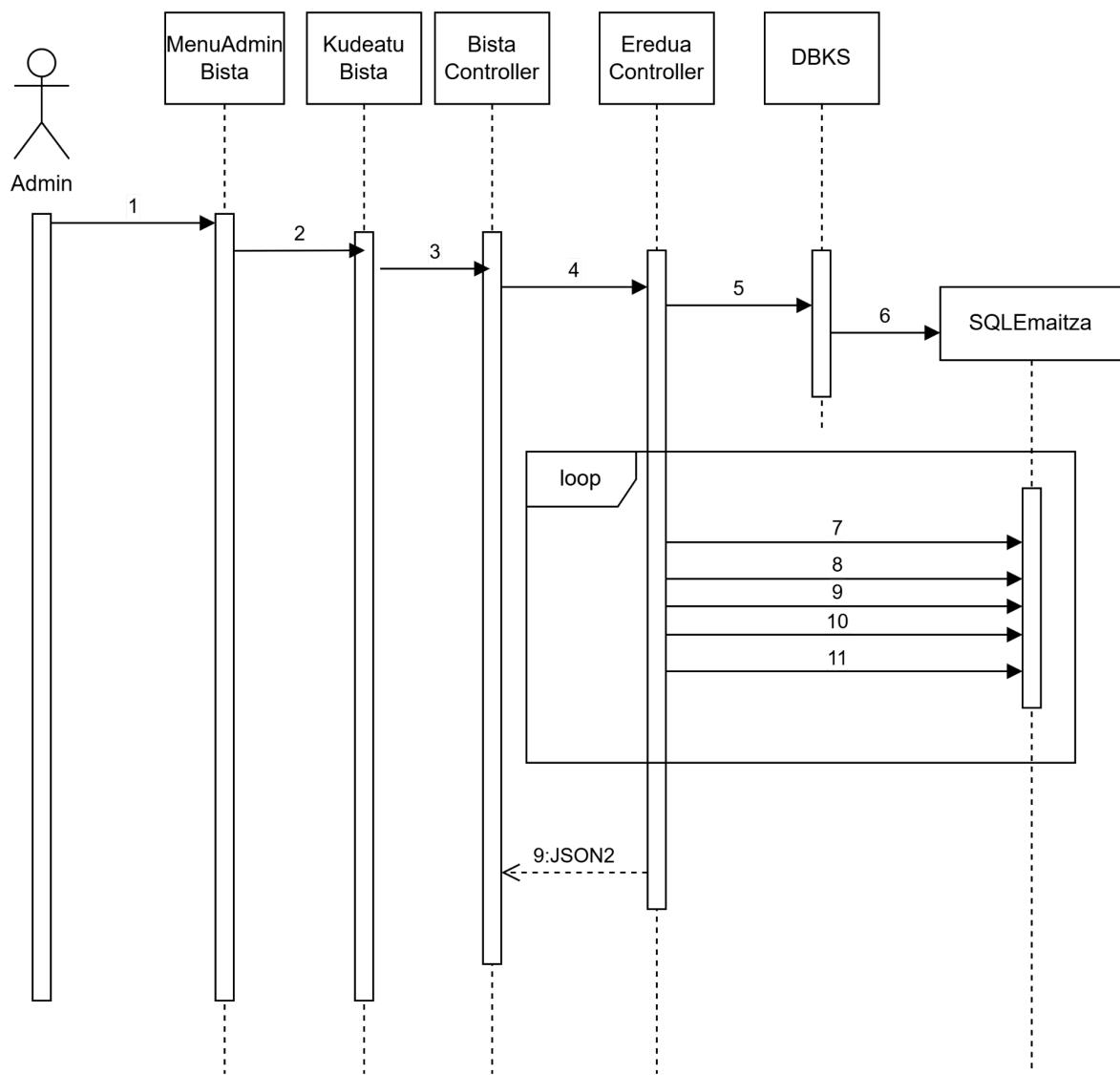
4: execSQL(SQL8)

SQL8: UPDATE JarraitzenDu SET Notifikatu = %notifikazioak% WHERE

Jarraitzailezena = %erabiltzailezena% AND Jarraitulzena = %jarraitulzena%

JSON2: [{ondo:boolean, mezua: string},{}]]

5.1.1.8 (ADMIN) Erabiltzaileak Kargatu



Irudia 149: Erabiltzaileak Kargatu funtzionalitatearen sekuentzia diagrama

1: Administratzaileak "Kudeatu" botoia sakatzen du

2: new Kudeatu()

3: erabiltzaileakKargatu()

4: erabiltzaileakKargatu():JSON1

5: execSQL(SQL1): SQLEmaitza

6: new SQLEmaitza()

7: next()

8: getString("izena"): String

9: getString("email"): String

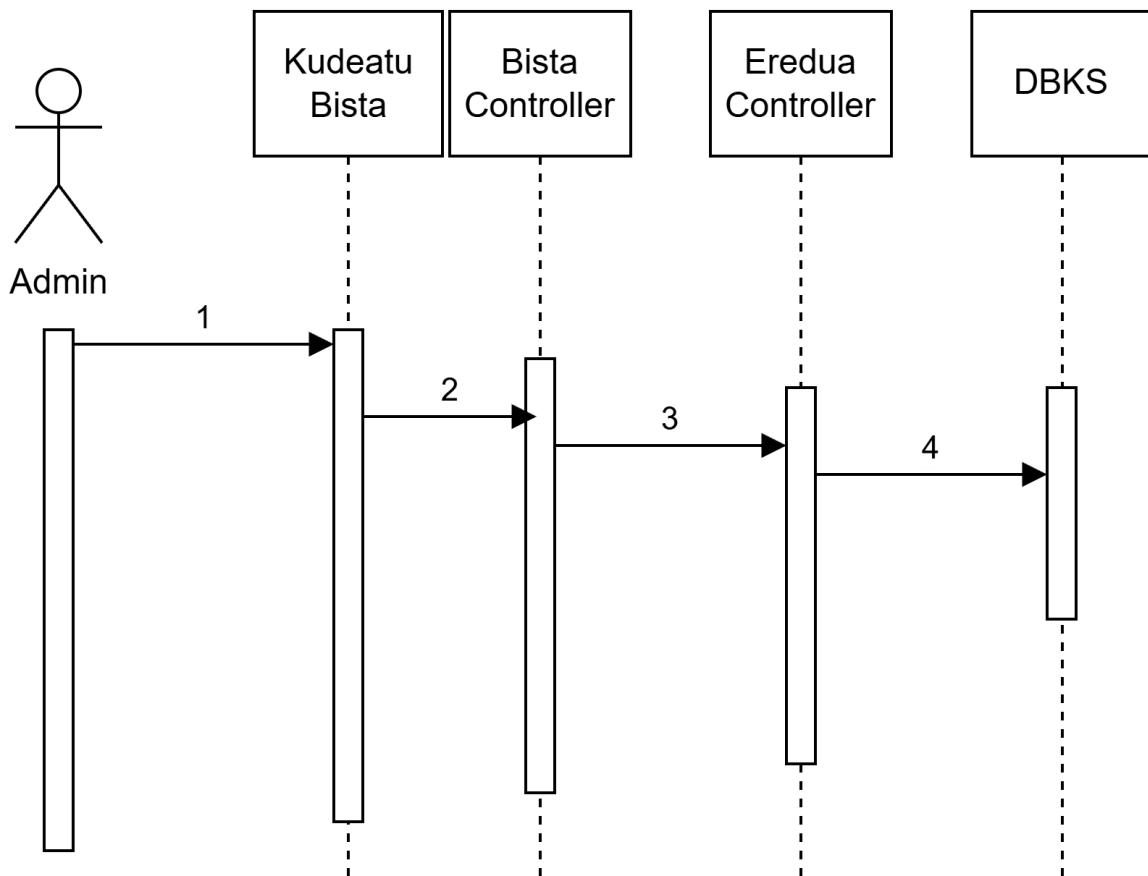
10: getString("rola"): String

11: getString("jaiotze_data"): String

JSON1:

```
[{erabiltzaileLista},]
```

5.1.19 (ADMIN) Ezabatu Erabiltzailea



Irudia 150: Ezabatu Erabiltzailea funtzionalitatearen sekuentzia diagrama

1: "Ezabatu" botoia sakatzen du

2: ezabatu(izena)

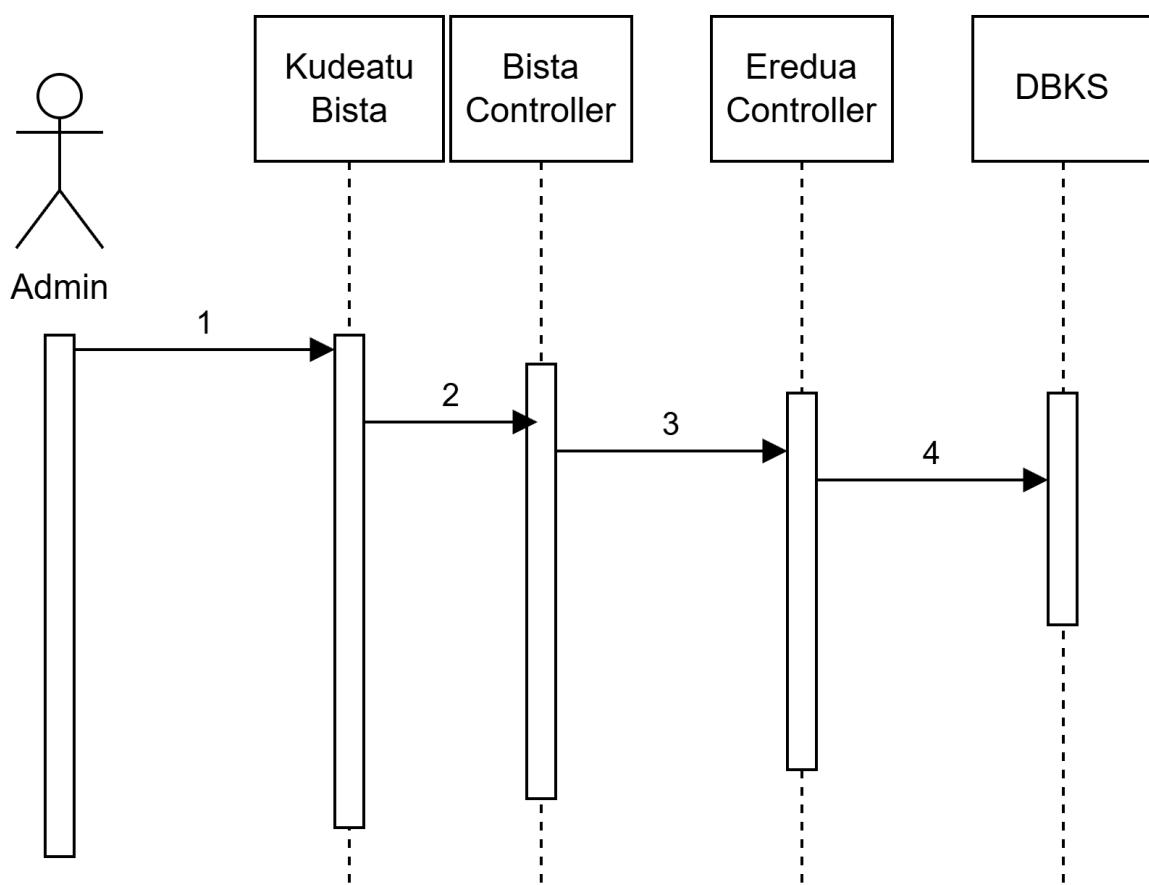
3: ezabatu(izena)

4: execSQL(SQL2)

SQL2:

```
DELETE FROM Erabiltzaile WHERE izena=%izena%
```

5.1.1.10 (ADMIN) Admin Bihurtu



Irudia 151: Admin Bihurtu funtzionalitatearen sekuentzia diagrama

1: "Baimendu" botoia sakatzen du

2: baimendu(izena)

3: baimendu(izena)

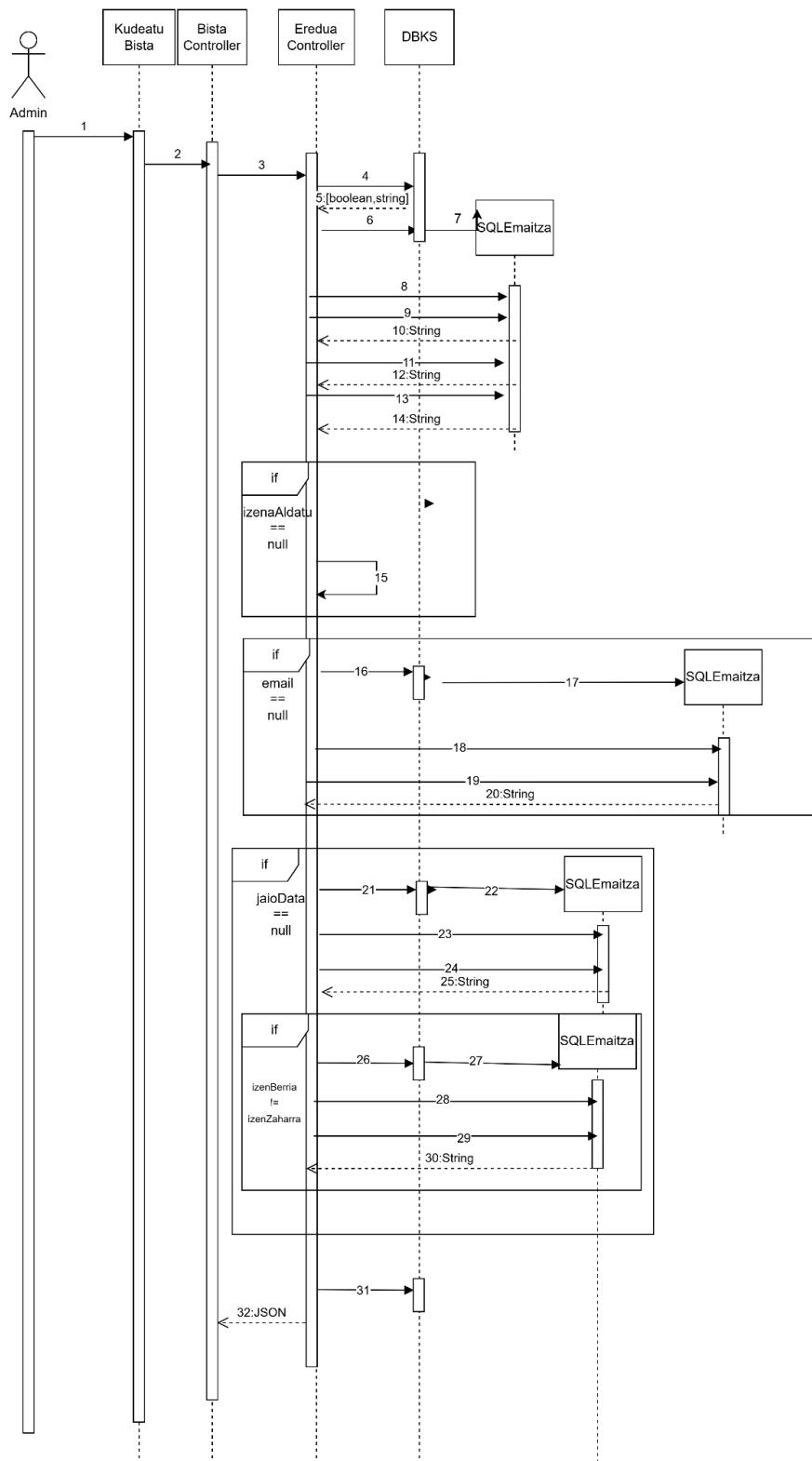
4: execSQL(SQL3)

SQL3: UPDATE Erabiltzaile

SET rola="admin"

WHERE izena=%izena%

5.1.1.11 (ADMIN) Editatu



Irudia 152: Admin Editatu funtzionalitatearen sekuentzia diagrama

1: "Editatu" botoia sakatzen du (bakarrik aldatzen dira sartutako datuak, besteak zeuden moduan gelditu dira)

2: datuakAldatu(erabiltzailezena)

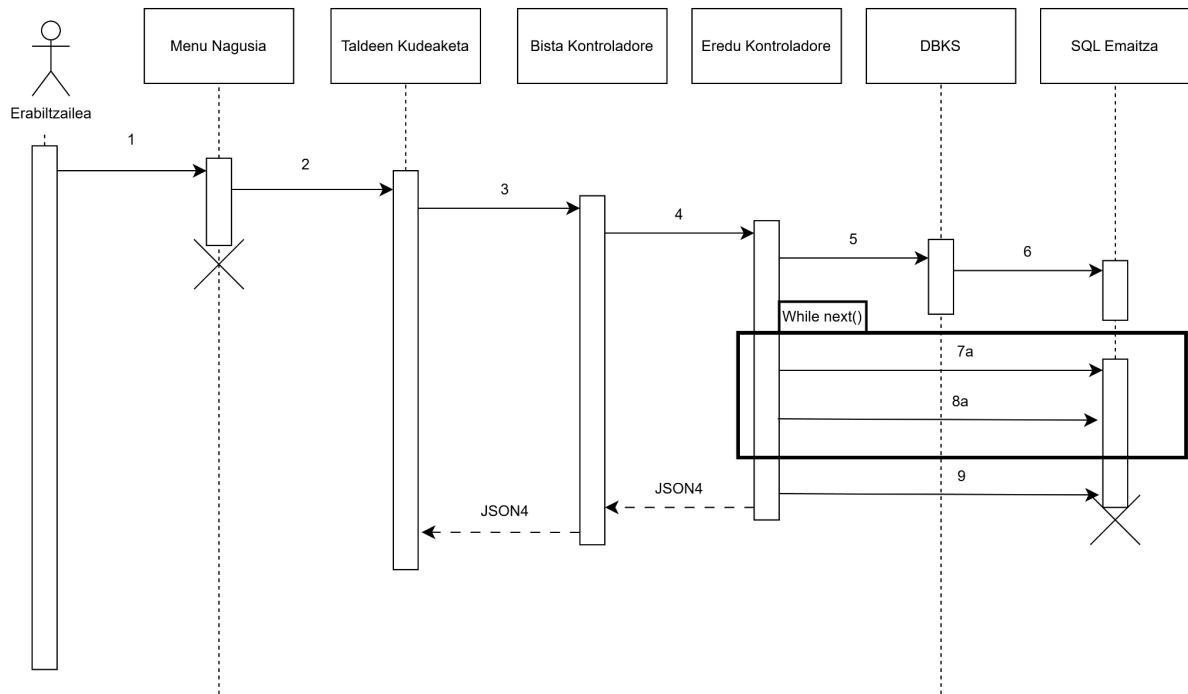
```
3: eguneratuErabiltzailea(erabiltzailelzena, izenaAldatu,pasahitzal,  
    emaila,jaioData):JSON  
6: execSQL(SQL4)  
    7: new SQLEmaitza()  
8:next()  
9:getString("izena"):String  
11:getString("pasahitza"):String  
13:getString("rola"):String  
15:izenBerria=izenZaharra  
16:execSQL(SQL5)  
    17: new SQLEmaitza()  
18:next()  
19:getString("email"):String  
21: execSQL(SQL6)  
    22: new SQLEmaitza()  
23:next()  
24:getString("jaioData"):String  
26: execSQL(SQL7)  
    27: new SQLEmaitza()  
28:next()  
    29:getString("izena"):String  
31: execSQL(SQL8)
```

SQL4:SELECT izena, pasahitza, rola FROM Erabiltzailea WHERE izena = %izena%
SQL5:SELECT email FROM Erabiltzailea WHERE izena = %erabiltzailelzena%
SQL6:SELECT jaiotze_data FROM Erabiltzailea WHERE izena = %erabiltzailelzena%
SQL7:SELECT izena FROM Erabiltzailea WHERE izena = %erabiltzailelzena%
SQL8: UPDATE UPDATE Erabiltzaile
 SET izena=%izenaAldatu%,
 pasahitza=%pasahitzal%,
 email=%email%, jaiotze_data=%JaioData%
 WHERE izena=%erabiltzailelzena%
SQL8:
JSON: [{Ondo:Boolean, mezu:string},]

(JSON hau erabiltzen gero erabiltzailea behar den lekura bueltatzeko, izan ere, kudeaketa atalerako eta editatu atalerako atal bera erabiltzen.)

