

Informatika Ingeniaritzako Gradua

Konputagailuen Ingeniaritza

Gradu Amaierako Lana

NBAko sare-soziala

Egilea

Jon Guilló Rodríguez

2019

Informatika Ingeniaritzako Gradua

Konputagailuen Ingeniaritza

Gradu Amaierako Lana

NBAko sare-soziala

Egilea

Jon Guilló Rodríguez

Zuzendaria(k)

Juan Ignacio Iturrioz Sánchez

Laburpena

Dokumentu hau "NBA-ko sare sozial"izeneko Gradu Amaierako Lanaren memoriari dagokio, Jon Guilló Rodríguez-ek garatua Ingeniaritza Informatikako Graduako Software Ingeniaritzako espezialitatean.

Proiektu honen helburua, zerbitzari (negozio-logika) bakar batean gainean, bezero aplikazio desberdinak garatzeko aukera ikustea da. Modu horretan, zerbitzariarekin konexioak sortuz, Android mugikorretarako aplikazioak edo web-aplikazioak zerbitzari beraren gainean lan egiteko aukera emango digute, ekosistema bakar bat sortuz.

NBAren inguruko sare sozial honek lagunen arteko komunitateak sortzeko aukera eskainiko du, erabiltzaileek beraien arteko txapelketak egiteko posibilitatea izanez. Denboraldi guztian zehar, datuak eguneratzen joango dira: bai erabiltzaileen zein jokalarien puntuak, partiduen emaitzak, jokalarien estatistikak...

Hori guztia modu automatiko batean egiteko NBA-ak eskainitako API-a ulertzea eta erabiltzea beharrezkoa izango da.

Gaien aurkibidea

| | |
|---|------------|
| Laburpena | i |
| Gaien aurkibidea | iii |
| Irudien aurkibidea | ix |
| Taulen aurkibidea | xv |
| 1 Proiektuaren Helburuen Dokumentua | 1 |
| 1.1 Sarrera | 1 |
| 1.2 Helburuak | 2 |
| 1.3 Irismena | 3 |
| 1.3.1 Betekizunak | 3 |
| 1.3.2 Baztertutakoaren deskribapena | 5 |
| 1.3.3 Mugarriak | 6 |
| 1.3.4 Irismenaren kudeaketa | 6 |
| 1.4 Lanaren Deskonposaketa Egitura | 7 |
| 1.5 Atazak eta ezaugarriak | 9 |
| 1.6 Emangarri nagusiak | 10 |
| 1.7 Baliabideak | 10 |
| 1.8 Proiektua gauzatzeko plana | 11 |

| | | |
|----------|--|-----------|
| 1.8.1 | Lan-metodologia | 11 |
| 1.8.2 | Dedikazioaren estimazioa | 11 |
| 1.8.3 | Gantt diagrama | 14 |
| 1.8.4 | Lan-karga | 14 |
| 1.9 | Kalitate-plana | 15 |
| 1.9.1 | Kalitatearen adierazleak | 15 |
| 1.9.2 | Kalitatea ziurtatzea | 16 |
| 1.9.3 | Kalitatearen kontrola | 17 |
| 1.10 | Arriskuen analisisa | 17 |
| 1.11 | Interesatuak | 19 |
| 1.11.1 | Komunikazio-plana | 19 |
| 2 | Eskakizunen bilketa | 21 |
| 2.1 | Aurrekariak | 21 |
| 2.1.1 | Testuingurua | 21 |
| 2.1.2 | Antzeko aplikazioak | 22 |
| 2.2 | Erabilpen Kasuak | 25 |
| 2.2.1 | Logeatu gabeko erabiltzailea | 26 |
| 2.2.2 | Logeaturako erabiltzailea (baita komunitateko administratzailea) | 26 |
| 2.2.3 | Administratzaile orokorra | 28 |
| 2.3 | Domeinuaren Eredua | 29 |
| 2.4 | Gertaera Fluxuak | 32 |
| 2.4.1 | Logeatu gabeko erabiltzailea | 32 |
| 2.4.2 | Logeaturako erabiltzailea | 34 |
| 2.4.3 | Logeatu gabeko edo logeaturiko erabiltzailea | 43 |
| 2.4.4 | Komunitateko administratzailea | 44 |
| 2.4.5 | Administratzaile globala | 45 |

| | | |
|----------|--|-----------|
| 3 | Diseinua | 49 |
| 3.1 | Teknologia posibleak | 49 |
| 3.1.1 | Zerbitzaria | 49 |
| 3.1.2 | Datu-Basea | 50 |
| 3.1.3 | Zerbitzaria eta Datu-Basearen arteko konexioa | 50 |
| 3.1.4 | Web-Bezeroa eta Zerbitzariaren konexioa | 50 |
| 3.1.5 | Web-Bezeroaren Interfazeak | 51 |
| 3.1.6 | Android-Bezeroa, Zerbitzariaren Konexioa eta Interfazeak | 51 |
| 3.2 | Erabilitako tresna eta teknologiak | 52 |
| 3.2.1 | Eclipse eta Java | 52 |
| 3.2.2 | MySQL | 53 |
| 3.2.3 | Hibernate | 53 |
| 3.2.4 | PHP eta Nusoap | 53 |
| 3.2.5 | HTML5, JavaScript eta Bootstrap | 53 |
| 3.2.6 | Android Studio eta Ksoap | 53 |
| 3.3 | Datu-basearen diseinua | 54 |
| 3.4 | Logeatu gabeko erabiltzailea | 54 |
| 3.4.1 | Komunitatea sortu (eta administratzailea) | 55 |
| 3.4.2 | Kontua sortu | 56 |
| 3.4.3 | Kontura sartu | 58 |
| 3.5 | Logeatutako erabiltzailea | 60 |
| 3.5.1 | Erabiltzailearen jokalarien datuak ikusi | 60 |
| 3.5.2 | Alineazioa ikusi, editatu, gorde eta publikatu | 61 |
| 3.5.3 | Jokalari bat merkatuan jarri | 63 |
| 3.5.4 | Merkatua kontsultatu | 64 |
| 3.5.5 | Merkatuko jokalaria bati eskaintza bat egitea | 65 |