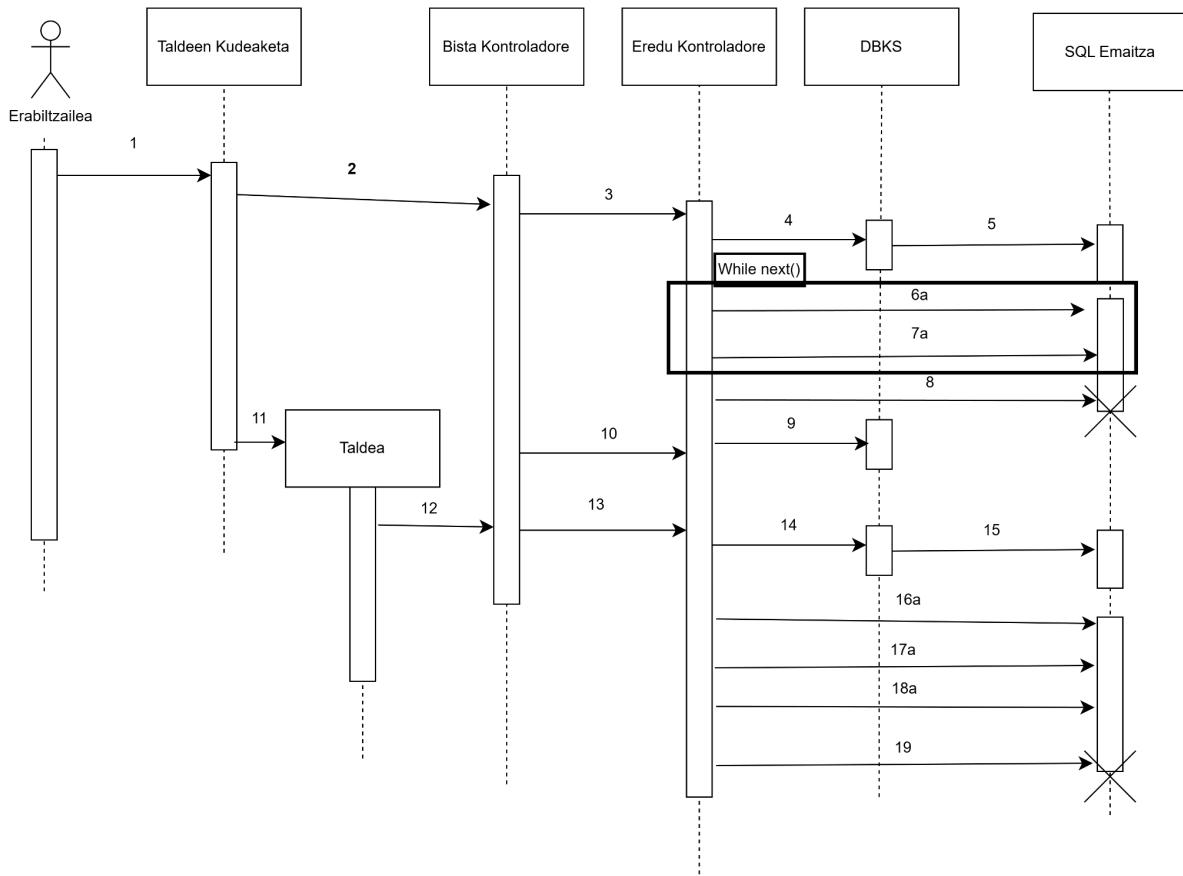
5.1.2 Pokemon taldeak sortu (Asier Barrio)

5.1.2.1 Talde berria sortu eta Pokemon sartu



Irudia 153: Talde berria sortu eta Pokemon sartu funtzionalitateen sekuentzia diagrama

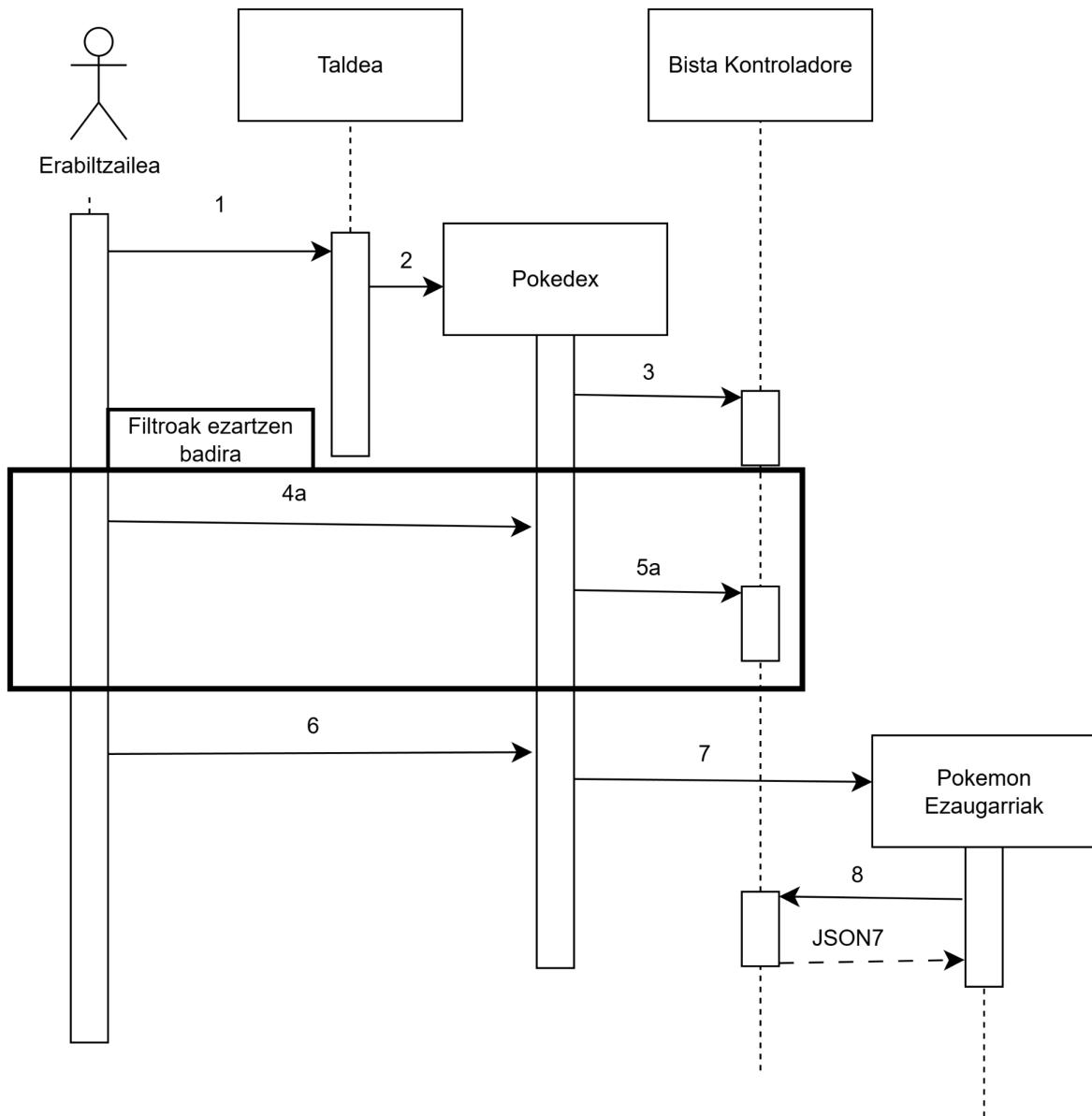
- 1- Erabiltzaileak Menu Nagusian "Taldeak" botoia sakatzen du
- 2- new Taldeak()
- 3- taldeak_kargatu(erabiltzailezena): JSON4
- 4- taldeak_kargatu(erabiltzailezena): JSON4
- 5- execSQL(sql_taldeak): SQL Ematiza
- 6- new SQL Emaitza()
- [next() = true]
- 7a- next(): boolean
- 8a- getString("taldelezena"): String
- 9- close()



Irudia 154: Talde berria sortu funtzionalitateen sekuentzia diagrama

- 1- Erabiltzailea "+" botoia sakatzen du.
- 2- sortu_taldeak()
- 3- sortu_taldea_hutsa(erabiltzailelzena)
- 4- execSQL(sql_taldeak): SQL Emaitz
- 5- new SQL Emaitz()
- 6- next(): boolean
[next() true den bitartean]
- 6a- next(): boolean
- 7a- getString("taldezena"): String
- 8- close()
- 9- execSQL(sql_sortu)
- 10- notifikazioBerria(erabiltzailelzena, data, deskripzioa)
- 11- new Talde(taldezena)
- 12- taldea(taldezena): JSON3
- 13- get_taldea(taldezena, erabiltzailelzena): JSON3
- 14- execSQL(sql_poketalde): SQL Emaitz
- 15- new SQL Emaitz()
- 16- next() = true

- 16a- next(): boolean
- 17a- getInt("pokeld"): Int
- 18a- getString("argazkia"): String
- 19- close()



Irudia 155: Pokedex-etik Pokemon aukeratzea funtzionalitateen sekuentzia diagrama

[Erabiltzaileak "Corde" edo "Ezabatu" sakatzen ez duen bitartean hurrengo diagramak emango dira] 1- Erabiltzailea "+" botoia sakatzen du.

2- new Pokedex(taldezena)

3- pokedexKargatu(JSON1): JSON12

[Erabiltzaileak filtroak ezartzen baditu]

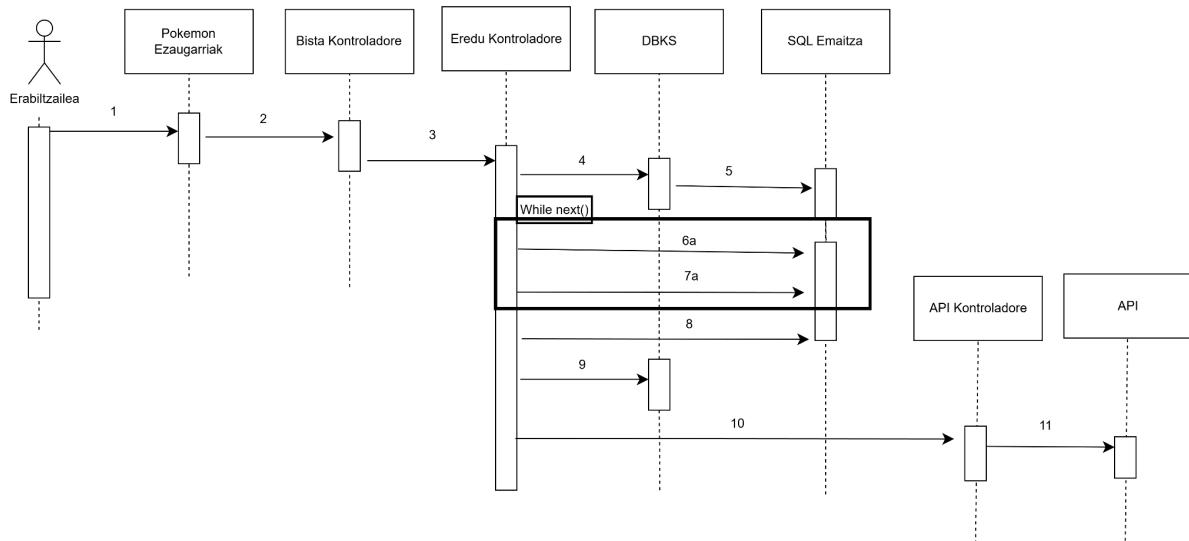
4a- Filtroa aukeratu eta bilaketa sakatzen du

5a- pokedexKargatu(JSON1): JSON2

6- Erabiltzaileak Pokemon baten argazkian klik egiten du.

7- new PokemonEzaugarriak(taldelezena, pokemonID)

8- bistaratuPokemon(pokemonID): JSON7



Irudia 156: Pokemon aukeratu funtzionalitateen sekuentzia diagrama

1- Erabiltzaileak "Hautatu" botoia sakatzen du.

2- sartu_taldera()

3- sartu_taldera(taldelezena, erabiltzailelzena, pokemon_id)

4- execSQL(sql_tamaina): SQL Emaitza

5- new SQL Emaitza()

[next() true den bitartean]

6a- next(): boolean

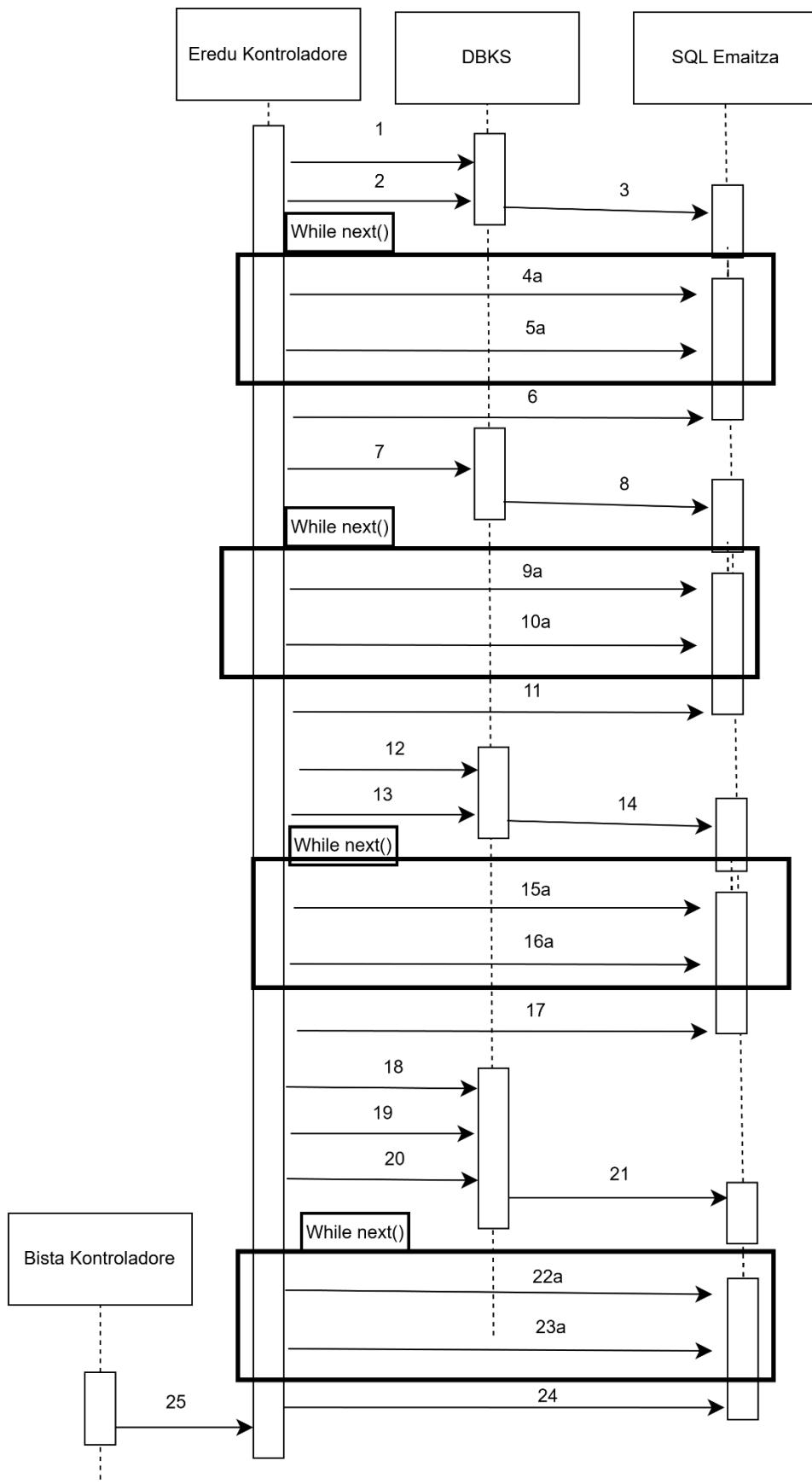
7a- getInt("count"): Int

8- close()

9- execSQL(sql_generoa)

10- hartu_stats(pokemonId)