| | | |
|--------|---|----|
| 3.5.6 | Erabiltzaileak egindako eskaintzak kontsultatu | 66 |
| 3.5.7 | Erabiltzaileak jasotako eskaintzak kontsultatu, onartu edo ezeztatu | 67 |
| 3.5.8 | Jokalari baten puntuen eboluzioa | 69 |
| 3.5.9 | Jokalari guztien puntuak | 70 |
| 3.5.10 | Transakzioen historikoa | 71 |
| 3.5.11 | Mezuak irakurri | 73 |
| 3.5.12 | Mezuak idatzi | 74 |
| 3.5.13 | Jardunaldiko klasifikazioa | 75 |
| 3.5.14 | Klasifikazio orokorra | 76 |
| 3.6 | Logeatu gabeko edo logeaturiko erabiltzailea | 77 |
| 3.6.1 | Eguneko partiduen emaitzak | 77 |
| 3.6.2 | Jokalarien datuak kontsultatu | 78 |
| 3.6.3 | Erabiltzaileen alineazio publikoak kontsultatu | 79 |
| 3.7 | Komunitateko administratzailea | 81 |
| 3.7.1 | Sariak aldatu | 81 |
| 3.7.2 | Mezua ezabatu | 82 |
| 3.7.3 | Erabiltzailea ezabatu | 83 |
| 3.8 | Administratzaile globala | 85 |
| 3.8.1 | Jardunaldiak hasieratu | 86 |
| 3.8.2 | Soldatak eguneratu | 86 |
| 3.8.3 | Jokalari eta erabiltzaileei jardunaldiko puntuak esleitu | 87 |
| 3.8.4 | Alineazioak finkatu | 88 |
| 3.8.5 | NBA-ko jokalarien informazioa eguneratu gure datu-basean | 89 |
| 3.8.6 | NBA-ko taldeen informazioa eguneratu gure datu-basean | 90 |
| 3.8.7 | Komunitateei mezu orokorrak bidali | 91 |
| 3.8.8 | Komunitateen merkatuak eguneratu | 92 |

| | | |
|----------|--|-----------|
| 4 | Garapena | 93 |
| 4.1 | Zerbitzaria | 93 |
| 4.1.1 | MySQL, PhpMyAdmin, Apache eta Xampp | 93 |
| 4.1.2 | Hibernate | 95 |
| 4.1.3 | Java Endpoint | 96 |
| 4.1.4 | Datuak JSON formatutik irakurri | 98 |
| 4.1.5 | Web scraping | 98 |
| 4.2 | Web-Bezeroa | 99 |
| 4.2.1 | Apache | 99 |
| 4.2.2 | Nusoap | 99 |
| 4.2.3 | Bootstrap | 100 |
| 4.2.4 | Jquery, JavaScript, CSS eta HTML5 | 100 |
| 4.3 | Android-Bezeroa | 101 |
| 4.3.1 | Android-Studio | 101 |
| 4.3.2 | Picasso | 101 |
| 4.3.3 | Ksoap | 102 |
| 4.3.4 | Firebase | 104 |
| 4.3.5 | Locale | 105 |
| 4.3.6 | Interfazeen karga asinkronoa | 105 |
| 4.4 | Probak | 107 |
| 4.4.1 | Denboraldia hasi aurretik datuak eguneratzea | 108 |
| 4.4.2 | Komunitateak eta erabiltzaile kontuak sortu | 109 |
| 4.4.3 | Logeaturiko erabiltzaileen ekintzak | 109 |

| | | |
|---------------------|--|------------|
| 5 | Jarraipena eta Kontrola | 113 |
| 5.1 | Desbideraketak | 113 |
| 5.1.1 | Kudeaketa | 115 |
| 5.1.2 | Ikerkuntza | 115 |
| 5.1.3 | Analisi eta diseinua | 115 |
| 5.1.4 | Garapena | 115 |
| 5.1.5 | Egiaztapena | 116 |
| 5.1.6 | Dokumentazioa | 116 |
| 5.2 | Gantt diagrama | 116 |
| 5.3 | Mugarriak | 117 |
| 5.4 | Kalitate-plana | 118 |
| 5.5 | Arriskuak | 120 |
| 6 | Ondorioak | 121 |
| 6.1 | Ikasitako lezioak | 121 |
| 6.2 | Etorkizunerako aukerak | 122 |
| Eranskinak | | |
| A | Bilera-aktak | 127 |
| B | Erabilitako materiala | 133 |
| C | Erabilera gida | 135 |
| C.1 | Komunitate eta administratzailea sortu | 135 |
| C.2 | Erabiltzailea sortu | 138 |
| C.3 | Kontura sartu | 139 |
| C.4 | Alineazioa aldatu | 140 |
| Bibliografia | | 141 |

Irudien aurkibidea

| | | |
|------|--|----|
| 1.1 | LDE diagrama. | 8 |
| 1.2 | Dedikazio estimazioaren grafikoa. | 13 |
| 1.3 | Gantt diagrama. | 14 |
| 2.1 | Erabilpen Kasuak. | 25 |
| 2.2 | Logeatu gabeko erabiltzailearen erabilpen kasuak. | 26 |
| 2.3 | Logeatutako erabiltzailearen erabilpen kasuak. | 27 |
| 2.4 | Administratzaile orokorraren erabilpen kasuak. | 28 |
| 2.5 | Domeinuaren eredia. | 29 |
| 3.1 | Erabilitako teknologien arkitektura. | 52 |
| 3.2 | Datu-basearen diseinua. | 54 |
| 3.3 | Komunitatea sorturen interfazea. | 55 |
| 3.4 | Erabiltzailea sorturen interfazea. | 55 |
| 3.5 | Komunitatea sortu (eta administratzailea) sekuentzia-diagrama. | 56 |
| 3.6 | Komunitatea aukeratu interfazea. | 57 |
| 3.7 | Erabiltzailea sortu interfazea. | 57 |
| 3.8 | Kontua sortu sekuentzia-diagrama. | 58 |
| 3.9 | Mezuak irakurri interfazea. | 58 |
| 3.10 | Mezuak irakurri sekuentzia-diagrama. | 59 |

| | |
|--|----|
| 3.11 Erabiltzailearen jokalarien interfazea. | 60 |
| 3.12 Erabiltzailearen jokalarien sekuentzia-diagrama. | 61 |
| 3.13 Alineazioaren interfazea. | 61 |
| 3.14 Alineazioa ikusi, editatu, gorde eta publikaturen sekuentzia-diagrama. . . | 62 |
| 3.15 Jokalari bat merkatuan jarriren interfazea. | 63 |
| 3.16 Jokalari bat merkatuan jarriren sekuentzia-diagrama. | 64 |
| 3.17 Merkatua kontsultatu interfazea. | 64 |
| 3.18 Merkatua kontsultatu sekuentzia-diagrama. | 65 |
| 3.19 Merkatuko jokalari bati eskaintza egitearen interfazea. | 65 |
| 3.20 Merkatuko jokalari bati eskaintza egitearen sekuentzia-diagrama. | 66 |
| 3.21 Erabiltzaileak egindako eskaintzak kontsultaturen interfazea. | 66 |
| 3.22 Erabiltzaileak egindako eskaintzak kontsultaturen sekuentzia-diagrama. . | 67 |
| 3.23 Erabiltzaileak jasotako eskaintzak kontsultatu eta onarturen interfazea. . . | 67 |
| 3.24 Erabiltzaileak jasotako eskaintzak kontsultatu eta onarturen sekuentzia- diagrama. | 68 |
| 3.25 Jokalari baten puntuen eboluzioaren interfazea. | 69 |
| 3.26 Jokalari baten puntuen eboluzioaren sekuentzia-diagrama. | 70 |
| 3.27 Jokalari guztien puntuen interfazea. | 70 |
| 3.28 Jokalari guztien puntuen sekuentzia-diagrama. | 71 |
| 3.29 Transakzioen historikoen interfazea. | 72 |
| 3.30 Transakzioen historikoen sekuentzia-diagrama. | 72 |
| 3.31 Mezuak irakurriri dagokion interfazea. | 73 |
| 3.32 Mezuak irakurriri dagokion sekuentzia-diagrama. | 73 |
| 3.33 Mezuak idazteari dagokion interfazea. | 74 |
| 3.34 Mezuak idazteari dagokion sekuentzia-diagrama. | 74 |
| 3.35 Jardunaldiko klasifikazioaren interfazea. | 75 |

| | | |
|------|---|----|
| 3.36 | Jardunaldiko klasifikazioaren sekuentzia-diagrama. | 75 |
| 3.37 | Klasifikazio orokorraren interfazea. | 76 |
| 3.38 | Klasifikazio orokorraren sekuentzia-diagrama. | 76 |
| 3.39 | Partiduen emaitzen interfazea. | 77 |
| 3.40 | Jardunaldiko partiduen emaitzen sekuentzia-diagrama. | 78 |
| 3.41 | Jokalarien datuen interfazea. | 78 |
| 3.42 | Jokalarien datuak kontsultatu sekuentzia-diagrama. | 79 |
| 3.43 | Erabiltzaileen alineazio publikoen interfazea. | 79 |
| 3.44 | Erabiltzaileen alineazio publikoak kontsultaturen sekuentzia-diagrama. . . | 80 |
| 3.45 | Sariak aldatu dagokion interfazea. | 81 |
| 3.46 | Sariak aldatu dagokion sekuentzia-diagrama. | 82 |
| 3.47 | Mezua ezabatzeari dagokion interfazea. | 82 |
| 3.48 | Mezua ezabatzeari dagokion sekuentzia-diagrama. | 83 |
| 3.49 | Erabiltzailea ezabatzeari dagokion interfazea. | 84 |
| 3.50 | Erabiltzailea ezabatzeari dagokion sekuentzia-diagrama. | 84 |
| 3.51 | Administratzaile globalaren interfazea. | 85 |
| 3.52 | Jardunaldiak hasieraren sekuentzia-diagrama. | 86 |
| 3.53 | Jokalarien soldatak eguneratzearen sekuentzia-diagrama. | 87 |
| 3.54 | Jokalari eta erabiltzaileei jardunaldiko puntuak esleitu sekuentzia-diagrama. | 88 |
| 3.55 | Alineazioak finkaturen sekuentzia-diagrama. | 89 |
| 3.56 | NBA-ko jokalarien informazioa eguneratu gure datu-basean sekuentzia- diagrama. | 90 |
| 3.57 | NBA-ko taldeen informazioa eguneratu gure datu-basean sekuentzia-diagrama. | 90 |
| 3.58 | Komunitatei mezua orokorrak bidaliri dagokion sekuentzia-diagrama. . . | 91 |
| 3.59 | Komunitateen merkatuak eguneratzearen sekuentzia-diagrama. | 92 |
| 4.1 | Xampp-en kontrol panela. | 94 |

| | | |
|------|--|-----|
| 4.2 | phpMyAdmin-en interfazean gure datu-basea. | 94 |
| 4.3 | Erabiltzaileen kudeaketa phpMyAdmin. | 94 |
| 4.4 | Hibernate konexioa ezartzen. | 95 |
| 4.5 | Hibernate konexioa ezartzen. | 96 |
| 4.6 | Web-zerbitzua martxan jartzen. | 96 |
| 4.7 | Adibidea: Nusoap liburutegiaren erabilera e-maila libre dagoen kontsul- tatzeko. | 100 |
| 4.8 | Interfaze grafikoak Android Studion. | 101 |
| 4.9 | build.gradle app-ren zati bat. | 102 |
| 4.10 | Adibidea: Mezua klasean KvmSerializable-ren funtzioen implementazioa. | 103 |
| 4.11 | Adibidea: Erabiltzailearen jokalaria lortzeko deia. | 103 |
| 4.12 | Erabiltzaileen estatistikak. | 104 |
| 4.13 | Notifikazioak bidali. | 104 |
| 4.14 | Datu-base hutsik. | 107 |
| 4.15 | Datu-basea hasieratua. | 108 |
| 4.16 | Komunitateak eta erabiltzaileak datu-basean. | 109 |
| 4.17 | Erabiltzailearen jokalaria datu-basean. | 110 |
| 4.18 | Alineazioak datu-basean. | 110 |
| 4.19 | Erabiltzailea4-k ikusten dituen alineazio publikoak. | 110 |
| 4.20 | Eskaintzak datu-basean. | 111 |
| 4.21 | Transakzioak datu-basean. | 111 |
| 4.22 | Alineazioak datu-basean. | 112 |
| 4.23 | Erabiltzaileen puntuak datu-basean | 112 |
| 4.24 | Jokalarien puntuak datu-basean | 112 |
| 4.25 | Transakzioak datu-basean. | 112 |
| 5.1 | Gantt diagramaren egoera erreala. | 117 |

| | | |
|-----|--|-----|
| C.1 | Logeatu gabeko erabiltzailearen menua. | 135 |
| C.2 | Komunitate berria sortu. | 136 |
| C.3 | Komunitate berria sortu. | 136 |
| C.4 | Erabiltzaile berria sortu. | 137 |
| C.5 | Erabiltzaile berria sortu. | 137 |
| C.6 | Erabiltzaile berria sortu. | 138 |
| C.7 | Erabiltzaile berria sortu. | 138 |
| C.8 | Erabiltzaile kontura sartu. | 139 |