11- pokemon(pokemonId): JSON9



Irudia 157: Pokemon sartu funtzionalitateen sekuentzia diagrama

1- execSQL(sql_harrapatu)
2- execSQL(sql_azkenharrapatu): SQL Emaitz
3- new SQL Emaitz()
[next() true den bitartean]
4a- next(): boolean
5a- getInt("id"): Int
6- close()
7- execSQL(sql_ablezi): SQL Emaitz
8- new SQL Emaitz()
[next() true den bitartean]
9a- next(): boolean
10a- getString("izena"): string
11- close()
12- execSQL(sql_dauka)
13- execSQL(sql_mugi): SQL Emaitz
14- new SQL Emaitz()
[next() true den bitartean]
15a- next(): boolean
16a- getString("mugilzena"): string
17- close()
18- execSQL(sql_mugimenduak)
19- execSQL(sql_taldera)
20- execSQL(sql_izena): SQL Emaitz
21- new SQL Emaitz()
[next() true den bitartean]
22a- next(): boolean
23a- getString("izena"): string
24- close()
25- notifikazioBerria(erabiltzailelzena, data, deskripzioa)

sql_taldeak: SELECT taldezena
FROM Taldea
WHERE erabiltzailelzena = %erabiltzailelzena%

sql_sortu: INSERT INTO
Taldea (taldezena, erabiltzailelzena)
VALUES (%taldezena%,%erabiltzailelzena%)

sql_poketalde: SELECT PT.harrapatuld, PP.irudia
FROM Taldea T
JOIN PokemonTaldean PT ON T.taldezena = PT.taldezena
JOIN PokemonTalde PKT ON PT.harrapatuld = PKT.harrapatuld

```
JOIN PokemonPokedex PP ON PKT.PokemonPokedexID = PP.pokeld  
WHERE T.erabiltzailelzena = %erabiltzailelzena% AND T.taldezena = %taldezena%
```

```
sql_tamaina: SELECT COUNT(*) as count  
FROM PokemonTaldean PT  
JOIN Taldea T ON PT.taldezena = T.taldezena  
WHERE T.erabiltzailelzena = %erabiltzailelzena% AND T.taldezena = %taldezena%
```

```
sql_generoa: SELECT generoa  
FROM PokemonPokedex  
WHERE pokeld = %pokeld%
```

```
sql_harrapatu: INSERT INTO  
PokemonTalde (izena, maila, adiskidetasun_maila, generoa, HP, ATK, SPATK, DEF,  
SPDEF, SPE, PokemonPokedexID, Erabiltzailelzena)  
VALUES (%izena%, 50, 0, %generoa%, %hp%, %atk%, %spatk%, %def%, %spdef%,  
%spe%, %pokeld%, %erabiltzailelzena%)
```

```
sql_azkenharrapatu: SELECT MAX(harrapatuld) as id  
FROM PokemonTalde  
WHERE Erabiltzailelzena = %erabiltzailelzena% AND PokemonPokedexID =  
%pokeld%
```

```
sql_abilezi: SELECT izena  
FROM IzanDezake  
WHERE pokemonPokedexID = %pokeld%
```

```
sql_mugi: SELECT mugilzena  
FROM IkerDezake  
WHERE pokemonPokedexID = %pokeld%
```

```
sql_dauka: INSERT INTO  
Dauka (abilezilzena, harrapatuld)  
VALUES (%abilezilzena%, %harrapatuld%)
```

```
sql_mugimenduak: INSERT INTO  
MugimendulzanTalde (harrapatuld, mugimendulzena)  
VALUES (%harrapatuld%, %mugimendulzena%)
```

```
sql_taldera: INSERT INTO  
PokemonTaldean (taldezena, harrapatuld, erabiltzailelzena)  
VALUES (%taldezena%, %harrapatuld%, %erabiltzailelzena%)
```

```
sql_izena: SELECT izena  
FROM PokemonTalde  
WHERE harrapatuld = %harrapatuld%
```

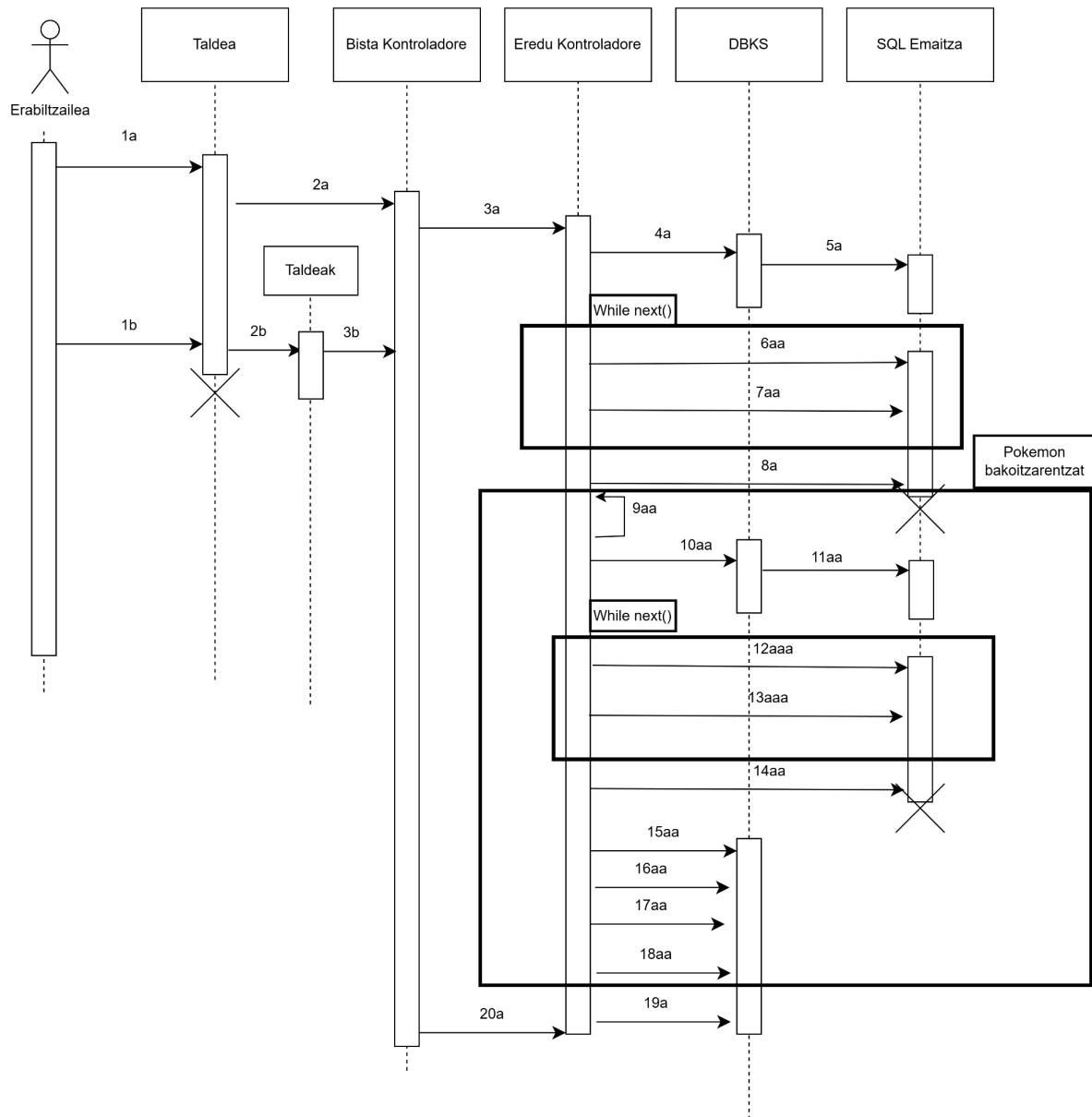
```
JSON1: {[izena: string, argazkia: string, ID: int]}  
JSON2: {[izena: string, motak: string[], generazioak: int[]}]  
JSON3: {[pokelID: int, argazkia: string]}
```

JSON4: {[{taldezena: String}]}

JSON7 = {izena: string, argazkia, image, deskripzioa: string, ID: int, abileziak: string[]}

JSON9 = {hp: int, atk: int, spatk: int, def: int, spdef: int, spe: int}

5.1.2.2 Taldea gorde eta ezabatu



Irudia 158: Taldea gorde eta ezabatu funtzionalitatearen sekuentzia diagrama

[Erabiltzaileak taldea ezabatu nahi badu]

1a- Erabiltzaileak "Ezabatu" botoia sakatzen badu

2a- taldea_ezabatu()

3a- ezabatu_taldea(taldezena, erabiltzailezena)

4a- execSQL(sql_pokemonguztiak): SQL Emaitza

5a- new SQL Emaitza()

[next() true den bitartean]

6aa- next()

7aa- getInt("harrapatuld"): Int

8a- close()

[Taldean dauden pokemon bakoitzarentzako]

9aa- ezabatu_taldetik(taldezena, erabiltzailezena, pokemonId): String

10aa- execSQL(sql_ezabatulzena): SQL Emaitza

11aa- new SQL Emaitza()

[next() true den bitartean]

12aaa- next()

13aaa- getString("izena"): String

14aa- close()

15aa- execSQL(sql_mugiezabatu)

16aa- execSQL(sql_ablezabatu)

17aa- execSQL(sql_taldekanpo)

18aa- execSQL(sql_askatu)

19a- execSQL(sql_taldeaezabatu)

20a- notigikazioBerria(erabiltzailezena, data, deskripzioa)

[Erabiltzaileak taldea gorde nahi badu]

1b- Erabiltzaileak "Gorde" botoia sakatzen badu

2b- new Taldeak()

3b- taldeak_kargatu():JSON4 (Taldea sortu inplementatuta)

sql_pokemonguztiak = SELECT harrapatuld FROM PokemonTaldean
WHERE taldezena = %taldezena% AND erabiltzailezena = %erabiltzailezena%

sql_ezabatulzena = SELECT izena FROM PokemonTalde
WHERE harrapatuld = %harrapatuld%

sql_mugiezabatu = DELETE FROM MugimendulzanTalde
WHERE harrapatuld = %harrapatuld%

sql_ablezabatu = DELETE FROM Dauka
WHERE harrapatuld = %harrapatuld%

sql_taldekanpo = DELETE FROM PokemonTaldean

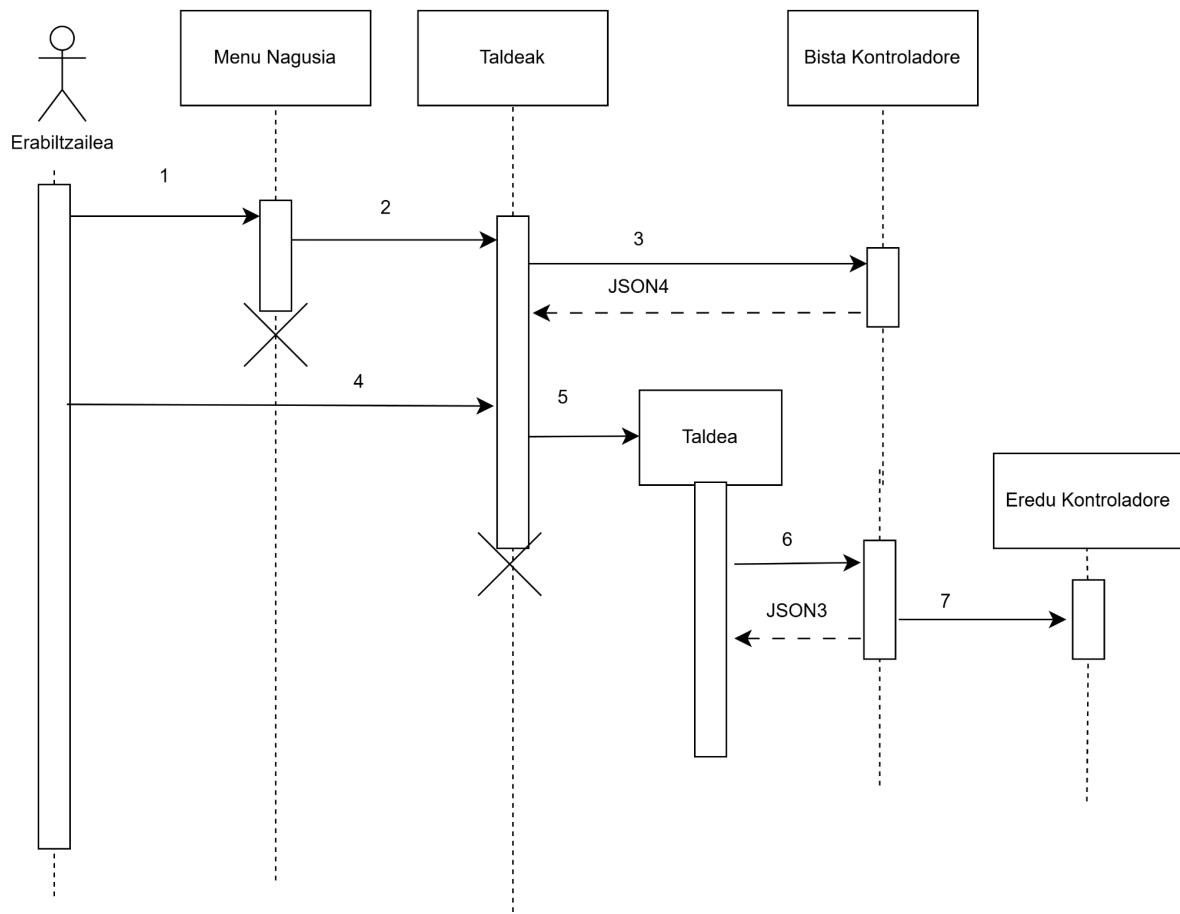
WHERE harrapatuld = ? AND taldelzena = %taldelzena% AND erabiltzailelzena = %erabiltzailelzena%

sql_askatu = DELETE FROM PokemonTalde
WHERE harrapatuld = %harrapatuld%

sql_taldeaezabatu = DELETE FROM Taldea
WHERE taldelzena = %taldelzena% AND erabiltzailelzena = %erabiltzailelzena%

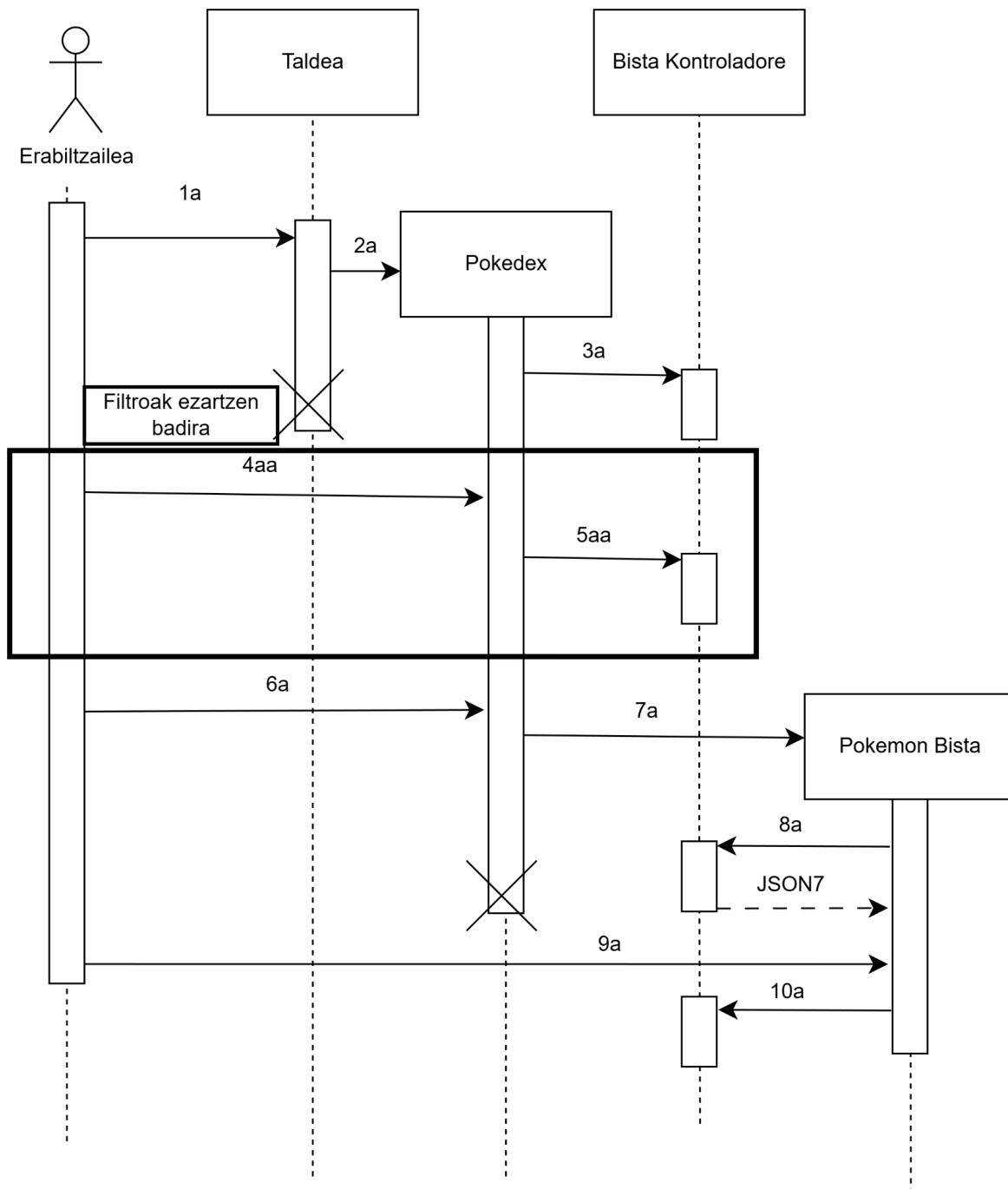
JSON4: {[{taldelzena: String}]}

5.1.2.3 Taldea editatu



Irudia 159: Taldea editatu funtzionalitatearen sekuentzia diagrama

- 1- Erabiltzaileak Menu Nagusian "Taldeak" botoia sakatzen du
- 2- new Taldeak()
- 3- taldeak_kargatu(): JSON4
- 4- Erabiltzailea "EDIT" botoia sakatzen du.
- 5- new Taldea()
- 6- taldea(taldezena): JSON3
- 7- notifikazioBerria(erabiltzailelzena, data, deskripzioa)



Irudia 160: Taldea editatu sekuentzia diagrama (Pokemon Gehitu)

[Erabiltzailea Gorde edo Ezabatu klikatzen ez duen bitartean hurrengoa errepikatu]

[Erabiltzailea "+" botoia sakatzen badu]

1a- Erabiltzailea "+" botoia sakatzen du.

2a- new Pokedex()

3a- pokedexKargatu(JSON1): JSON2

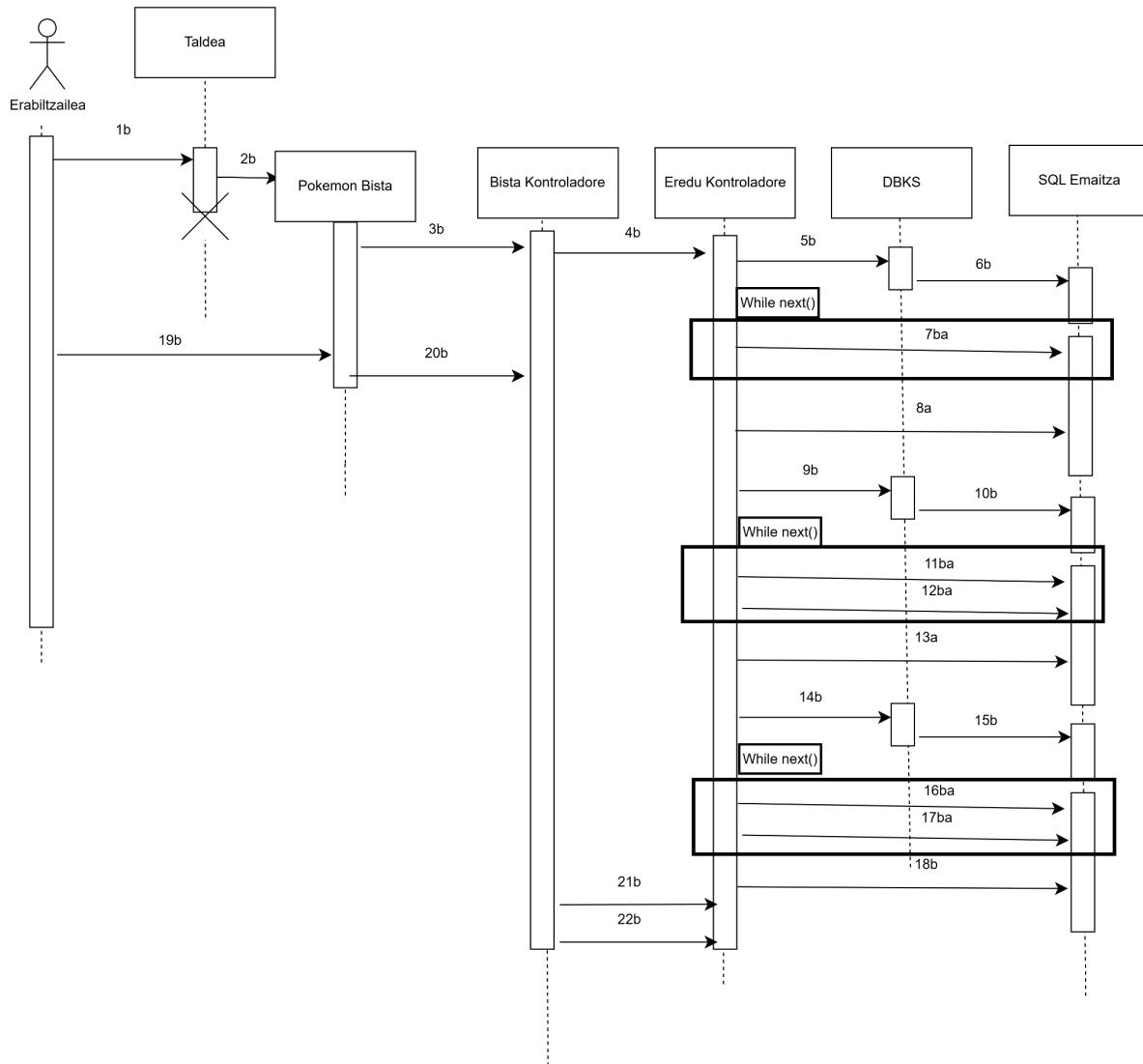
[Erabiltzaileak filtroak ezartzen baditu]

4aa- Filtroa aukeratu eta bilaketa sakatzen du

5aa- pokedexKargatu(JSON2): JSON1

6a- Erabiltzaileak Pokemon baten argazkian klik egiten du.