Taulen aurkibidea

| | | |
|-----|--|-----|
| 1.1 | LDE diagraman zehazturiko atazen azalpena | 9 |
| 1.2 | Ataza bakoitza burutzeko ordu estimazioa | 12 |
| 2.1 | Antzeko produktuen konparaketa | 24 |
| 4.1 | Komunitate eta kontuen datu basikoak. | 109 |
| 4.2 | Egin beharko diren eskaintzak. | 111 |
| 5.1 | Denbora estimazioaren eta errealaren arteko desbideraketa. | 114 |
| 5.2 | Mugarrien eguneraketa. | 117 |

1. KAPITULUA

Proiektuaren Helburuen Dokumentua

Memoriaren atal honetan, proiektua aurrera eramateko bete beharko diren atazak daude. Lehendabizi sarrera txiki bat dugu eta jarraipenean proiektuaren helburuak. Bestalde, irismenari buruz hitz egiten da, lanaren deskonposaketa nolako izango den, ataza bakoitzaren ezaugarriak aipatuz.

Horrez gain, emangarri nagusiak, erabiliko diren baliabideak, proiektua gauzatzeko plana, kalitate-plana, lan-metodologia eta arriskuen analisia egingo da.

Bukatzeko, interesatuen inguruko hausnarketa txiki bat burutuko da.

1.1 Sarrera

Proiektuaren helburua NBAren inguruko sare sozial bat sortzea izango da. Erabiltzaile bakoitzak kontu bat izango du, eta komunitate baten (eta bakarraren) barruan egongo da. Bertan beraren lagunekin txapelketa bat osatzeko aukera izango du.

Erabiltzaile bakoitzak NBA-ko jokalaria izango ditu. Horiekin bosteko bat aukeratu beharko du eta horren bidez puntuak irabazi.

Astero, puntuak esleituko zaizkie jokalariei eta ondorioz erabiltzaile bakoitzak jarritako jokalarien puntuak jasoko ditu. Aste horretan puntu gehien lortu dituzten erabiltzaileek (komunitate bakoitzean) sariak edukiko dituzte.

Sari hori diru kantitate bat izango da. Diru horren bitartez jokalaria erosteko aukera izango du erabiltzaile bakoitzak. Komunitate bakoitzak merkatu bat izango du, eta hau

periodikoki eguneratuko da. Merkatu horretan edozein erabiltzaileek jokalaria jartzeko aukera izango du, merkatuko jokalaria horiei eskaintzak egin eta jasotako eskaintzak kontsultatzeko.

Ondorioz, helburua denboraldi bukaeran zure komunitatean puntu gehien lortu dituen erabiltzailea izatea izango da.

1.2 Helburuak

Hauek dira proiektu honekin lortu nahi diren helburu nagusiak:

- NBaren inguruko sare sozial bat garatzea.
 - Erabiltzaileak eta komunitateak sortzea ahalbidetu.
 - Modu automatiko batean, NBAko datu-baseetako datuak irakurri, tratatu eta gure aplikaziorako erabilgarriak bihurtu.
 - Administratzaile orokorraren lana modu automatiko batean egiteko diseinatu.
 - Erabiltzaileari, entrenatzaile eta presidente lanak simulatze aldera aukera ugari eskaini.
 - Zerbitzari komun batetik, bezero mota desberdinak diseinatu: *Android* eta Web aplikazioa.
- Gradu Amaierako Lanaren garapenaren inguruko informazioa biltzen duen memoria garatzea.
- Egindako proiektuaren defentsarako aurkezpena.

Aldi berean, helburu nagusi horiek aurrera eramateko beharrezkoak izango dira ondorengo gaitasunak lortzea. Hauek dira bigarren mailako helburuak:

- Arazo erreal baten aurrean domeinu bat diseinatzeko trebezia.
- Erabiliko diren teknologiak aukeratu. Horretarako, hausnarketa sakon bat egin.
- *Android* mugikorren egitura eta teknologiak ezagutu.
- Web-zerbitzuak aztertu eta ulertu.
- Ikasketa auto-didaktikoa bultzatu.

1.3 Irismena

Helburuak ikusi ondoren, irismena zehaztea dago. Proiektuaren betekizunak, baztertutakoaren deskribapena, mugariak, produktuaren irismen-mailak eta kudeaketa aztertuko dugu.

1.3.1 Betekizunak

Proiektuaren helburua NBAren inguruko sare sozial bat sortzea izango da. Erabiltzaile bakoitza komunitate batean (bakarrean) egongo da. Komunitate bakoitzak gehienez 10 erabiltzaile eta beti administratzaile bat izango du.

Aplikazio barruan dagoen edozein erabiltzailek partiduen emaitzak ikusteko aukera izango du (eta partidua oraindik jokatzeaz baldin badago, partiduaren ordutegia). Horrez gain, NBA-ko jokalarien bilaketak egitea ahalbidetu beharko zaio (jokalariaren oinarrizko datuak).

Erabiltzaileek momentuan duten taldea publikoki erakusteko aukera izango dute, beraz edozein erabiltzailek jendeak proposaturiko taldeak ikusteko aukera izan behar du.

Azkenik, edozein erabiltzailek aplikazioa pertsonalizatzeko posibilitatea izango du. Bertan hizkuntza (euskara, gaztelera, ingelesa), ordua (espainiar ordutegia. . .), eta koloreak aldatzeko aukera izango du besteak beste.

Oraindik logeatua ez dagoen erabiltzaile batek bere kontura sartzeko, kontu berria sortzeko eta beraren liga (komunitate) propioa egiteko aukera izango du. Komunitate berri bat sortzeak, kontu berri bat sortzea ekarriko du eta kontu hori komunitate horretako administratzailea izango da.

Behin kontuan logeatua, ekintza berriak egiteko aukera eskainiko zaio.

Lehendabizi erabiltzaile arrunt baten paperean jarriko gara. Hainbat sekziotan banaturik agertuko dira eskuragarri dituen ekintzak.

- *Jokalariak*: Bertan erabiltzaile bakoitzaren jokalarien zerrenda azalduko da (izen, abizen, argazkia, soldata, posizioa eta NBAko frankiziaren izena eta logoa), momentuan erabiltzen ari den boskotea (alineazioa) eta hau aldatzeko aukera izango

du. Gainera, jokalaria zehatz baten informazioa kontsultatzeko posibilitatea (oinarriko datuez gain, estatistika aurreratuak).

- *Merkatua*: Komunitate bakoitzak merkatu propio bat izango du. Bertan jokalaria jartzeko aukera izateaz gain, merkatuan dauden jokalarien kontsulta edo jokalaria zehatz bateri eskaintza bat egiteko aukera izango du. Gainera, historiko bat gordeko da komunitate horretan egon diren fitxaketen ingurukoa.
- *Finantzak*: Finantzei jarraipen bat egite aldera, erabiltzaile horren transakzioen historiko bat gordeko da, eta momentuan duen diru kantitatea.
- *Estatistika*: Estatistikak jokalarietan oinarritzen dira. Aurreko jarduneko (edo aukeraturikoa) jokalarien puntuazioa edo jokalaria zehatz baten asteroko puntuak ikusteko aukera izango dugu. Gainera, beste erabiltzaileen jokalarien puntuazio orokorrak ikusteko aukera edukiko dugu.
- *Klasifikazioa*: Azken jardunaldiaren (edo aukeraturikoaren) klasifikazioa edo denboraldi guztiaren orokorra.
- *Komunitatea*: Hemen mezuak idazteko eta irakurtzeko aukera izango dugu. Gainera, puntuazioak gauzatzeko erregelak azalduko dira kontsultatzeko nahi baldin bada.

Bestalde, komunitate bakoitzeko administratzailea dugu. Honek, aurreko ekintza guztiak egiteko gaitasuna edukitzeaz gain, komunitatearen kontrol zehatza edukitzeko beste aukera batzuk ditu.

- *Finantzak*: Atal honetan jarduneko irabazleei emango zaien diru kantitatea zehazteko aukera izango du (hasieran ezarriko da, komunitatea sortzean). Aldaketarik egin nahi baldin bada, komunitate horretako gainontzeko erabiltze guztiek onartu beharko dute.
- *Komunitatea*: Publikaturik dagoen mezuren batek ez baldin bada egokia, administratzaileak hori ezabatzeko ahalmena izango du.
- *Erabiltzaileen kontrola*: Komunitatean dagoen erabiltzaile bat ezabatzea nahi baldin bada (aktiboa ez delako, gainontzeko erabiltzaileei errespetu falta konstanteak...), administratzaileak ezabatzeko posibilitatea izan beharko du.

Eta bukatzeko Administratzaile global bat izango dugu. Honek erabiliko duen aplikazio/web-orria independentea da, ezingo du aurrekoa atzitu baina aldi berean, aurrekotik ezinezkoa izango da hau atzitzea.

- NBAko jokalarien jarduneko puntuazioak eguneratzeko aukera edukiko du. Hau algoritmo batean oinarrituriko izango da, non administratzaileak zehaztu beharko duen gauza bakarra egun tartea izango da.
- Aldi berean, erabiltzaileei puntuazioak eguneratuko zaizkie. Hau, aurrekoarekin batera gertatuko da beti.
- NBA-ko jokalarien informazioa eguneratu gure datu basean. Honek kanpoko API-a erabiliz gure datu-baseko informazioa eguneratuko du. Jokalariak zein taldeak eguneratzeko balioko digu.
- Komunitateei (guztiei batera), mezuak bidaltzeko aukera izango dugu. Mezu hauek ezingo ditu administratzaileak ezabatu.
- Gainera, urtetik urtera administrazioa automatizatze aldera, asteen identifikazioa bakarrik sortuko da. Hau da, liga ofizialki hasiko den eguna jarritz (urtea, hilabete eta eguna), automatikoki asteak identifikazio baten bidez datu-basean gordeko dituen aukera eskaini.
- Jardunaldi bakoitzaren hasieran erabiltzaile bakoitzak aukeratu dituen jokalarien boskotea finkatzeko beste aukera bat beharko dugu.
- Bukatzeko, komunitate guztien merkatuak egunero eguneratuko dira. Hori lortzeko, komunitatean momentu horretan dauden merkatuko jokalaria guztiak tratatu eta berriak jarri beharko dira. Akzio hau automatizatua egotea guztiz garrantzitsua da.

1.3.2 Baztertutakoaren deskribapena

Gradu Amaierako Lan honek 12 kreditu exijitzen ditu, eta horrek 300 orduko lan muga ezartzen du. Hori dela eta, interesgarriak diren zenbait gauza baztertu behar dira.

Bezera mota desberdinak sortzea interesgarri bada ere, ezinezkotzat jo dugu *iPhone* mugikorretarako aplikazio propioa sortzea. Hirurak egitea denbora aldetik ezinezkoa zenez, garapen kostu handiena eta gainera merkatu txikiena duen aukera ezeztatutzat eman genuen hasieratik. Gauza bera gertatu da *WindowsPhone* telefono mugikorrekin.

Horrez gain, oso polita izango zen sekulako interfaze grafikoak diseinatzea, baina hori ez da gure lana ezta ere gure helburua, beraz denbora maximo bat ezarri beharko diogu prozesu horri.

1.3.3 Mugarriak

Horrelako proiektu handi eta konplexuetan oso garrantzitsua da atazak burutzeko epeak ezartzea. Kasu honetan, ataza multzo bakoitzari mugarri bat jartzea oso gomendagarria izaten da, gerta daitezkeen atzerapenak kontrolpean edukitzeko.

Hauek dira kontuan eduki beharreko mugarriak:

- **2018-12-14** Plangintza garatu.
- **2018-12-28** Beharrezkoa den informazioa biltzea (ikerkuntza lana).
- **2019-01-16** Eskakizunen bilketa.
- **2019-01-30** Diseinua.
- **2019-04-19** Proiektuaren inplementazioa.
- **2019-05-31** Memoria idatzi.
- **2019-06-07** Aurkezpena prestatu.

Mugarriak errespetatzea garrantzitsua den arren, orientagarriak izango dira. Proiektuaren garapenean zehar arazorik sortzen baldin bada, mugarri horiek moldatu beharko dira. Horregatik, garrantzitsua da gure hasierako planifikazioan ezarritako bukaera data eta entrega eguna berdinak ez izatea.