- 7a- new PokemonBista(pokemonID)
 8a- bistaratuPokemon(pokemonID): JSON7
 9a- Erabiltzaileak "Hautatu" botoia sakatzen du.
 10a- sartu_taldera()



Irudia 161: Taldea editatu sekuentzia diagrama (Pokemon Aldatu edo Ezabatu)

[Erabiltzailea Gorde edo Ezabatu klikatzen ez duen bitartean hurrengoa errepikatu]

[Erabiltzaileak Pokemon baten argazkian sakatzen badu]

1b- Erabiltzaileak Pokemon baten argazkian sakatzen du

2b- new PokemonEzaugarriak(pokemonID)

3b- bistaratuPokemon(pokemonId): JSON6

4b- bistaratu_pokemon_taldea(pokemonID): JSON6

5b- execSQL(sqlPokeBista): SQL Emaitza

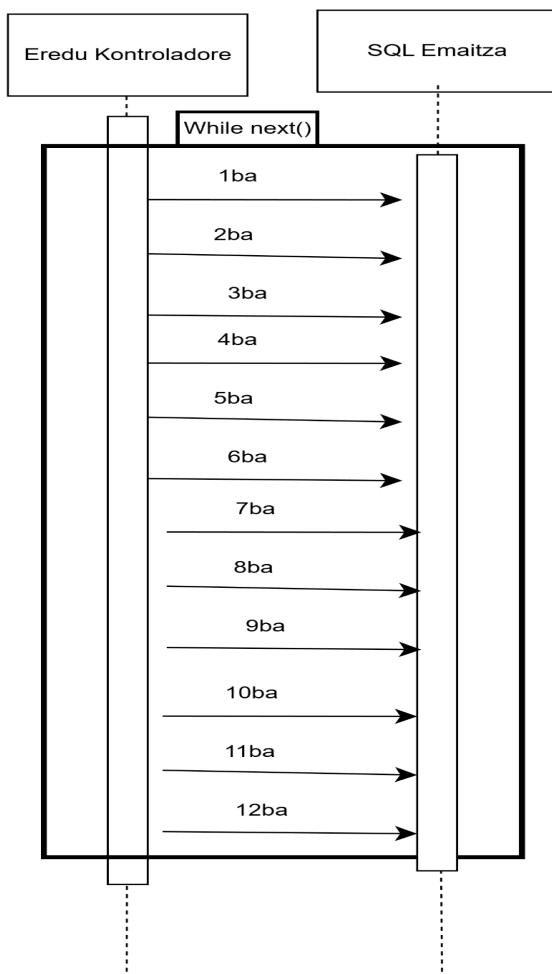
6b- new SQL Emaitza()

[next() true den bitartean]

7ba- next(): boolean

[Instrukzioak hurrengo diagraman]

8b- close()
 9b- execSQL(sqlAbil): SQL Emaitza
 10b- new SQL Emaitza()
 [next() true den bitartean]
 11ba- next(): boolean
 12ba- getString("abilezilzena"): String
 13b- close()
 14b- execSQL(sqlMugi): SQL Emaitza
 15b- new SQL Emaitza()
 [next() true den bitartean]
 16ba- next(): boolean
 17ba- getString("mugimendulzena"): String
 18b- close()
 [Erabiltzaileak "Ezabatu" edo "Aldatu" botoia sakatzen badu]
 19b- Erabiltzaileak "Ezabatu" botoia sakatzen du
 20b- ezabatu_talde()
 21b- ezabatu_talde(taldezena, erabiltzailezena, pokemonId): String
 (Implementazioa gorde eta ezabatu diagraman)
 22b- notifikazioBerria(erabiltzailezena, data, deskripzioa)
 [Ezabatu aukeratu bada taldera bueltatzen da]
 [Gorde aukeratu bad, aurreko diagrama jarraitu behar da]



1ba- getInt("harrapatuld"): Int
 2ba- getString("izena"): String
 3ba- getString("argazkia"): String
 4ba- getInt("id"): Int
 5ba- getfloat("pisua"): float
 6ba- getfloat("altuera"): float
 7ba- getInt("hp"): Int
 8ba- getInt("atk"): Int
 9ba- getInt("spatk"): Int
 10ba- getInt("def"): Int
 11ba- getInt("spdef"): Int
 12ba- getInt("spe"): Int

sql_ezabatulzena = SELECT izena FROM PokemonTalde
 WHERE harrapatuld = %harrapatuld%

sql_mugiezabatu = DELETE FROM MugimendulzanTalde
 WHERE harrapatuld = %harrapatuld%

sql_ablezabatu = DELETE FROM Dauka
 WHERE harrapatuld = %harrapatuld%

Irudia 162: Taldea editatu sekuentzia diagrama (Pokemon Bistaratzeko informazioa lortu)

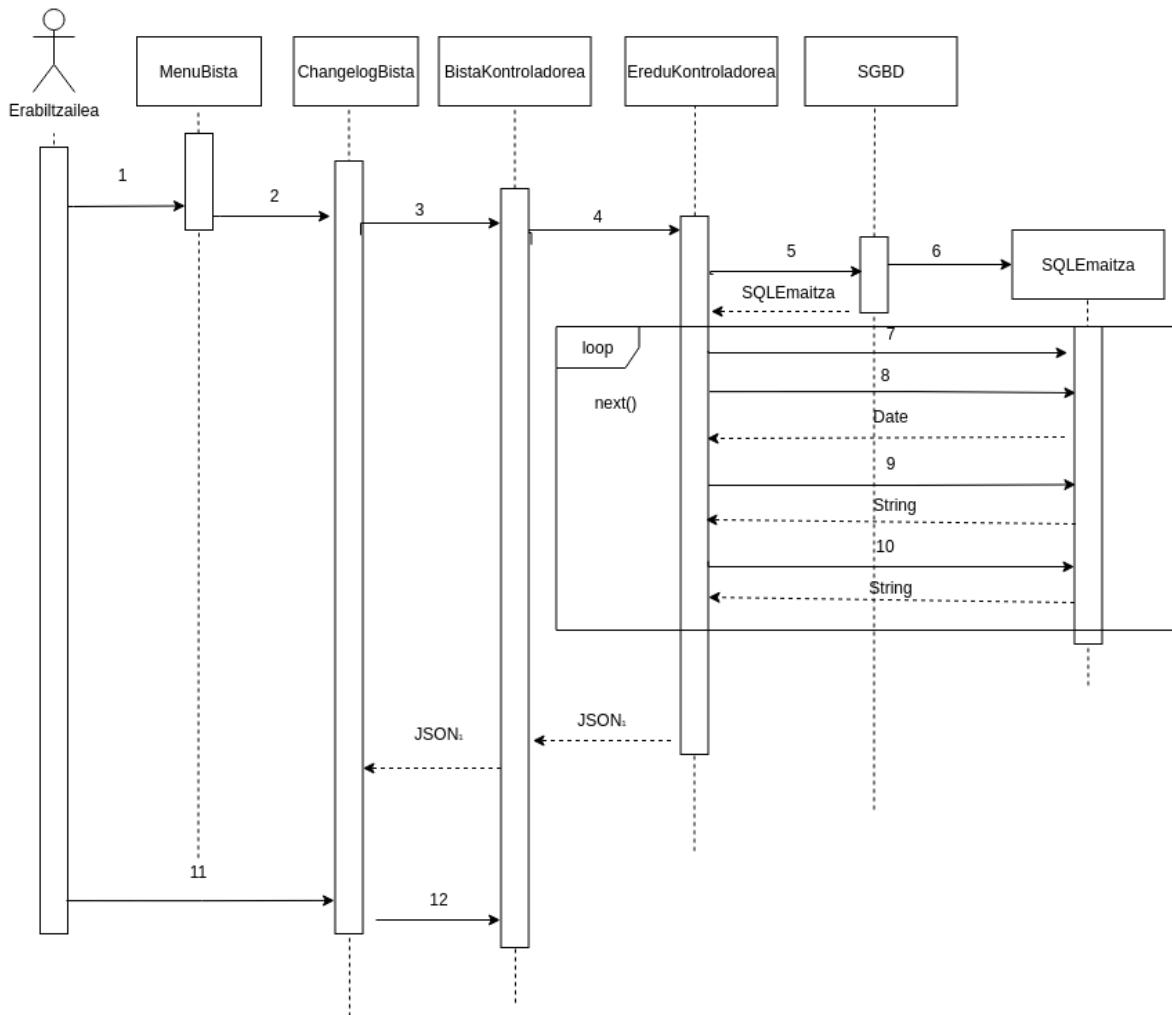
```
sql_taldekanpo = DELETE FROM PokemonTaldean  
WHERE harrapatuld = %harrapatuld% AND taldelzena = %taldelzena% AND  
erabiltzailezena = %erabiltzailezena%
```

```
sql_askatu = DELETE FROM PokemonTalde  
WHERE harrapatuld = %harrapatuld%
```

```
JSON1: {[izena: string, argazkia: string, ID: int]}  
JSON2: {[izena: string, motak: string[], generazioak: int[]}]}  
JSON3: {[pokelID: int, argazkia: string]}  
JSON4: {[{taldelzena: String}]}  
JSON6: {[{harrapatuld: int, izena: String, argazkia: String, id: int, pisua: float, altuera: float, hp: int, atk: int, spatk: int, def: int, spdef: int, spe: int, abileziak: string, mugilzenak: [String]}]}  
JSON7 = {izena: string, argazkia, image, deskripzioa: string, ID: int, abileziak: string[]}
```

5.1.3 Changelog-a (Unai Rodriguez)

5.1.3.1 Notifikazioak kargatu izen baten menpe (edo ez)



Irudia 163: Notifikazioak kargatu izen baten menpe funtzionalitatearen sekuentzia diagrama

1. Erabiltzailea changelog botoia sakatzen du menuBistan
 2. new ChangelogBista(erabiltzailezena, **bilatutakolzena***)
 3. notifikazioenInformazioaLortu(erabillzena, bilatutakolzena): JSON1
 4. notifikazioenInformazioaLortu(erabiltzailezena, bilatutakolzena): JSON1
 5. execSQL(SQL1): SQLEmaitza
 6. new SQLEmaitza(): SQLEmaitza
 7. next()
 8. getDate("DataOrdua"): Date
 9. getString("deskripzioa"): String
 10. getString("jarraitulzena"): String
 11. Erabiltzailea izen bat idaztend du bilaketa filtroan eta bilatu botoiari ematen du
 12. notifikazioenInformazioaLortu(erabillzena, **bilatutakolzena***): JSON1

SQL1: "SELECT J.Jarraitulzena, N.DataOrdua, N.deskripzioa FROM JarraitzenDu J
JOIN Notifikatu N ON J.Jarraitulzena = N.Erabiltzailelzena WHERE
J.Jarraitzailelzena = ? AND J.Notifikatu = 1 **AND J.Jarraitulzena** LIKE ?** ORDER
BY N.DataOrdua DESC;"

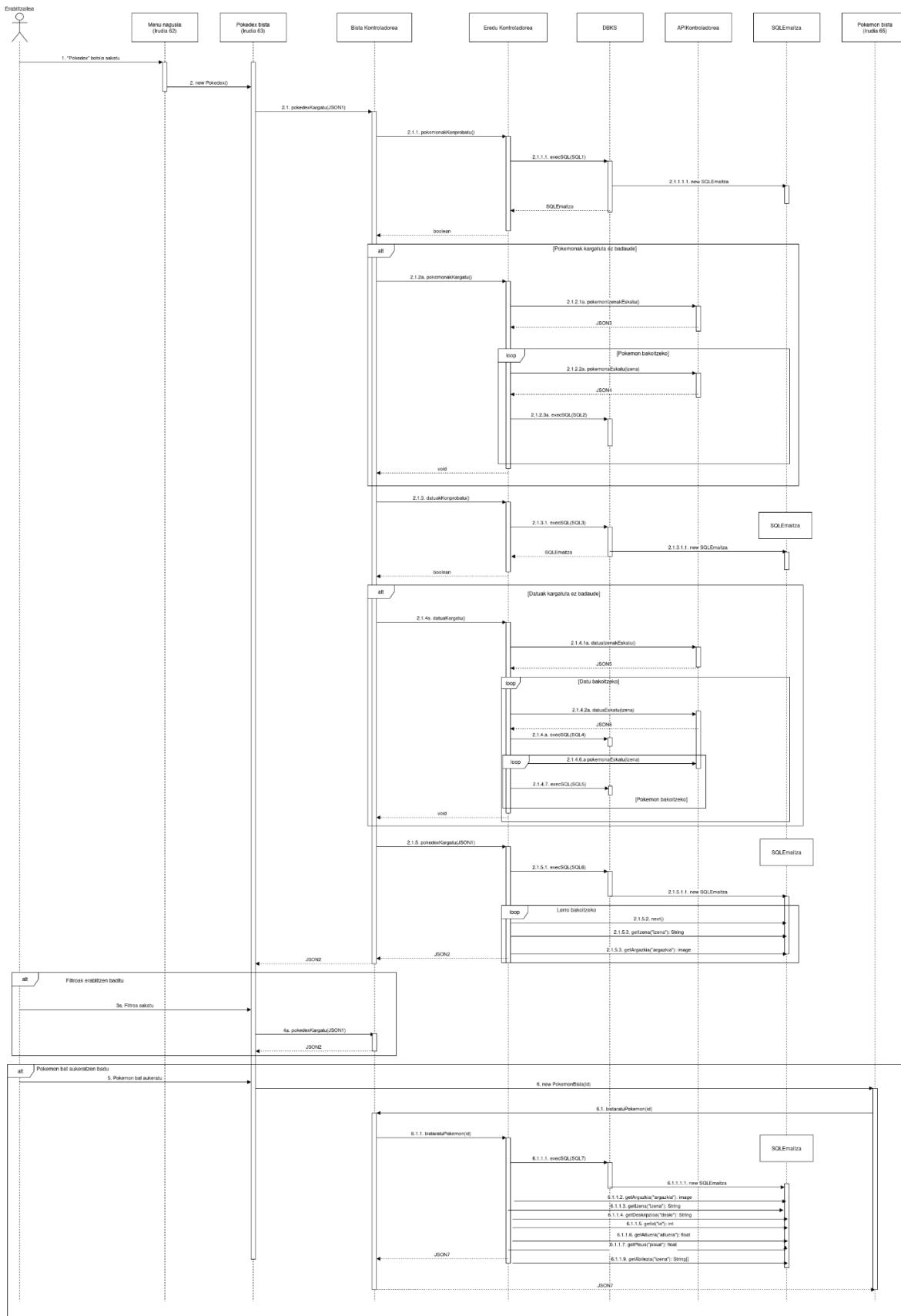
JSON1: {[{DataOrdua: Date, deskripzioa: String, jarraitulzena: String}...{}]}

Oħarrak:

* bilatutakolzena aldagaia 2. eta 12. deian desberdina da. Bigarren deian NULL
balioa izango du erabiltzailea menu nagusitik changelog-era mugitzen delako,
hori egiten du ezinezkoa izatea erabiltzailea zeozer idaztea bilatu filtroan

** Query-an agertzen den zati hori, if baten ondorioz gehitzen da ala ez,
komunikazio diagraman bezala. Erabiltzaileak zeozer idatzi baldin badu bilatzeko
filtroan exekutatuko den query-a beltzez agertzen den zatiarekin batera
exekutatuko da. Aldiz, erabiltzailea ez badu ezer idazten, zati hori query-tik
baztertuko da