1.3.4 Irismenaren kudeaketa

Proiektua garatu bitartean, gerta daiteke hasiera batean zehaztutako irismenean aldaketak gertatzea, eta ondorioz hau birplanteatu behar izatea. Kasu horretan, garrantzitsua izango da proiektuaren mugarriak birplanteatzea eta hasierako plangintzan ezarritako orduen konbinatzea eraldatzea.

1.4 Lanaren Deskonposaketa Egitura

Proiektua ataza desberdinetan banatzea oso garrantzitsua izango da hasieran ezarritako irismena lortu ahal izateko. Horretarako oso baliagarria den teknika bat erabiltzea gomen-datzen da, LDE diagramak. Horietan proiektuan zehar egin beharreko lan guztia, ataza eta azpiatazetan banatzen da.

Proiektuaren arrakasta eta antolamendua bermatzeko sei ataletan banatzea erabaki da:

- **Kudeaketa**

Izenak dioen bezala, ataza honetan proiektuaren kudeaketa-lanak sartzen dira. Hasierako plangintza, proiektua aurrera eraman ahala egin beharreko jarraipen kontrola eta egindako bilera desberdinak (kudeaketa bilerak batiz bat).

- **Ikerkuntza**

Proiektuan ezer egin aurretik, beharrezkoak diren ezagutza eta informazioa lortu behar ditugu. Oso garrantzitsua da testuingurua ondo aztertzea. Azterketa sakon bat egin ondoren, teknologia egokienak aukeratzea errazagoa izango da.

- **Analisi eta diseinua**

Proiektuari dagokion eskakizunen bilketa eta diseinua egin beharko dugu. Ataza honetan beteko diren erabilpen-kasuak, sekuentzia diagramak, domeinua, datu-basearen diseinua etab. lortuko ditugu besteak beste.

- **Garapena**

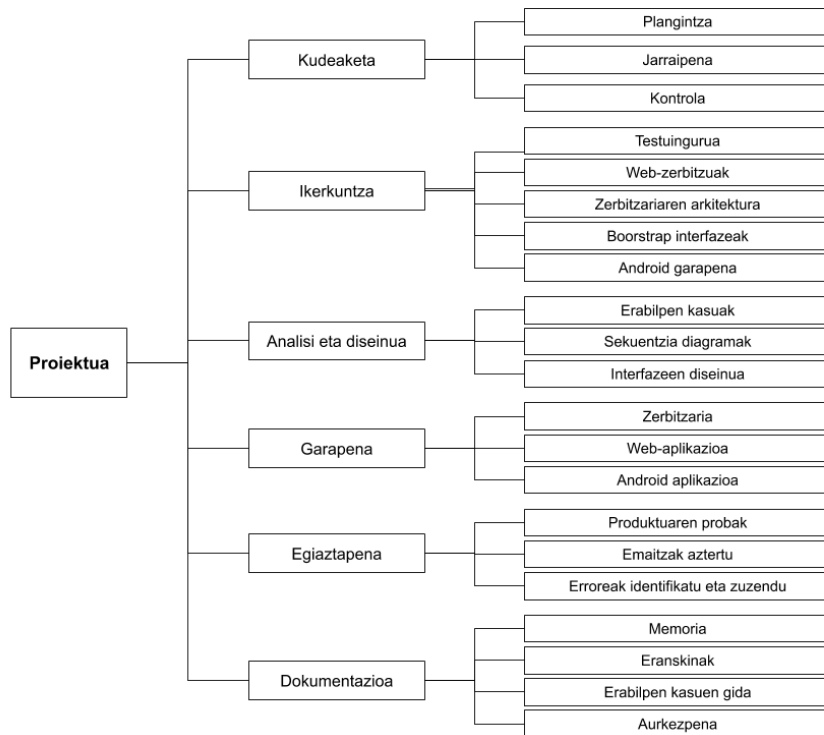
Eskakizunen bilketa eta diseinua osatu ondoren, lortutako sekuentzia diagramak jarraitu beharko dira hasieran planteatutako erabilpen-kasuak osatuz.

- **Egiaztapena**

Proiektuaren garapena bukatutzat dagoela uste dugunean, egindako lanaren egiaztapena egin beharko da. Garrantzitsua da azken bi ataza hauen artean ziklo bat egitea, proiektuari itxiera bat eman arte.

- **Dokumentazioa**

Bukatzeko, gainontzeko atal guztiak garatzen joan ahala, lan hori dokumentatu beharko da. Modu horretan proiektuaren memoria osatuko dugu, eta ondoren dagokion aurkezpena prestatu.

**1.1 Irudia:** LDE diagrama.

1.5 Atazak eta ezaugarriak

| Kodea | Ataza | Azalpena |
|-----------------------------|---------------------------------|--|
| Kudeaketa | | |
| 1.1 | Plangintza | Proiektuaren plangintza dokumentua; irismena, egutegia, kalitate-planak, mugarriak... zehaztuz. |
| 1.2 | Jarraipena | Proiektuaren egoera nolakoa den ikusteko zuzendariarekin dauden bilera desberdinak. |
| 1.3 | Kontrola | Azaltzen diren arriskuak baloratu, kalitate-maila beteko dela zihurtatu, egin beharreko aldaketak aztertu... |
| Ikerkuntza | | |
| 2.1 | Testuingurua | NBA-ren API-ak eskeintzen dituen datuak aztertu. |
| 2.2 | Web-zerbitzuak | SOAP protokoloa erreparasatu, bai zerbitzari zein bezero aldetik. |
| 2.3 | Zerbitzariaren arkitektura | Zerbitzaria garatzeko erabiliko diren teknologia egokienak aukeratu |
| 2.4 | Boorstrap | Funtzionamendua gogoratu eta adibideak ikusi |
| 2.5 | Android garapena | Android Studio herra mintza aztertu |
| Analisi eta diseinua | | |
| 3.1 | Erabilpen kasuak | Erabilpen-kasuak deskribatu |
| 3.2 | Sekuentzia diagramak | Erabilpen-kasu bakoitzari dagokion sekuentzi diagrama diseinatu |
| 3.3 | Interfazeen diseinua | Produktuak dituen aukera guztiak ahalik eta modu erosoenean bistaratzea lortu |
| Garapena | | |
| 4.1 | Zerbitzaria | Zerbitzariaren funtzioak garatu. |
| 4.2 | Web-aplikazioa | Web interfazea eta dagozkion bezero funtzioak eraiki. |
| 4.3 | Android aplikazioa | Android interfazea eta dagozkion bezero funtzioak eraiki. |
| Egiaztapena | | |
| 5.1 | Probak | Produktua probatzeko eginiko probak. |
| 5.2 | Probak aztertu | Probetan lortutako emaitzak aztertu eta ondorioak lortu. |
| 5.3 | Errore identifikatu eta zuzendu | Probetan lorturiko erroreak identifikatu, ulertu, konpondu eta berriro gertatzen ez direla ziurtatu. |
| Dokumentazioa | | |
| 6.1 | Memoria | Proiektua biltzen duen dokumentua garatu. |
| 6.2 | Eranskinak | Bilera aktak. |
| 6.3 | Erabilpen kasuen gida | Produktuak eskaintzen dituen aukerak azaldu. |
| 6.4 | Aurkezpena | Proiektuaren defentsa aurrera eramateko aurkezpena egin. |