5.1.4 Pokedex-a (Yassin Mehani)

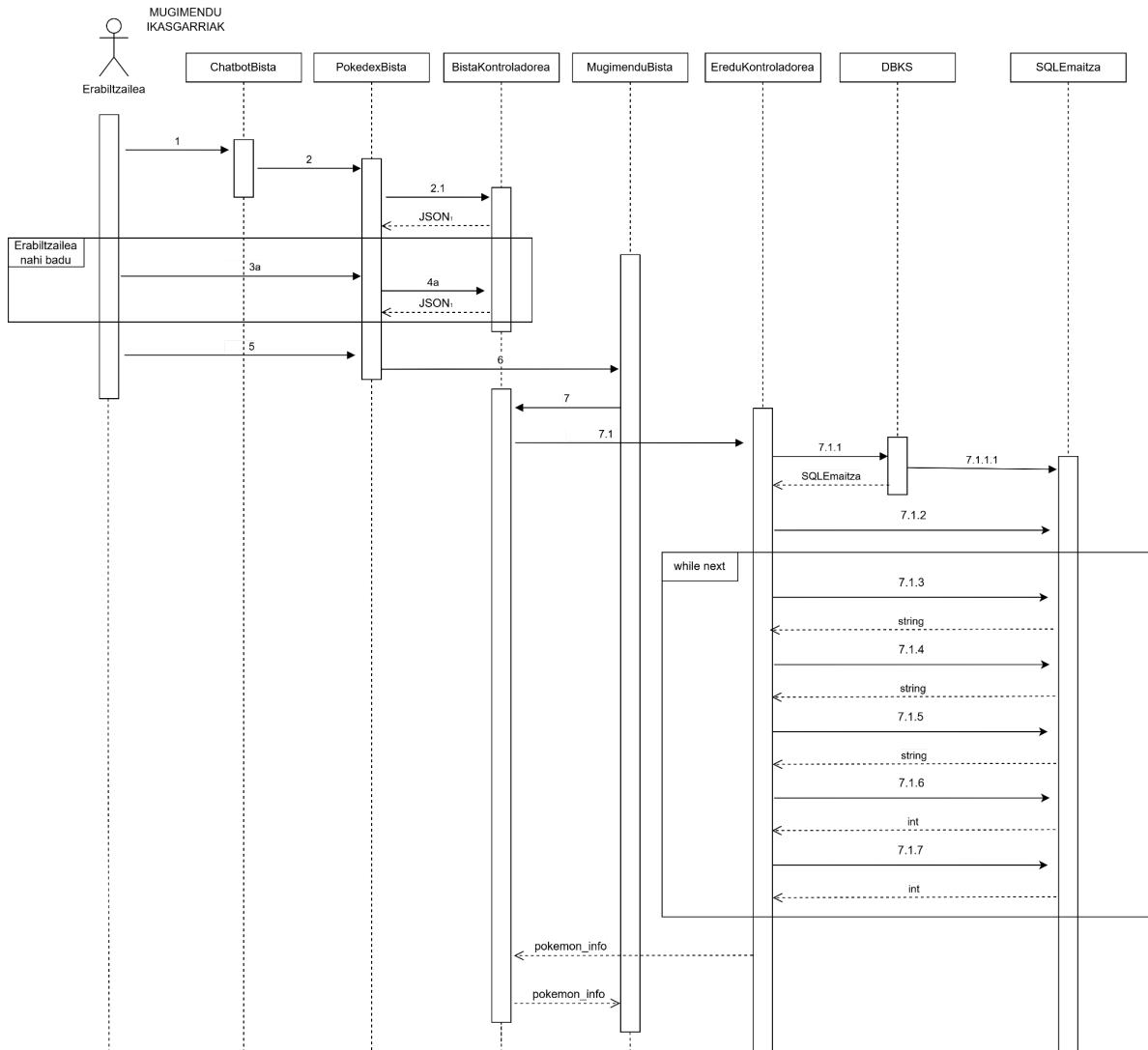


Irudia 164: Pokedex-a funtzionalitatearen sekuentzia diagrama

JSON1 = {izena: String, motak: String[], generazioak:[]}
JSON2 = [{id: int, izena: String, irudia: Image}]
JSON3 = [{izena:string, id: int}]
JSON4 = {id: int, izena: String, altuera: float, pisua: float, generoa: String, deskripzioa: String, irudia: Image, generazia: int, preEboluziold: int}
JSON5 = [{izena:string, id: int}]
JSON6: JSON honek aldaketak ditu eskatzen den datu-motaren arabera
 JSON6.1(Mota) = {izena: string, id: int, pokemonak: [{id: int, izena: string}]}
 JSON6.2(Abilezia) = {izena: string, id: int, deskripzioa: pokemonak = [{id: int, izena: string, ezkutua: boolean}]}
 JSON6.3(Mugimendua) = {izena: string, potentzia: int, pp: int, efektua: string, pokemonMotalzena: string, pokemonak: [{id: int, izena: string}]}
JSON7 = {izena: string, argazkia, image, deskripzioa: string, ID: int, abileziak: string[]}
SQL1 = SELECT * FROM PokemonPokedex
SQL2 = INSERT INTO PokemonPokedex (pokeld, izena, altuera, pisua, generoa, deskripzioa, irudia, generazia, preEboluziold)
SQL3 = SQL honek aldaketak ditu insertazen den datu-motaren arabera
 SQL3.1(Mota) = SELECT * FROM MotaPokemon
 SQL3.2(Abilezia) = SELECT izena FROM Abilezia
 SQL3.3(Mugimendua) = SELECT * FROM Mugimendua
SQL4 = SQL honek aldaketak ditu insertatzen den datu-motaren arabera
 SQL4.1(Mota) = INSERT OR IGNORE INTO MotaPokemon (pokemonMotalzena, irudia)
 SQL4.2(Abilezia) = INSERT OR IGNORE INTO Abilezia (izena, deskripzioa)
 SQL4.3(Mugimendua) = INSERT OR IGNORE INTO Mugimendua (izena, potentzia, zehatzazuna, PP, efektua, pokemonMotalzena)
 SQL5 = SQL honek aldaketak ditu insertatzen den datu-motaren arabera
 SQL5.1(Mota) = INSERT OR IGNORE INTO DaMotaPokemon (motalzena, pokemonID)
 SQL5.2(Abilezia) = INSERT OR IGNORE INTO IzanDezake (pokemonPokedexID, izena, ezkutua)
 SQL5.3(Mugimendua) = INSERT OR IGNORE INTO IkasDezake (pokedexId, mugilzena)
 SQL6 = SELECT izena, irudia, pokeld
 FROM PokemonPokedex P
 JOIN DaMotaPokemon DM ON P.pokeld = DM.pokemonID
 JOIN MotaPokemon M on DM.motalzena = M.pokemonMotalzena
 WHERE P.izena Like %izena% (% gehigarri bat dago, erabiltzaileak izen ez oso bat ipintzen badu)
 AND P.generazioa IN ({generazioak})
SQL7 = SELECT izena, pokeld, argazkia, deskripzioa FROM Pokemon(Pokedex) P
WHERE pokeld = %id%
SQL8 = SELECT izena FROM IzanDezake WHERE pokemonPokedexID = %id%

5.1.5 Chatbot-a (Xinyan Wang)

5.1.5.1 Mugimendu ikasgarriak



Irudia 165: Mugimendu ikasgarriak funtzionalitatearen sekuentzia diagrama

- 1- "MUGIMENDUAK" botoia sakatzen du.
- 2- new Pokedex()
- 2.1- pokedexKargatu(JSON2): JSON1
[Erabiltzaileak nahi badu]
 - 3a- filtro edo bilaketa sakatzen du.
 - 4a- pokedexKargatu(JSON2): JSON1
- 5- Erabiltzaileak pokemon bat aukeratzen du.
- 6- new MugimenduBista(id)
- 7- getMugimendulkasgarriak(id): pokemon_info
 - 7.1- getMugimendulkasgarriak(id): pokemon_info
 - 7.1.1- execSQL(SQL): SQLEmaitza

7.1.1.1- new SQLEmaitza()
7.1.2- next()
7.1.3- getString("pokemon_izena"): string
7.1.4- getString("pokemon_irudia"): string
7.1.5- getString("mugimendu_izena"): string
7.1.6- getInt("potentzia"): int
7.1.7- getInt("zehaztasuna"): int

JSON1: [{id: int, izena: string, irudia:string}]

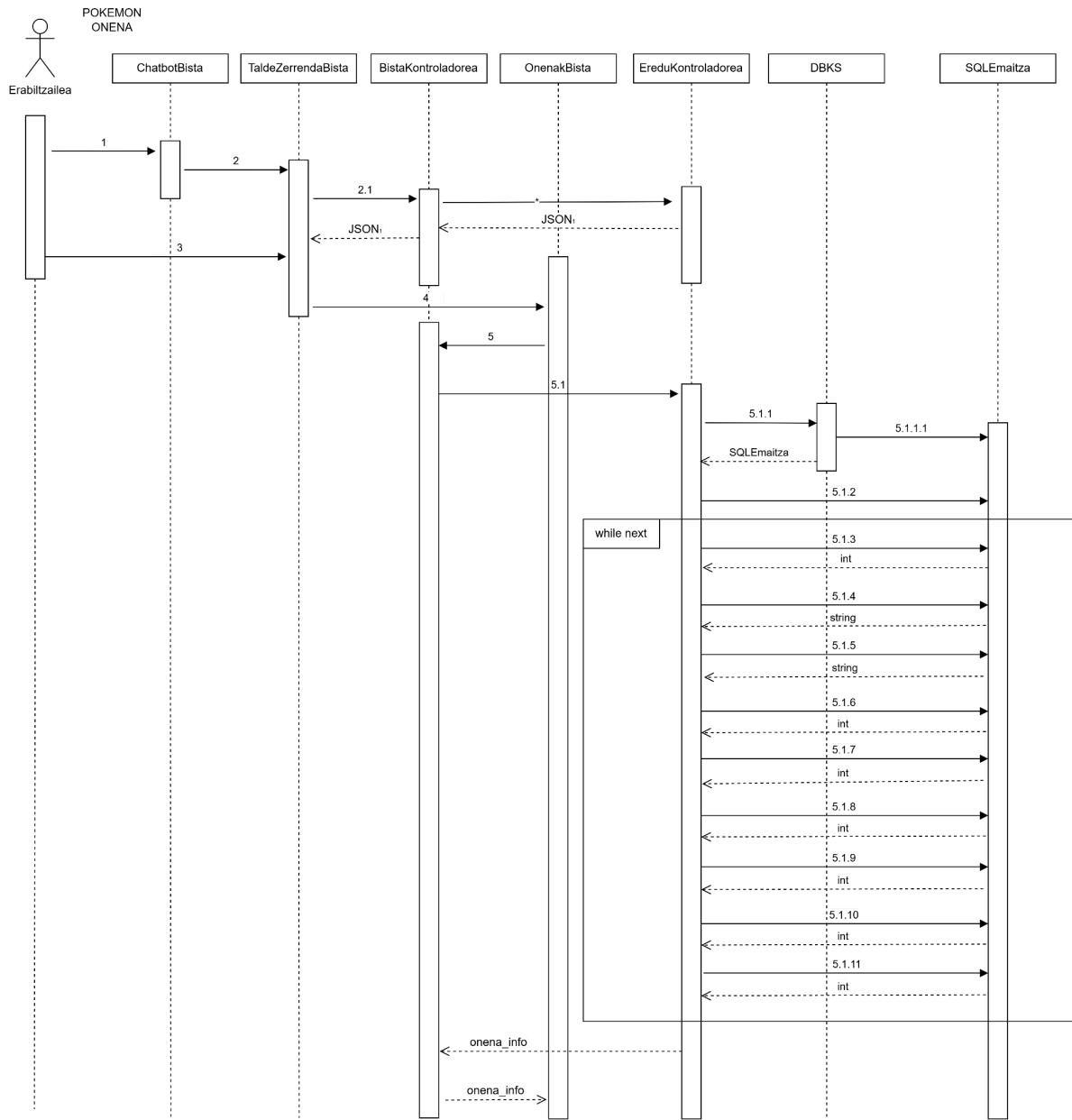
JSON2: {izena: string, motak: string[], generazioak: int[]}

pokemon_info: {izena: string, irudia: string, mugimenduak:[{mugimendulzena: string, potentzia: int, zehaztasuna: int}]}]

SQL1:

```
SELECT p.izena AS pokemon_izena,
       p.irudia AS pokemon_irudia,
       m.izena AS mugimendu_izena,
       m.potentzia,
       m.zehaztazuna
  FROM PokemonPokedex p
    LEFT JOIN IkasDezake i ON p.pokeld = i.pokedexId
    LEFT JOIN Mugimendua m ON i.mugilzena = m.izena
 WHERE p.pokeld = %pokemonID%;
```

5.1.5.2 Taldeko pokemon hoberena



Irudia 166: Taldeko pokemon hoberena funtzionalitatearen sekuentzia diagrama

1. "ONENAK" botoia sakatzen du.
 2. new TaldeZerrenda()
 - 2.1 taldeZerrenda(): JSON1
 - * taldeak_kargatu(erabiltzailelzena): JSON1
 3. Erabiltzaileak talde bat aukeratzen du.
 4. new Onenak(JSON2)
 5. getOnenak(JSON2): onena_info
 - 5.1 getOnenak(JSON2): onena_info

5.1.1 execSQL(SQL1): SQLEmaitza
 5.1.1.1 new SQLEmaitza()
 5.1.2 next()
 5.1.3 getInt("PokemonPokedexID"): int
 5.1.4 getString("izena"): string
 5.1.5 getInt("HP"): int
 5.1.6 getInt("ATK"): int
 5.1.7 getInt("DEF"): int
 5.1.8 getInt("SPATK"): int
 5.1.9 getInt("SPDEF"): int
 5.1.10 getInt("SPE"): int
 5.1.11 getString("irudia"): string

JSON1: {[{taldezena: String}]}

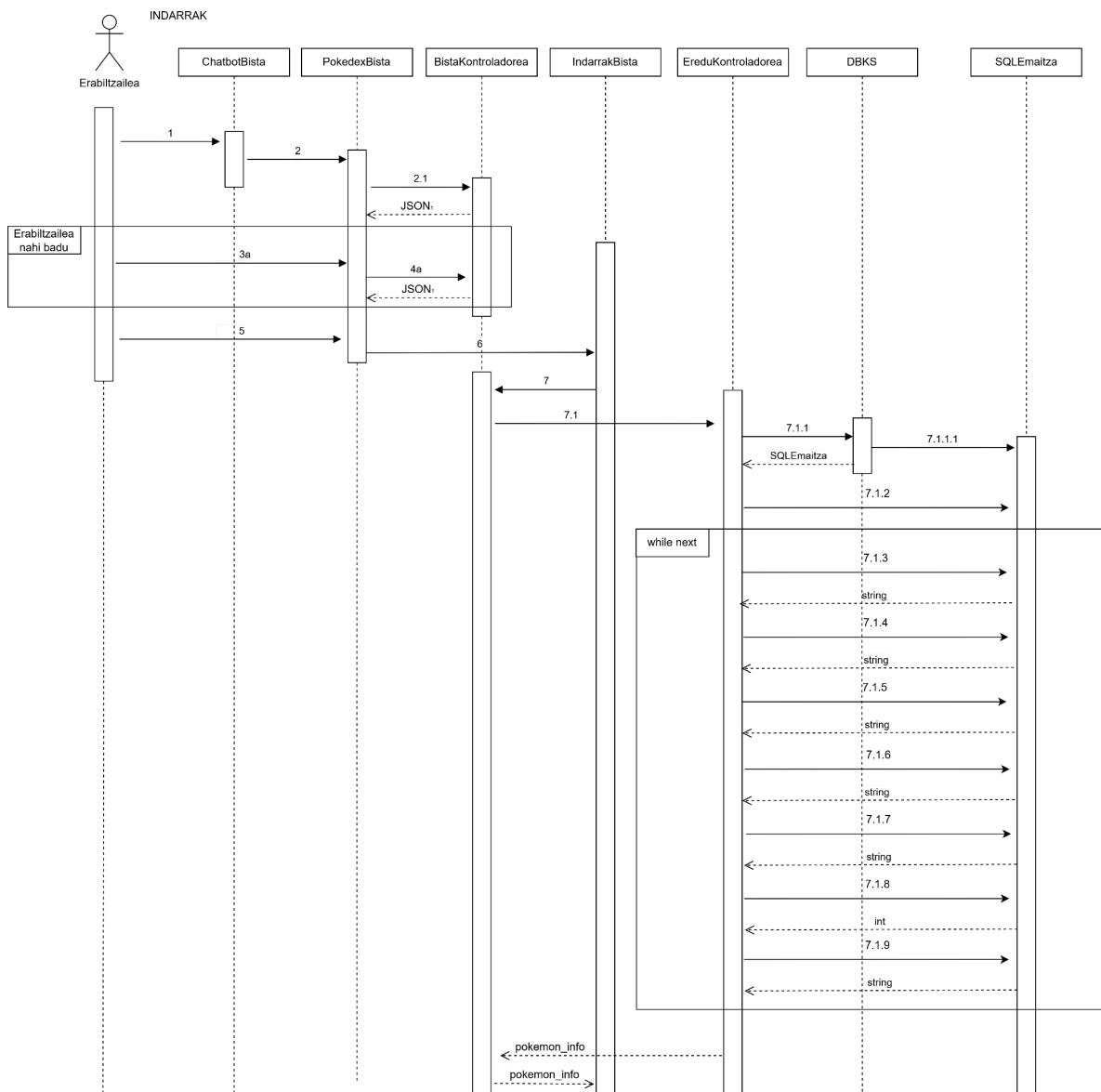
JSON2: {taldezena: string, erabiltzailezena: string}

onena_info: {PokemonPokedexID:int, izena: string, irudia: string, HP: int, ATK: int, DEF: int, SPATK: int, SPDEF: int, SPE: int, puntuazioa: int, taldezena: string}

SQL1:

```
SELECT pt.PokemonPokedexID,
       pt.izena,
       pt.HP,
       pt.ATK,
       pt.DEF,
       pt.SPATK,
       pt.SPDEF,
       pt.SPE,
       pd.irudia
  FROM Taldea t
 INNER JOIN PokemonTaldean pta ON pta.taldezena = t.taldezena AND
 pta.erabiltzailezena = t.erabiltzailezena
 LEFT JOIN PokemonTalde pt ON pt.harrapatuld = pta.harrapatuld AND
 pt.erabiltzailezena = t.erabiltzailezena
 LEFT JOIN PokemonPokedex pd ON pt.PokemonPokedexID = pd.pokeld
 WHERE t.taldezena = %taldezena% AND t.erabiltzailezena = %erabiltzailezena%;
```

5.1.5.3 Indarrak eta ahuleziak



Irudia 167: Indarrak eta ahuleziak funtzionalitatearen sekuentzia diagrama

1. "INDARRAK" botoia sakatzen du
 2. new Pokedex()
 - 2.1 pokedexKargatu(JSON2): JSON1
[Erabiltzaileak nahi badu]
 - 3a. filtro edo bilaketa sakatzen du.
 - 4a pokedexKargatu(JSON2): JSON1
 5. Erabiltzaileak pokemon bat aukeratzen du.
 6. new Indarrak(id)
 7. getIndarrak(id): pokemon_info
 - 7.1 getIndarrak(id): pokemon_info

```
7.1.1 execSQL(SQL1): SQL emaitza
    7.1.1.1 newSQLemaitza()
    7.1.2 next()
    7.1.3 getString("izena"): string
    7.1.4 getString("irudia"): string
    7.1.5 getString("mota_izena"): string
    7.1.6 getInt("mota_irudia"): string
    7.1.7 getString("pokemonMotaEraso"): string
    7.1.8 getString("multiplikatzaila"): int
    7.1.9 getString("erasoko_mota_irudia"): string
```

JSON1: [{id: int, izena: string, irudia:string}]