1.1 Taula: LDE diagraman zehazturiko atazen azalpena

1.6 Emangarri nagusiak

Proiektuak lau emangarri nagusi izango ditu:

- **Produktua:** Garatu behar den zerbitzaria eta bi bezeroen exekutagarri desberdinak. Gainera guzti horren kodea *GitHub*-en (edo antzeko plataforma batean) publikatu.
- **Memoria:** Dokumentu hau, proiektua aurrera eramateko egindako planifikazioa, kudeaketa etab. azaltzen duen txostena.
- **Aurkezpena:** Defentsa aurrera eramateko egin beharreko aurkezpena.
- **Erabilpen kasuen gida:** Edozeinek proiektua erabili nahi badu, gida txiki bat. Printzipioz ez da beharrezkoa aplikazioa erabiltzeko.

1.7 Baliabideak

Proiektua aurrera eraman ahal izateko, erabiliko diren software-baliabideak zeintzuk izango diren zehaztu beharko ditugu. Ahal den neurrian, eduki irekiak aukeratuko dira. Hala ere, lanerako erabiliko den ekipo garrantzitsuena Windows 10 bertsioa duen konputagailua izango da.

Zerbitzaria garatu ahal izateko, *Eclipse* IDE-a erabiliko dugu eta programatzeko hizkuntza Java. Aldi berean, zerbitzari honek datu-base erlazional bat erabiliko du *SQL* eta *Hibernate* framework-a erabiliz ezarriko du konexioa zerbitzariarekin.

Web aplikazioa PHP eta HTML bidez idatzia egongo da, eta interfazei diseinua emateko Bootstrap erabiliko da.

Android aplikazioa *Android Studio* erabiliz garatuko da, Java erabiliz.

1.8 Proiektua gauzatzeko plana

Horrelako lan bat modu egokian aurrera eramateko, ezinbestekoa izango da plan bat diseinatzea. Horretarako, lan-metodologia, egutegia, Gantt diagrama eta lan-karga finkatzea beharrezkoa izango da.

1.8.1 Lan-metodologia

Oso garrantzitsua da lan-metodologia bat zehaztea proiektua ahalik eta egokien garatzeko. Modu horretan, denbora galerak ekiditea eta lana eraginkorragoa izatea bermatuko dugu.

- Gaiari dagokion ikerkuntza egin, ideia guztiak argi edukiz.
- Domeinuaren diseinua eta gertaera fluxua finkatu.
- Erabilpen-kasuak eta interfazeen diseinua egin.
- Zerbitzariaren atala garatu.
- Bezeroak eraiki.

1.8.2 Dedikazioaren estimazioa

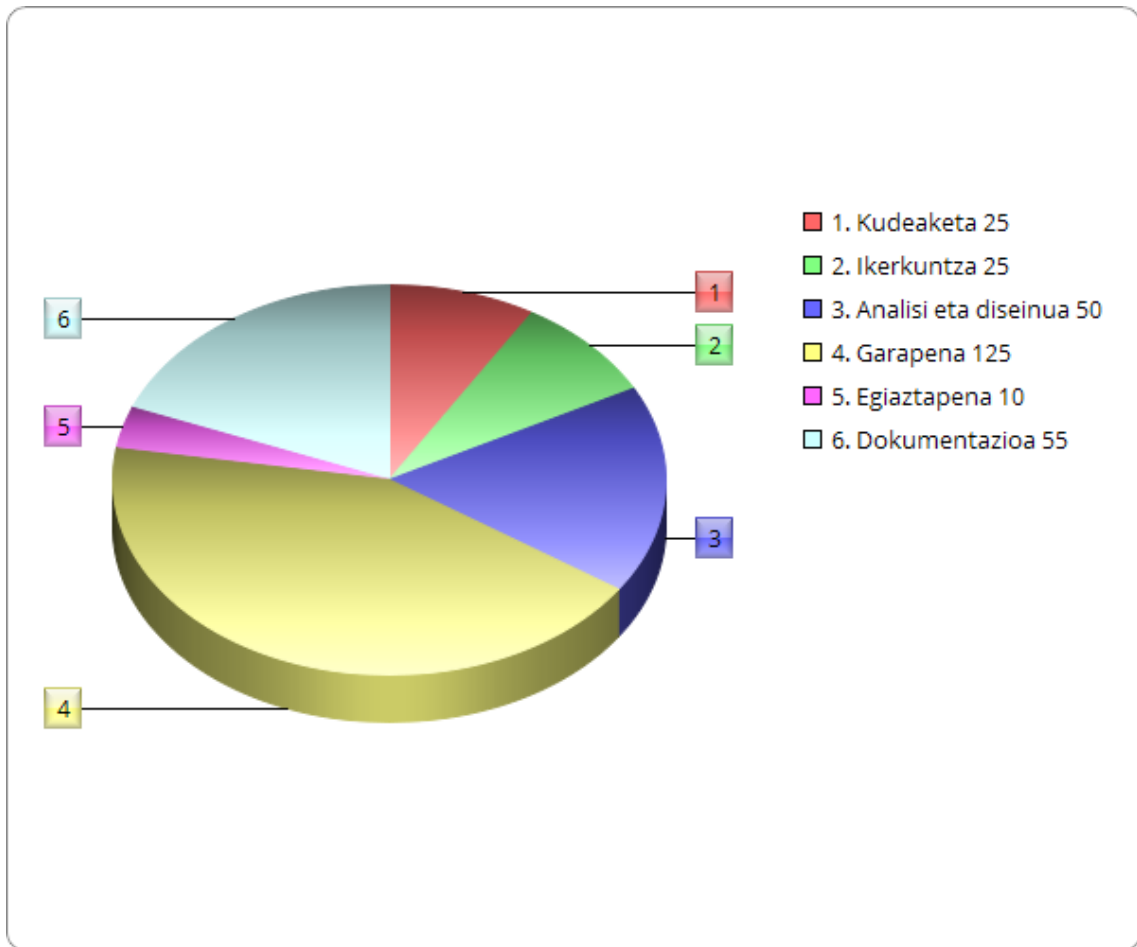
Proiektu hau Informatika Ingeniaritzako Gradu Amaierako lana da, eta gutxi gorabehera 300 orduko dedikazioa eskatzen du. Lehenengo aldia denez horrelako proiektu bat kudeatzen dugula, kontuan hartu behar dugu desbiderapen handiak egoteko aukera. Hori dela eta, aurretik kontuan hartu behar dugun zerbait da.