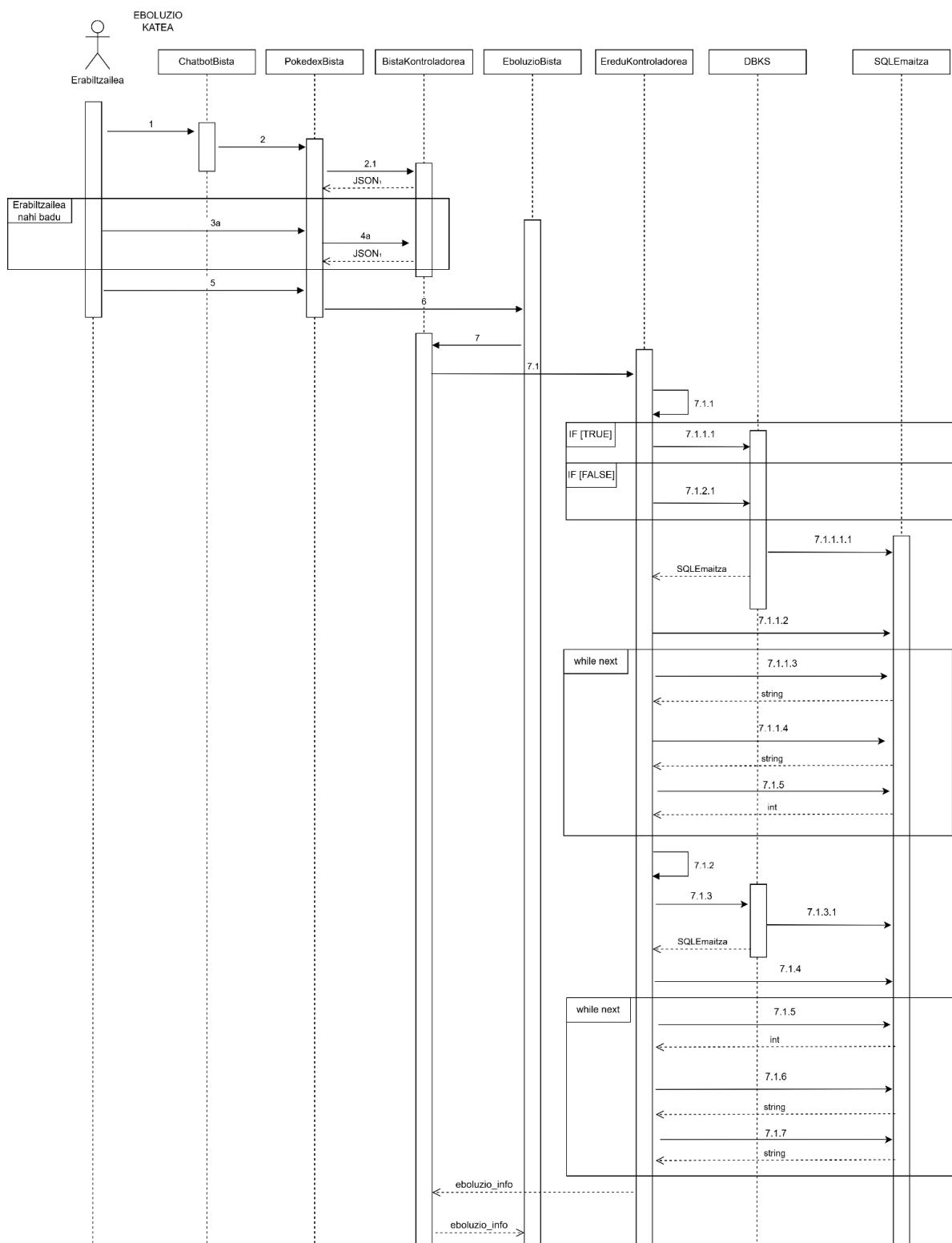
JSON2: {izena: string, motak: string[], generazioak: int[]}

pokemon_info: {izena: string, irudia: string, pokemon_motak:[{izena:string, irudia:string}], Ahuleziak:[{izena: string, irudia: string}], Indarrak:[{izena: string, irudia: string}]}]

SQL1:

```
SELECT p.izena,
       p.irudia,
       t.pokemonMotalzena AS mota_izena,
       t.irudia AS mota_irudia,
       m.pokemonMotaEraso,
       m.multiplikatzaila,
       t2.irudia AS erasoko_mota_irudia
FROM PokemonPokedex p
      LEFT JOIN DaMotaPokemon d ON p.pokeld = d.pokemonID
      LEFT JOIN MotaPokemon t ON d.Motalzena = t.pokemonMotalzena
      LEFT JOIN Multiplikatzaila m ON t.pokemonMotalzena =
          m.pokemonMotaJaso
      LEFT JOIN MotaPokemon t2 ON m.pokemonMotaEraso =
          t2.pokemonMotalzena
WHERE p.pokeld = %pokemonId%;
```

5.1.5.4 Eboluzio katea



Irudia 168: Eboluzio kateafuntzionalitatearen sekuentzia diagrama

1. "EBOLUZIOA" botoia sakatzen du.
2. new Pokedex()
- 2.1 pokedexKargatu(JSON2): JSON1

[Erabiltzaileak nahi badu]

3a filtro edo bilaketa sakatzen du.

4a pokedexKargatu(JSON2): JSON1

5. Erabiltzaileak pokemon bat aukeratzen du.

6. new EboluzioBista(id)

7. getEboluzioa(id): eboluzio_info

7.1 getEboluzioa(id):eboluzio_info

7.1.1 bilatu_eboluzioak(id, aurrekoak=True): emaitza

7.1.1.1 execSQL(SQL1): SQLemaitza

7.1.1.1.1 new SQLemaitza()

7.1.1.2 next()

7.1.1.3 getInt("pokeld"): int

7.1.1.4 getString("irudia"): string

7.1.1.5 getString("izena"): string

7.1.2 bilatu_eboluzioak(id, aurrekoak=False): emaitza

7.1.2.1 execSQL(SQL2): SQLemaitza

7.1.2.1.1 new SQLemaitza()

7.1.2.2 next()

7.1.2.3 getInt("pokeld"): int

7.1.2.4 getString("irudia"): string

7.1.2.5 getString("izena"): string

7.1.3 execSQL(SQL3): SQLemaitza

7.1.3.1 new SQLemaitza()

7.1.4 next()

7.1.5 getInt("pokeld"): int

7.1.6 getString("irudia"): string

7.1.7 getString("izena"): string

JSON1: [{id: int, izena: string, irudia:string}]

JSON2: [{izena: string, motak: string[], generazioak: int[]}]

emaitza: [{pokeld: int, izena: string, irudia:string}]

eboluzio_info: {izena: string, irudia: string, aurrekoak: emaitza, hurrengoak: emaitza}

SQL1:

SELECT P1.pokeld, P1.izena, P1.irudia

FROM PokemonPokedex P1

JOIN PokemonPokedex P2 ON P1.pokeld = P2.preEboluzioid

WHERE P2.pokeld= %pokemonID%;

SQL2:

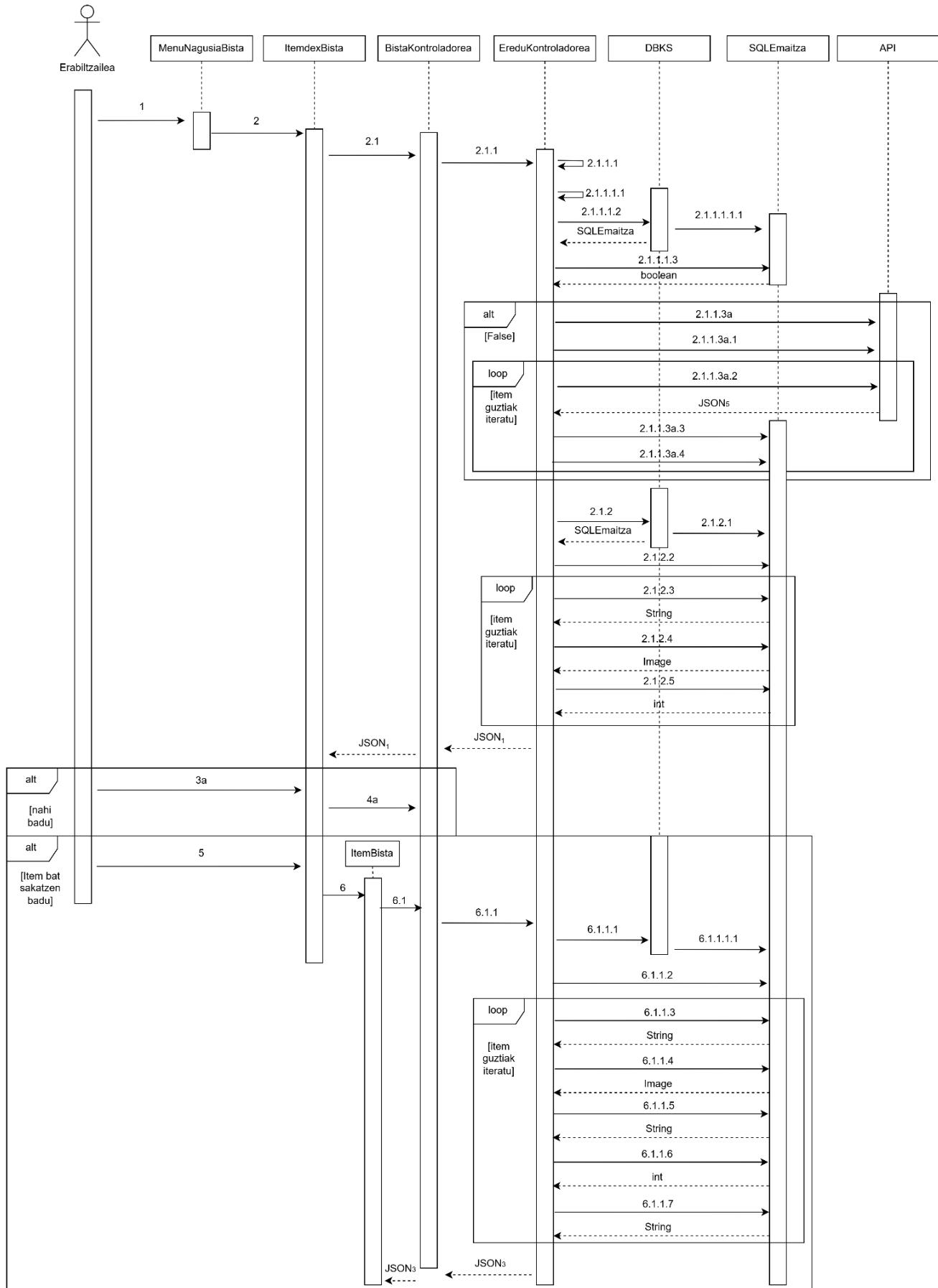
SELECT P2.pokeld, P2.izena, P2.irudia

```
FROM PokemonPokedex P1
JOIN PokemonPokedex P2 ON P1.pokeld = P2.preEboluziold
WHERE P1.pokeld = %pokemonID%;
```

SQL3:

```
SELECT izena, irudia FROM PokemonPokedex WHERE pokeld = %pokemonID%;
```

5.1.6 Itemdex (Yoel Justel)



Irudia 169: Itemdex funtzionalitatearen sekuentzia diagrama

1. "Itemdex" botoia sakatzen du
2. new Itemdex()
 - 2.1. itemdex_kargatu(JSON2): JSON1
 - 2.1.1 itemdex_kargatu(JSON2): JSON1
 - 2.1.1.1. itemak_konprobatu(): boolean
 - 2.1.1.1.1: kendumazentuak(): String
 - 2.1.1.1.2. execSQL(SQL4): SQLEmaitza
 - 2.1.1.1.2.1 new SQLEmaitza
 - 2.1.1.1.3. next(): boolean
 - [Item-ak Datu Basean kargatuta ez badaude]
 - 2.1.1.3a. itemak_kargatu(): void
 - 2.1.1.3a.1. item_izenak_eskatu(): JSON4
 - [Item bakoitzeko]
 - 2.1.1.3a.2. itema_eskatu(): JSON5
 - 2.1.1.3a.3. execSQL(SQL5): void
 - 2.1.1.3a.4. execSQL(SQL3): void
 - 2.1.2. execSQL(SQL1): SQLEmaitza
 - 2.1.2.1. new SQLEmaitza
 - 2.1.2.2. next()
 - [Item bakoitzeko]
 - 2.1.2.3. getIzena(): String
 - 2.1.2.4. getArgazkia(): image
 - 2.1.2.5. getId(): int
 - [Erabiltzaileak nahi badu]
 - 3a. Filtro edo bilaketa sakatzen du
 - 4a. itemdex_kargatu(JSON2): JSON1
 - (2.1. pausuak berriro exekutatzen dira)
 5. Erabiltzaileak item bat aukeratzen du.
 6. new Item(id)
 - 6.1. bistaratu_item(id): JSON3
 - 6.1.1. item(id): JSON3
 - 6.1.1.1. execSQL(SQL2): SQLEmaitza
 - 6.1.1.1.1. new SQLEmaitza
 - 6.1.1.2. next()
 - [Item bakoitzeko]
 - 6.1.1.3. getArgazkia(): image
 - 6.1.1.4. getIzena(): String
 - 6.1.1.5. getDeskripzioa(): String
 - 6.1.1.6. getId(): int
 - 6.1.1.7. getMotalzena(): String

JSON1: {[{izena: string, argazkia: string, ID: int}]}

JSON2: {izena: string, motak: string[], alfabetikokiAlderantziz: boolean}

JSON3: {izena: string, argazkia: image, deskripzioa: string, itemID: int, Motalzena: string}

JSON4: {[{name: string}]} (item izenak)

JSON5: {id: int, izena: string, deskripzioa: string, argazkia: string, mota: string}

SQL1: SELECT I.itemID as ID, I.izena, I.argazkia
FROM Item I
WHERE I=1
AND I.izena LIKE '%izena%'
AND I.Motalzena IN (...)
ORDER BY I.izena ASC/DESC

SQL2: SELECT itemID, izena, deskripzioa, argazkia, Motalzena
FROM Item
WHERE itemID = ?
LIMIT 1

SQL3: INSERT OR IGNORE INTO Item (itemID, izena, deskripzioa, argazkia,
Motalzena)
VALUES (?, ?, ?, ?, ?)

SQL4: SELECT * FROM Item LIMIT 1

SQL5: INSERT OR IGNORE INTO MotalItem (ItemMotalzena) VALUES (?)

5.2 Probak

5.2.1 Erabiltzaileen kudeaketa (Eneko Puente)

Proba Kodea	Deskribapena	Espero den emaitza	Lortutako Emaitza	Oharrak
5.2.1.1	Erabiltzaileak baliozko kredentialak sartzen ditu eta "HASI SAIOA" botoiari eman.	Menu nagusira sartzen da.	Menu nagusira sartzen da.	Proba ondo atera da.
5.2.1.2	Erabiltzaileak baliozko lzen bat ipini, baina pasahitza ez da egokia.	Ez du saioa hasiko.	Ez du saioa hasiko.	Proba ondo atera da.
5.2.1.3	Erabiltzaileak existitzen ez diren kredentialak erabiltzen ditu.	Ez du saioa hasiko.	Ez du saioa hasiko.	Proba ondo atera da.
5.2.1.4	Erabiltzaileak hutsik utzi du eremu bat eta "HASI SAIOA" botoia sakatzen du.	Ez du saioa hasiko.	Ez du saioa hasiko.	Proba ondo atera da.
5.1.2.5	Erabiltzailea "ERREGISTRATU" botoiari ematen dio.	Erregistroko menura sartzen da.	Erregistroko menura sartzen da.	Proba ondo atera da.
5.2.1.6	Erabiltzaileak, karaktere espezialak dituen pasahitza ipintzen ditu eta "ERREGISTRATU" botoiari ematen dio.	"Pasahitza" eremuaren azpian errorea agertuko da, pasahitza karaktere bereziak dituela esanez.	"Pasahitza" eremuaren azpian errorea agertuko da, pasahitza karaktere bereziak dituela esanez.	Proba ondo atera da.
5.2.1.7	Erabiltzaileak, pasahitz ezberdinak ipintzen ditu eta "ERREGISTRATU" botoiari ematen dio.	Errore bat agertuko da, pasahitzak ezberdinak direla esanez.	Errore bat agertuko da, pasahitzak ezberdinak direla esanez.	Proba ondo atera da.
5.2.1.8	Erabiltzaileak balio ez duen posta	Errore bat agertuko da, posta baliozkoa	Errore bat agertuko da,	Proba ondo atera da.

	elektroniko bat ipintzen du eta “ERREGISTRATU” botoiari ematen dio.	ez dena esanez.	posta baliozkoa ez dena esanez.	
5.2.1.9	Erabiltzaileak balio ez duen jaiotze data bat ipintzen du eta “ERREGISTRATU” botoiari ematen dio.	“Jaiotze-data” eremuaren azpian, errore bat agertuko da, jaiotze-data baliozkoa ez dena esanez	“Jaiotze-data” eremuaren azpian, errore bat agertuko da, jaiotze-data baliozkoa ez dena esanez	Proba ondo atera da.
5.2.1.10	Erabiltzaileak, eremu bat hutsik uzten du eta “ERREGISTRATU” botoiari ematen dio.	Eremu horren azpian errore mezua agertuko da.	Eremu horren azpian errore mezua agertuko da.	Proba ondo atera da.
5.2.1.11	Erabiltzaileak eremu guztietan baliozko datuak sartzen ditu eta “ERREGISTRATU” botoiari ematen dio.	Erabiltzailea erregistratuko da eta saioa hasteko pantailara itzuliko da.	Erabiltzailea erregistratuko da eta saioa hasteko pantailara itzuliko da.	Proba ondo atera da.
5.2.1.12	Behin menu nagusian, Erabiltzaileak “LAGUNAK” botoia sakatzen du.	Lagunak kudeatzeko menura sartu.	Lagunak kudeatzeko menura sartu.	Proba ondo atera da.
5.2.1.13	Erabiltzaileak “GEHITU” botoia saketzen du.	Entrenatzaile bat ID-aren bidez gehitzeko pantaila agertuko da.	Entrenatzaile bat ID-aren bidez gehitzeko pantaila agertuko da.	Proba ondo atera da.
5.2.1.14	Erabiltzaileak izena eremua hutsik uzten du eta “BILATU” botoiari ematen dio.	izena berriro sartzeko eskatuko dio erabiltzaileari.	izena berriro sartzeko eskatuko dio erabiltzaileari.	Proba ondo atera da.
5.2.1.15	Erabiltzaileak existitzen ez den baliozko izen bat sartzen du eta “BILATU” botoiari ematen dio.	“Entrenatzaile hori ez da existitzen” mezua agertuko da.	“Entrenatzaile hori ez da existitzen” mezua agertuko da.	Proba ondo atera da.
5.2.1.16	Erabiltzaileak bere	Entrenatzaile hori	Entrenatzaile	Proba ondo

	lagun zerrenda jadanik dagoen entrenatzaile baten izena sartzen du eta “BILATU” botoiari ematen dio.	jadanik jarraitzen duzula agertuko da.	hori jadanik jarraitzen duzula agertuko da.	atera da.
5.2.1.17	Erabiltzaileak bere lagun zerrenda ez dagoen entrenatzaile baten ID-a sartzen du eta “BILATU” botoiari ematen dio.	Entrenatzaile hori bere lagun zerrendan gehituko da.	Entrenatzaile hori bere lagun zerrendan gehituko da.	Proba ondo atera da.
5.2.1.18	Erabiltzaileak “HOME” botoiari ematen dio.	Menu nagusira bueltatuko da.	Menu nagusira bueltatuko da.	Proba ondo atera da.
5.2.1.19	Erabiltzaileak “EZABATU” botoia sakatzen du.	Lagun bat ID-aren bidez ezabatzeko pantaila agertuko da.	Lagun bat ID-aren bidez ezabatzeko pantaila agertuko da.	Proba ondo atera da.
5.2.1.20	Erabiltzaileak bere lagun zerrendako isilarazita ez dagoen entrenatzaile baten notifikazioak “ISILARAZI” botoiari ematen dio.	Entrenatzailearen notifikazioak isilaraziko ditu.	Entrenatzailearen notifikazioak isilaraziko ditu.	Proba ondo atera da.
5.2.1.21	Erabiltzaileak bere lagun zerrendako isilarazita dagoen entrenatzaile baten notifikazioak “ISILARAZI” botoiari ematen dio.	Entrenatzailearen notifikazioak aktibatuko ditu.	Entrenatzailearen notifikazioak aktibatuko ditu.	Proba ondo atera da.
5.2.1.22	Erabiltzaileak “HOME” botoiari ematen dio.	Menu nagusira bueltatuko da	Menu nagusira bueltatuko da	Proba ondo atera da.
5.2.1.23	Behin menu nagusian, Administratzaile den erabiltzaile bat “KUDEATU” botoia sakatzen du.	Erabiltzaileak kudeatzeko menura sartuko da.	Erabiltzaileak kudeatzeko menura sartuko da.	Proba ondo atera da.

5.2.1.24	Administratzaileak kontu baten “EDITATU” botoia sakatzen du.	Erabiltzaile horren datuak editatzeko pantailara sartuko da.	Erabiltzaile horren datuak editatzeko pantailara sartuko da.	Proba ondo atera da.
5.2.1.25	Administratzaileak eremuren bat hutsik uzten du eta “GORDE” botoia sakatzen du.	Eremu hori sartzeko eskatuko dio administratzaileari	Eremu hori sartzeko eskatuko dio administratzaileari	Proba ondo atera da.
5.2.1.26	Administratzaileak “Pasahitza” eremuian karaktere bereziak ipintzen ditu eta “GORDE” botoia sakatzen du.	Pasahitza berriro sartzeko eskatuko dio.	Pasahitza berriro sartzeko eskatuko dio.	Proba ondo atera da.
5.2.1.27	Administratzaileak pasahitz ezberdinak ipintzen ditu eta “GORDE” botoia sakatzen du.	Pasahitzak ondo sartzeko eskatuko dio.	Pasahitzak ondo sartzeko eskatuko dio.	Proba ondo atera da.
5.2.1.28	Administratzaileak existitzen den izen bat sartzen du (uneko erabiltzailearena ez dena) eta “GORDE” botoia sakatzen du.	“izena erabilgarri dago jada” mezua agertuko da.	“izena erabilgarri dago jada” mezua agertuko da.	Proba ondo atera da.
5.2.1.29	Administratzaileak eremu guztiak betetezen ditu, posta elektroniko eta pasahitz egokiak sartzen ditu eta “GORDE” botoia sakatzen du.	Erabiltzailearen datuak eguneratuko dira.	Erabiltzailearen datuak eguneratuko dira.	Proba ondo atera da.
5.2.1.30	Erabiltzaileak “HOME” botoiari ematen dio.	Menu nagusira bueltatuko da.	Menu nagusira bueltatuko da.	Proba ondo atera da.
5.2.1.31	Administratzaileak, kontua ezabatzeko “EZABATU” botoia sakatzen du.	Kontua ezabatuko da.	Kontua ezabatuko da.	Proba ondo atera da.

5.2.1.32	Administratzailak “BILATZAILE” eremua hutsik uzten du eta “BILATU” botoia sakatzen du.	Izen bat berriro sartzeko eskatuko dio.	Izen bat berriro sartzeko eskatuko dio.	Proba ondo atera da.
5.2.1.33	Administratzailak “BILATZAILE” eremuan existitzen ez den entrenaztaile baten izena ipintzen du eta “BILATU” botoia sakatzen du.	Erabiltzaileen zerrenda hutsik agertuko da.	Erabiltzaileen zerrenda hutsik agertuko da.	Proba ondo atera da.
5.2.1.34	Administratzailak “BILATZAILE” eremuan existitzen den entrenaztaile baten izena ipintzen du eta “BILATU” botoia sakatzen du.	Erabiltzaileen zerrendan, izen hori duten trenatzailak agertuko dira.	Erabiltzaileen zerrendan, izen hori duten trenatzailak agertuko dira.	Proba ondo atera da.
5.2.1.35	Administratzailak erabiltzaile bat administratzaille bihurtzeko botoia sakatzen du.	Erabiltzaile horrek administratzaille bihurtzen da.	Erabiltzaile horrek administratzaille bihurtzen da.	Proba ondo atera da.
5.2.1.36	Erabiltzaileak “HOME” botoiari ematen dio.	Menu nagusira bueltatuko da.	Menu nagusira bueltatuko da.	Proba ondo atera da.
5.2.1.37	Erabiltzaileak “EDITATU” botoia sakatzen du.	Profilak eguneratzeko pantaila aterako da.	Profilak eguneratzeko pantaila aterako da.	Proba ondo atera da.
5.2.1.38	Erabiltzaileak eremuren bat hutsik uzten du eta “GORDE” botoia sakatzen du.	Eremu hori sartzeko eskatuko dio erabiltzaileari.	Eremu hori sartzeko eskatuko dio erabiltzaileari.	Proba ondo atera da.
5.2.1.39	Erabiltzaileak “Pasahitzza” eremuan karaktere bereziak ipintzen ditu eta “GORDE” botoia sakatzen du.	Pasahitzza berriro sartzeko eskatuko dio.	Pasahitzza berriro sartzeko eskatuko dio.	Proba ondo atera da.
5.2.1.40	Erabiltzaileak	Pasahitzak ondo	Pasahitzak	Proba ondo

	pasahitz ezberdinak ipintzen ditu eta “GORDE” botoia sakatzen du.	sartzeko eskatuko dio.	ondo sartzeko eskatuko dio.	atera da.
5.2.1.41	Erabiltzaileak existitzen den izen bat sartzen du (uneko erabiltzailearena ez dena) eta “GORDE” botoia sakatzen du.	“izena erabilgarri dago jada” mezua agertuko da.	“izena erabilgarri dago jada” mezua agertuko da.	Proba ondo atera da.
5.2.1.42	Erabiltzaileak eremu guztiak betetzeen ditu, posta elektroniko eta pasahitz egokik sartzen ditu eta “GORDE” botoia sakatzen du.	Erabiltzailearen datuak eguneratuko dira.	Erabiltzailearen datuak eguneratuko dira.	Proba ondo atera da.
5.2.1.43	Erabiltzaileak “HOME” botoiari ematen dio.	Menu nagusira bueltatuko da.	Menu nagusira bueltatuko da.	Proba ondo atera da.

5.2.2 Pokemon taldeak sortu (Asier Barrio)

Proba Kodea	Deskribapena	Espero den emaitza	Lortutako emaitza	Oharrak
3.2.1	Erabiltzailea “ EDIT ” botoia sakatzen du.	Hautatutako taldearen partaideak aldatzeko (gehitu edo ezabatu) menu berri bat agertuko zaio. (3.2.4-3.2.8)	Talde menua zabaldu da	Proba ondo atera da.
3.2.2	Erabiltzailea “+” botoia sakatzen du eta 10 talde baino gutxiago ditu sortuta.	EDIT botoia sakatzerakoan menu berdina agertuko da baina kasu honetan taldea hutsik agertuko da. (talde berri bat sortzen ari zarelako) (3.2.4-3.2.8)	Talde menua agertu da hutsik.	Proba ondo atera da.
3.2.3	Erabiltzailea “+” botoia sakatzen du eta 10 talde ditu sortuta.	Errore mezua agerutko da erabiltzaile bakoitzak gehienez 10 talde izan ahal dituelako sortuta aldi berean.	Errore mezua agertzen da.	Proba ondo atera da.
3.2.4	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) “ PARTAIDE BAT ”-en (pokemon baten) argazkia sakatzen du.	Hautatutako pokemonarekin zer egin nahi duen erabiltzailea beste menu bat agertuko da. (3.2.9-3.2.10)	Pokemon-a ezabatzeko edo aldatzeko menua zabaldu da.	Proba ondo atera da.
3.2.5	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) “+” botoia sakatzen du eta 6 pokemon baino gutxiago daude taldean.	Pokedex-aren menua agertuko da erabiltzaileak taldean sartu nahi duen pokemon-a hautatzeko (3.2.11-3.2.15)	Pokedex-a zabaltzen da	Proba ondo atera da.
3.2.6	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) “+” botoia	Errore mezua agertuko da talde pokemon bakoitza gehienez 6 pokemon	--	Zuzenean ez da “+” botoia agertuko 6 pokemon

	sakatzen du eta 6 pokemon daude taldean.	izan ahal dituelako.		daudenean.
3.2.7	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) " EZABATU " botoia sakatzen du.	Aukeratutako taldea sistematik ezabatuko da eta taldeen menu zerrendara bueltatuko da. (3.2.1-3.2.3)	Taldea sistematik ezabatzen da.	Proba ondo atera da.
3.2.8	Erabiltzaileak behin talde bat hautatuta (edo sortzen ari bada) " GORDE " botoia sakatzen du.	Aukeratutako taldea gordeko da aldaketa berrikin (ala ez) eta taldeen menu zerrendara bueltatuko da. (3.2.1-3.2.3)	Menu zerrendara bueltatzen da, taldea gordeta.	Proba ondo atera da.
3.2.9	Erabiltzaile batek pokemon bat aukeratzerakoan " EZABATU " botoia sakatzen du.	Hautatutako pokemon-a taldetik ezabatuko da eta taldeko menura bueltatuko da. (3.2.5-3.2.12)	Taldeko menuan Pokemonen ez dago	Proba ondo atera da.
3.2.10	Erabiltzaile batek pokemon bat aukeratzerakoan " ALDATU " botoia sakatzen du.	Pokedex-aren menua agertuko da pokemon hori ordezko bat bilatzeko. (3.2.11-3.2.15)	Pokedex menua zabaltzen da.	Proba ondo atera da.
3.2.11	Pokedex menuaren barruan erabiltzailea " POKEMON BAT "-en argazkia sakatzen du.	Hautatutako pokemonaren laburpen txiki bat eta aukeratzeko aukera agertuko da beste menu batean. (3.2.19-3.2.20)	Pokemon-aren laburpen txikia erakusten da, beste estatistika batzuekin	Proba ondo atera da.
3.2.12	Pokedex menuaren barruan erabiltzailea bilatzailean pokemon baten izena idazten du eta " ENTER " botoia sakaten du.	Pokedex-aren zerrenda filtratuko da pokemon-aren izenaren arabera.	Bilatutako Pokemon-a agertzen da	Proba ondo atera da.
3.2.13	Pokedex menuaren	Pokedex-aren	Bilatzailean	Proba ondo

	barruan erabiltzailea bilatzailean pokemon baten izena idazten du (baina ez osorik) eta “ ENTER ” botoia sakatzen du.	zerrenda filtratuko da dagoeneko idatziakoaren arabera, kointzidentziarik ez badaude zerrenda hutsik agertuko da	bilatutakoaren arabera filtratu da zerrenda	atera da.
3.2.14	Pokedex menuaren barruan erabiltzailea bilatzailean ez du ezer idatzten eta “ ENTER ” botoia sakatzen du.	Pokedex-aren zerrenda dagoen bezala geratuko da ez dagoelako ezer ez idatzita	Pokedex guztia agertzen da.	Proba ondo atera da.
3.2.15	Pokedex menuaren barruan erabiltzailea “ IRAGAZKETA ” botoia sakatzen du.	Pokedex-ean agertzen diren pokemon zerrenda filtratzeko aukerak agertuko dira. (3.2.16-3.2.18)	Iragazketa aukerak agertzen dira	Proba ondo atera da.
3.2.16	Iragazketa menuaren barruan erabiltzailea “ MOTAREN ARABERA ” botoia sakatzen du eta mota bat aukeratzen badu (kontuan izanda ez duela generaziorik aukeratu).	Pokedex-aren barruko pokemon zerrenda soilik motaren arabera filtratuko da.	Pokemon-ak motaren arabera filtratu daude	Proba ondo atera da.
3.2.17	Iragazketa menuaren barruan erabiltzailea “ GENERAZIOAREN ARABERA ” botoia sakatzen du eta generazio bat aukeratzen du (kontuan izanda ez duela motarik aukeratu).	Pokedex-aren barruko pokemon zerrenda soilik generazioaren arabera ordenatuko dira.	Pokemon-ak generazio arabera filtratu daude	Proba ondo atera da.
3.2.18	Iragazketa menuaren arabeta erabiltzaileak “ GENERAZIO BAT ETA MOTA BAT ”	Pokedex-aren zerrenda generazio horretako moten arabera filtratuko da. (Ez badaude	Pokemon-ak generazio eta motaren arabera filtratu daude	Proba ondo atera da.

	aukeratzen du.	kointzidentziarik zerrenda hutsik agertuko da)		
3.2.19	Pokemon informazioaren barruan erabiltzailea “AUKERATU” botoia sakatzen du eta hautatutako pokemon-a ez dago taldean.	Hautatutako pokemon-a taldera sartuko da (ordezkapena edo berria) eta taldeko menura bueltatuko da (3.2.4-3.2.8)	Pokemon-a taldera gehitu da.	Proba ondo atera da.
3.2.20	Pokemon informazioaren barruan erabiltzailea “AUKERATU” botoia sakatzen du eta hautatutako pokemon-a taldean dago.	Errore mezua agertuko da erabiltzailea aukeratu nahi duen pokemon-a taldean dagoela jada esanez.	Ez da errore mezurik agertu.	Azkenean erabaki da Pokemon espezie berdina errepikatu ahal dela talde berdinean.
3.2.21	Pokemon informazioaren barruan erabiltzailea “AUKERATU” botoia sakatzen du eta hautatutako pokemon-a taldean dago.	Pokemon-a taldera gehituko da.	Pokemon-a taldera gehitu da	Proba ondo atera da.
3.2.22	Erabiltzailea “HOME” botoia sakatzen du.	Erabiltzailea menu nagusira bueltatuko da. (3.1 puntu)	Menu nagusia agertzen da	Proba ondo atera da.

5.2.3 Changelog-a (Unai Rodriguez)

Proba Kodea	Deskribapena	Espero den emaitza	Lortutako emaitza	Oharrak
3.3.1	Erabiltzaileak menuko “JAKINARAZPENAK” botoia sakatzen du.	Jakinazpenak pantailara aldatzea eta bertan dauden jakinazpenak ikusi ahal izatea.	Espero den emaitzaren berdina	Proba ondo atera da, beraz funtzionalitatea ondo doa
3.3.2	Erabiltzaileak bilatzailean bilatu nahi duen erabiltzailearen izena erabiliz bilatzea. (Existitzen den izena)	Izen horri dagokion erabiltzailearen jakinazpenak pantailartuko dira.	Espero den emaitzaren berdina	Proba ondo atera da, beraz funtzionalitatea ondo doa. Gainera, notifikazioak dataren arabera agertuko dira ordenatuta baita ere
3.3.3	Erabiltzaileak bilatzailean bilatu nahi duen erabiltzailearen izena erabiliz bilatzea. (Existitzen den izena, baina erabiltzaile bat baino gehiago agertzea)	Erabiltzaile guzti hauen jakinarazpenak agertzea dataren arabera ordenatuta.	Espero den emaitzaren berdina	Proba ondo atera da, beraz funtzionalitatea ondo doa
3.3.4	Erabiltzaileak bilatzailean bilatu nahi duen erabiltzailearen izena erabiliz bilatzea. (Existitzen ez den izena)	ID-a berriz sartzeko eskatuko dio.	Ez da ezer agertuko zerrendan	NAME ID interfazea kendu dugunez ez zeukan zentzurik mezurik pantailatzea. Gainera, izena esaktzen da, ez ID-a
3.3.5	Bilatzailean letra bat edo letra para bat sartzea, hau da, izen partziala eta bilatu botoiari eman. (P edo Pa, Patxi izena	Letra horrekin edo izenaren parte partzialarekin dauden erabiltzaileak agertu behar dira.	Espero den emaitzaren berdina	Proba ondo atera da, beraz funtzionalitatea ondo doa. Gainera, notifikazioak

	bilatzekoan)			dataren arabera agertuko dira ordenatuta baita ere
3.3.6	Erabiltzaileak bilatzailea hutsik uzten du eta “ Bilatu ” botoia sakatzen du.	ID-a berriz sartzeko eskatuko dio.	Ez da ezer agertuko zerrendan	NAME ID interfazea kendu dugunez ez zeukan zentzurik mezurik pantailatzea. Gainera, izena esaktzen da, ez ID-a
3.3.7	Erabiltzaileak “ Menua ” botoia sakatzen du. (Jakinarazpenak barruan)	Menu nagusira bideratuko dio. (3.3.1)	Espero den emaitzaren berdina	Proba ondo atera da, beraz funtzionalitatea ondo doa

5.2.4 Pokedex-a (Yassin Mehani)

Proba Kodea	Deskribapena	Espero den emaitza	Lortutako emaitza	Oharrak
5.2.4.1	Erabiltzaileak Pokemon baten argazkian sakatzen du.	Hautatutako Pokemon-aren gaitasunak erakutsiko dira.	Hautatutako Pokemon-aren gaitasunak erakusten dira.	Zuzena
5.2.4.2	Erabiltzaileak bilatzailean Pokemon baten izena sartzen du eta " Enter " sakatzen du (Pokemon-a existitzen da).	Idatzitako Pokemon-aren argazkia agertzen da zerrendan.	Idatzitako Pokemon-aren argazkia agertzen da zerrendan.	Zuzena
5.2.4.3	Erabiltzaileak bilatzailean Pokemon baten izena sartzen du eta " Enter " sakatzen du (Pokemon-a ez da existitzen).	Errore mezua agertzen da, Pokemon-a existitzen ez dela adieraziz.	Errore mezua agertzen da, Pokemon-a existitzen ez dela adieraziz.	Zuzena
5.2.4.4	Erabiltzaileak bilatzailea hutsik uzten du eta " Enter " sakatzen du (Ez du bilatzailearen gainean klik egiten, edo klik egiten du baina ez du ezer idatzi).	Pokemon guztien zerrenda agertzen da.	Pokemon guztien zerrenda agertzen da.	Zuzena
5.2.4.5	Erabiltzaileak bilatzailea hutsik uzten du eta " Enter " sakatzen du (Bilatzailean klik egin du eta hutsune bat jarri du).	Errore mezua agertuko da eta Pokemon lista osoa erakutsiko da.	Zerrenda hutsik agertzen da eta errore mezua agertzen da.	Ez dira pokemon-ak agertzen, programak, hutsune hori izen bat bezala hartzen du eta datu-basean bilatzen du.

5.2.4.6	Erabiltzaileak Pokemon baten hitz partziala sartzen du eta “Enter” sakatzen du (Char adibidez).	Idatzitako hitz partzialarekin bat datozten Pokemon-ak erakusten ditu. (Charmander, Charmeleon, Charizard, Charcadet... erakutsiko ditu)	Idatzitako hitz partzialarekin bat datozten Pokemon-ak erakusten ditu.	Zuzena
5.2.4.7	Erabiltzaileak letra xehez edo letra larriz izen osoa idazten du bilatzailean eta Enter sakatzen du (CHARMANDER edo charmander , Charmander idatzi beharrean).	Sartutako Pokemon-aren izena erakutsiko du, nahiz eta letra larri edo letra xehe idatzi, emaitz bera agertuko da.	Sartutako Pokemon-aren izena erakusten da, nahiz eta letra larri edo letra xehe idatzi, emaitz bera agertzen da.	Zuzena
5.2.4.8	Erabiltzaileak Iragazki botoian sakatzen du.	Pokemon-ak filtrazeko aukerak erakutsiko dira. (3.4.9-3.4.15)	Pokemon-ak filtrazeko aukerak erakusten dira.	Zuzena
5.2.4.9	Erabiltzaileak Motaren Arabera iragazkia aukeratzen du eta bertan mota bat aukeratzen du (Generazio iragazketa ez du erabili).	Aukeratutako mota duten Pokemon-ak agertuko dira zerrendan.	Aukeratutako mota duten Pokemon-ak agertzen dira zerrendan.	Mota bat baino gehiago aukeratzen bada (adibidez, “Bicho” eta “Lucha”), ez dira agertuko bi mota horiek dituzten Pokemon-ak baizik eta “Bicho” mota duten Pokemon guztiak eta “Lucha” mota duten Pokemon guztiak.
5.2.4.10	Erabiltzaileak Generazioaren Arabera iragazkia aukeratzen du eta	Aukeratutako generazioan parte diren Pokemon-ak agertuko dira	Aukeratutako generazioan parte diren Pokemon-ak	Zuzena

	bertan generazio bat aukeratzen du (Mota iragazketa ez du erabili).	zerrendan.	agertuko dira zerrendan.	
5.2.4.11	Erabiltzaileak Generazioaren Arabera eta Motaren Arabera iragazkian erabiltzen ditu, generazio konkretu bat eta mota konkretu bat aukeratzeko	Aukeratutako generazio parten diren eta mota hori duten Pokemon-ak agertuko dira zerrendan.	Aukeratutako generazio parten diren eta mota hori duten Pokemon-ak agertuko dira zerrendan.	Mota bat baino gehiago aukeratzen bada (adibidez, “Bicho” eta “Lucha”), ez dira agertuko bi mota horiek dituzten Pokemon-ak baizik eta “Bicho” mota duten Pokemon guztiak eta “Lucha” mota duten Pokemon guztiak.
5.2.4.12	Erabiltzaileak iragazkiren bat (edo biak) erabiltzen d(it)u eta Pokemon baten izena idazten du bilatzailean eta “Enter” sakatzen du (Pokemon-a existitzen da).	Pokemon-a aukeratutako iragazkia betetzen badu zerrendan agertuko da, bestela hutsik erakutsiko da zerrenda.	Iragazkiak betetzen dituzten Pokemonak, zerrendan agertzen dira, bestela hutsik erakutsiko da zerrenda.	Zuzena
5.2.4.13	Erabiltzaileak iragazkiren bat (edo biak) erabiltzen d(it)u eta Pokemon baten izena idazten du bilatzailean eta “Enter” sakatzen du (Pokemon-a ez da existitzen).	Errore mezua agertzen da, Pokemon-a existitzen ez dela adieraziz.	Errore mezua agertzen da, Pokemon-a existitzen ez dela adieraziz.	Zuzena
5.2.4.14	Erabiltzaileak iragazkiren bat (edo biak) erabiltzen d(it)u eta Pokemon baten hitz partziala idazten du	Idatzitako hitz partzialarekin bat datozten Pokemon-ak iragazkia betetzen badute erakusten	Idatzitako hitz partzialarekin (espazio barik) bat datozten Pokemon-ak iragazkia	Zuzena

	bilatzailean eta “ Enter ” sakatzen du.	ditu.	betetzen badute erakusten dira.	
5.2.4.15	Edozein momentutan erabiltzaileak “ Home ” botoian sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena

5.2.5 Chatbot-a (Xinyan Wang)

Proba Kodea	Deskribapena	Espero den emaitza	Lortutako emaitza	Oharrak
5.2.5.1	Erabiltzaileak " CHATBOT " botoia sakatzen du menuan.	ChatBot pantaila irekitzen da.	ChatBot pantaila irekitzen da.	Zuzena.
5.2.5.2	Erabiltzaileak " MUGUMENDUAK " botoia sakatzen du.	Pokemon guztien zerrenda agertuko da. (3.5.2.X)	Pokemon guztien zerrenda agertuko da.	Zuzena.
5.2.5.3	Erabiltzaileak " ONENAK " botoia sakatzen du eta talde bat gutxienez lehendik sortuta dauka.	Pokemon taldeen zerrenda agertuko da. (3.5.3.X)	Pokemon taldeen zerrenda agertuko da.	Zuzena.
5.2.5.4	Erabiltzaileak " INDARRAK " botoia sakatzen du.	Pokemon guztien zerrenda agertuko da. (3.5.4.X)	Pokemon guztien zerrenda agertuko da.	Zuzena.
5.2.5.5	Erabiltzaileak " EBOLUZIOA " botoia sakatzen du.	Pokemon guztien zerrenda agertuko da. (3.5.5.X)	Pokemon guztien zerrenda agertuko da.	Zuzena.
5.2.5.6	Erabiltzaileak edozein momentutan " Home " botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.2.5.2.1	Erabiltzaileak edozein momentutan " Home " botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.2.5.2.2	Erabiltzaileak Pokemon baten ikonoa sakatzen du.	Pokemon horrek ikasi dezakeen mugimendu posible guztiak, horren argazkia eta izena pantilaratuko dira.	Pokemon horrek ikasi dezakeen mugimendu posible guztiak, horren argazkia eta izena	Zuzena.

			pantilaratu dira.	
5.2.5.2.2. 1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.2.5.3.1	Erabiltzaileak “ ONENAK ” botoia sakatzen du eta ez dauka gutxinez talde bat lehendik sortuta.	“Lehenik, taldea sortu” mezua agertuko da bistan.	“Lehenik, taldea sortu” mezua agertuko da bistan.	Zuzena.
5.2.5.3.2	Erabiltzaileak “ ONENAK ” botoia sakatzen du eta taldeak sortuta.	Taldeen zerrenda agertuko da.	Taldeen zerrenda agertuko da.	Zuzena.
5.2.5.3.2. 1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.2.5.3.3	Erabiltzaileak talde bat sakatzen badu eta talde honek ez du pokemonik.	“Talde honek ez du pokemonik” mezua agertuko da bistan.	“Talde honek ez du pokemonik” mezua agertuko da bistan.	Zuzena.
5.2.5.3.3. 1	Erabiltzaileak edozein momentutan “ Home ” botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.2.5.3.4	Erabiltzaileak talde bat sakatzen badu eta talde honek badu pokemonak.	Taldeko Pokemon hoberenaren izena, gaitasunak, estatistikak eta argazkia pantilaratzen dira.	Taldeko Pokemon hoberenaren izena, gaitasunak, estatistikak eta argazkia pantilaratu dira.	Zuzena.
5.2.5.4.1	Erabiltzaileak edozein	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.

	momentutan “Home” botoia sakatzen du.			
5.2.5.4.2	Erabiltzaileak Pokemon baten ikonoa sakatzen du. (Mota bakarreko)	Pokemon horrek noren aurka bere elementua den ahula eta noren aurka indartsua, horren argazkia eta izena pantailaratuko dira.	Pokemon horrek noren aurka bere elementua den ahula eta noren aurka indartsua, horren argazkia eta izena pantailaratuko dira.	Zuzena.
5.2.5.4.2.1	Erabiltzaileak edozein momentutan “Home” botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.2.5.4.3	Erabiltzaileak Pokemon baten ikonoa sakatzen du. (Bi motakoa)	Pokemon horrek noren aurka bere elementua den ahula eta noren aurka indartsua, horren argazkia eta izena pantailaratuko dira.	Pokemon horrek noren aurka bere elementua den ahula eta noren aurka indartsua, horren argazkia eta izena pantailaratuko dira.	Zuzena.
5.2.5.5.1	Erabiltzaileak edozein momentutan “Home” botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.2.5.5.2	Erabiltzaileak Pokemon baten ikonoa sakatzen du (Eboluzio katea duena).	Pokemon horrek daukan eboluzio katea agertuko da, Pokemon-en izena eta irudiarekin batera.	Pokemon horrek daukan eboluzio katea agertuko da, Pokemon-en izena eta irudiarekin batera.	Zuzena.
5.2.5.5.2.1	Erabiltzaileak edozein momentutan “Home” botoia	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.

	sakatzen du.			
5.2.5.5.3	Erabiltzaileak Pokemon baten ikonoa sakatzen du (Eboluzio katea ez duena).	“Eboluziorik ez” mezua pantailatzen da.	“Eboluziorik ez” mezua pantailatzen da.	Zuzena.

5.2.6 Itemdex (Yoel Justel)

Proba Kodea	Deskribapena	Espero den emaitza	Lortutako emaitza	Oharrak
5.6.1	Erabiltzaileak " ITEMDEX " botoia sakatzen du menuan.	ItemDex pantaila irekitzen da. (3.6.2-3.6.18)	ItemDex pantaila irekitzen da.	Zuzena.
5.6.1.1	Erabiltzaileak edozein momentutan " Home " botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.6.2	Erabiltzaileak " Iragazketa " botoia sakatzen du.	Iragazketa aukerak agertzen dira. (Motaren arabera eta alfabetikoki).	Iragazketa aukerak agertzen dira. (Motaren arabera eta alfabetikoki).	Zuzena.
5.6.3	Erabiltzaileak edozein momentutan " Home " botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.
5.6.4	Erabiltzaileak item baten izena idazten du eta " Enter " sakatzen du (itema ez da existitzen).	"Itema ez da existitzen" errore mezua agertzen da.	"Ez da itemik agertu" errore mezua agertzen da.	Zuzena.

5.6.5	Erabiltzaileak hutsik utzi du bilaketa eremua eta "Enter" sakatzen du.	Zerrenda osoa erakusten du.	Zerrenda osoa erakusten du.	Zuzena.
5.6.6	Erabiltzaileak zenbakiak sartzen ditu bilaketan.	<p>Item batzuek izen baten barruan zenbaki bat izango dute. Item horientzat, zenbaki hori duten itemen zerrenda erakutsiko da.</p> <p>Baina zenbaki luze bat bada, "Itema ez da existitzen" errore mezua agertzen da.</p>	<p>"0" bilatzean, 0 duten item guztiak agertu dira.</p>	Zuzena.
5.6.7	Erabiltzaileak karaktere bereziak sartzen ditu bilaketan (adibidez: @, #, \$).	<p>Karaktere berezi batzuentzat, "Itema ez da existitzen" errore mezua agertzen da.</p> <p>Baina, adibidez "-", ager daiteke, izan ere, badira izenean karaktere hori duten itemak.</p>	<p>"-" bilatzean (emaitza askoren artean) Ash-Pikastal Z agertu da.</p>	Zuzena.
5.6.8	Erabiltzaileak hitz partziala sartzen du bilaketan (adibidez: "ball" "Super Ball" bilatzeko).	Hitz partziala horrekin bat datorren item guztiak erakusten dira.	<p>"Ball" bilatzean (emaitza askoren artean) Super Ball agertu da.</p>	Zuzena.

5.6.9a	Erabiltzaileak maiuskulak /minuskulak erabiltzen ditu bilaketan (adibidez: "PASTA" vs "pasta").	Bilaketa maiuskula/ minuskula sentikorra ez dela egiaztatzen (emaitza bera itzuli behar da).	Emaitza bera itzuli du, azentorik ez badauzka hizkiak.	Gazteleraz dagoenez, azentua duten hizkiekin sentikorra da.
5.6.9b			"eter" edo "ETER" bilatzean agertzen da "Éter" itema.	Zuzena.
5.6.10a	Erabiltzaileak espazio gehiago sartzen ditu hasieran/amaieran bilaketan (adibidez: " pasta ").	Sistemak espazioak ezabatu behar ditu eta emaitza egokia erakutsi.	Ez da itemik agertu.	strip() metodoa erabiliz espazioak kendu dira.
5.6.10b			Itema agertu da.	Zuzena.
5.6.11	Erabiltzaileak luzeegia den bilaketa bat sartzen du (100 karaktere baino gehiago).	Sistemak ez du ezer agertuko.	"Ez da itemik agertu" errore mezua agertzen da.	Zuzena.
5.6.12	Erabiltzaileak iragazkiak aplikatzen ditu. "Motaren arabera" klikatzen du, eta motaren bat aukeratzen du. Adibidez, <i>abono</i> edo <i>caja de bayas</i> aukeratuz.	Iragazketa horri dagokion elementu-zerrenda agertuko da.	Iragazketa horri dagokion elementu-zerrenda agertu da.	Zuzena.
5.6.13	Erabiltzaileak iragazkiak kentzen ditu.	Iragazketa kentzen da eta zerrenda osoa berriro erakusten da.	Iragazketa kentzen da eta zerrenda osoa berriro erakusten da.	Zuzena.

5.6.14	Erabiltzaileak ordenaketa aldaketa egiten du (alfabetikoki).	Uneko zerrenda ordenatuta agertzen da (A-tik Z-ra).	Uneko zerrenda ordenatuta agertzen da (A-tik Z-ra).	Zuzena.
5.6.15	Erabiltzaileak mota bat aukeratzen du eta, gainera, item baten izena idazten du bilatzailean.	Mota horretan, izena betetzen duen item zerrenda agertuko da.	“Abono” iragazketa izatean eta “Br” idaztean bilatzailean bakarrik “Abono Brote” agertzen da.	Zuzena.
5.6.16	Erabiltzaileak item bat aukeratu eta gero “Itzuli” sakatzen du.	ItemDex pantailara itzultzen da.	ItemDex pantailara itzultzen da.	Zuzena.
5.6.17	Erabiltzaileak item bat bilatu eta gero beste bat bilatzen du.	Bigarren bilaketa egokia egiten da, lehenengoaren emaitzak ezabaturik.	Lehenengoz, “baya” bilatu da, ondoren, “ball” bilatu da, eta emaitzak ez dute zer ikusirik.	Zuzena.
5.6.18	Erabiltzaileak item baten gainean klik egiten du xehetasun pantailan.	Itemaren deskribapena, ikona eta erabilpena erakusten dituen leihoa berria irekitzen da.	Itemaren deskribapena, ikona eta erabilpena erakusten dituen leihoa berria irekitzen da.	Zuzena.
5.6.18.1	Erabiltzaileak edozein momentutan “Home” botoia sakatzen du.	Menu nagusira itzultzen da.	Menu nagusira itzultzen da.	Zuzena.

6. Implementazioan aurkitutako arazoak eta konponbideak

Atal honetan, proiektuaren implementazioan zehar aurkitu ditugun arazo nagusiak azalduko ditugu, baita arazo horiei aurre egiteko hartu ditugun konponbideak ere.

Lehenik eta behin, datuak API batetik lortu ditugu, baina zenbait kasutan APIak ez zuen behar adina informazio eskaintzen. Adibidez, Pokemonen moten kasuan, izena bakarrik jasotzen zen, baina ez zegoen dagokion irudirik. Arazo hori konpontzeko, irudiak eskuz bilatu genituen Interneten eta gure datu-basean gorde genituen. Horrez gain, item berezi batzuek ere ez zuten irudirik; kasu horietan, galdera-ikur baten irudia erabiltzea erabaki genuen.

Bestalde, hizkuntzarekin lotutako arazoak ere izan genituen. Datu gehienek gaztelaniazko itzulpena bazuten ere, zenbait elementuk ez zuten. Adibidez, item moten izenak ingelesetan bakarrik zeuden, eta ez zegoen gaztelaniazko bertsiorik. Hori dela eta, modu eskuzkoan itzulpen propioak sortu genituen item mota horiek gaztelaniara egokitzeko.

Azkenik, konexio-arazoak ere agertu ziren. Hasiera batean, pokebase liburutegia erabili genuen APIarekin konektatzeko, baina zenbait unetan konexioa huts egiten zuen, eta horrek datuak lortzea eragozten zuen. Arazo hori saihesteko, requests erabiltzera pasa ginen, URL bidez zuzenean APIra konektatuz. Soluzio honekin konexio-arazoak konpondu ziren, eta gainera, datuak eskuratzeko abiadura azkarragoa lortu genuen.

7. Ondorioak

Lan honi esker ikasi dugu nola den proiektu handi baten diseinua oinarritik egitea eta nola dokumentatu dena. Proiektuaren hasiera erabilpen kasuei buruzkoa izan zen, hori oso ondo eterri zitzaigun erabiltzaileak gure aplikazioa erabiltzean izango zituen aukera guztiak pentsatzeko eta ordena batekin programatu ahal izateko.

Gero, domeinu-ereduaren zatia ukitu genuen, horrek ere asko balio izan zigun jakiteko zer erakunde behar izan genituen proiektua aurrera eramateko. Gainera, diagrama horrek asko laguntzen zigun nola erlazionatu behar genituen entitateak, eta zer atributu estra erabili behar genituen erlazio bakoitzari buruzko informazio gehiago gordetzeko funtzionalitateak programatzerako orduan lana errazteko.

Horren ondoren, softwarearen probak egin genuen. Bertan, erabiltzaileak pentsatuta geneukan funtzionalitate bakoitzaren gainean zituen aukera guztiak aztertu genituen. Akats posibleak, akatsak saihesteko kontuan hartu beharreko funtzi bereziak... Aplikazio guztia implementatu ondoren, dena probatu beharko dugu, baina hori aurrerago egingo dugu.

Diseinuaren zatiarekin amaitzeko, klase-diagrama bat egin behar izan genuen, proiektuko objektuen erabilerari buruzko informazio guztia biltzeko. Gero, datu-base bat diseinatu, informazioa gorde ahal izateko (jakinarazpenak, erabiltzaileak...). Gainera, datu-base bat izan beharko genuke, APItik deskargatu beharreko informazio guztia gordetzeko. Funtzio garrantzitsuenen komunikazio- eta sekuentzia-diagrama ere diseinatu behar izan genuen.

Proiektu osoarekin amaitzeko, Visual Studio Code bidez implementatu behar izan genuen, Flask-en proba automatizatuekin batera. Dokumentazioan, Flask-en proba automatizatuen emaitzez gain, horiek implementatzeko akatsei buruz idatzi dugu.

Laburbilduz, proiektu oso konpletu bat izan da, bertan ikasi dugu nola dokumentatu, diseinatu, erabili APIak python eta ingurune berri batzuk programatzeko, hala nola PyCharm edo (gure kasuan) Visual Studio Code. Gainera, nabarmendu behar da git gehiago erabili dugula, tresna oso erabilgarria etorkizunari begira gure lanak adarkatu eta elkartu ahal izateko.