Proiektuen Kudeaketan ikasi genuen bezala, ataza bakoitzari ordu kopuru baten estimazioa jarri behar diogu, beti ere atzerapenak gerta daitezkeela kontuan izanik.

Hori guztia kontuan harturik, hau izango zen ataza bakoitzaren denbora estimazioa.

| Kodea | Ataza | Estimazioa |
|--------------------------------------|-----------------------------------|------------|
| Kudeaketa | | |
| 1.1 | Plangintza | 10 ordu |
| 1.2 | Jarraipena | 10 ordu |
| 1.3 | Kontrola | 5 ordu |
| Kudeaketa: 25 ordu | | |
| Ikerkuntza | | |
| 2.1 | Testuingurua | 5 ordu |
| 2.2 | Web-zerbitzuak | 5 ordu |
| 2.3 | Zerbitzariaren arkitektura | 5 ordu |
| 2.4 | Boorstrap interfazeak | 5 ordu |
| 2.5 | Android garapena | 5 ordu |
| Ikerkuntza: 25 ordu | | |
| Analisi eta diseinua | | |
| 3.1 | Erabilpen kasuak | 10 ordu |
| 3.2 | Sekuentzia diagramak | 35 ordu |
| 3.3 | Interfazeen diseinua | 5 ordu |
| Analisi eta diseinua: 50 ordu | | |
| Garapena | | |
| 4.1 | Zerbitzaria | 50 ordu |
| 4.2 | Web-aplikazioa | 35 ordu |
| 4.3 | Android aplikazioa | 40 ordu |
| Garapena: 125 ordu | | |
| Egiaztapena | | |
| 5.1 | Produktuaren probak | 4 ordu |
| 5.2 | Emaitzak aztertu | 1 ordu |
| 5.3 | Erroreak identifikatu eta zuzendu | 5 ordu |
| Egiaztapena: 10 ordu | | |
| Dokumentazioa | | |
| 6.1 | Memoria | 45 ordu |
| 6.2 | Eranskinak | 2 ordu |
| 6.3 | Erabilpen kasuen gida | 3 ordu |
| 6.4 | Aurkezpena | 5 ordu |
| Dokumentazioa: 55 ordu | | |
| Guztira: 290 ordu | | |

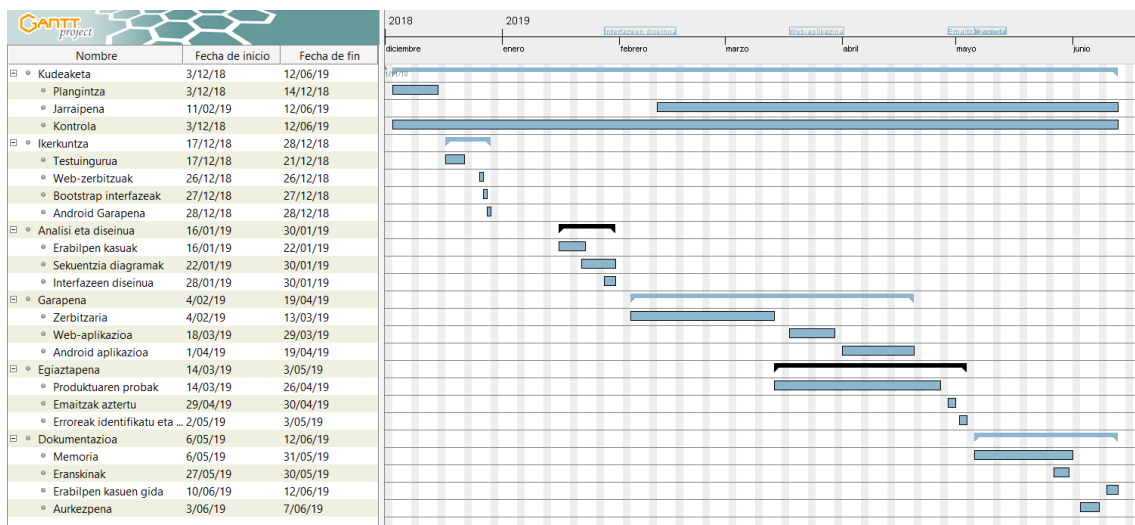
1.2 Taula: Ataza bakoitza burutzeko ordu estimazioa



1.2 Irudia: Dedikazio estimazioaren grafikoa.

1.8.3 Gantt diagrama

Ataza bakoitzari ezarri diogun ordu kantitateak egutegi batean kokatzeko modurik egokiena *Gantt* diagrama bat erabiltzea da. Modu horretan, ataza bakoitzaren estimazioa zein den ikusi daiteke, eta proiektuan zehar une oro egiten ari garen lana zein den eta ondoren datorrena.



1.3 Irudia: Gantt diagrama.

1.8.4 Lan-karga

Gantt diagraman jarri dugun bezala, proiektuaren hasiera eguna 2018ko abenduaren 3a izango da eta 2019ko ekainaren 12an itxiera. Guztira 138 lan egun daude, hau da, egunean 2-3 orduko lan karga.

Hala ere, hau testuinguruan jarri behar dugu. Urtarrila hasieran lehenengo lauhileko azterketak izango dira, baina ondoren 10 egunetan klaserik ez dagoenez bertan denbora gehiago izango dut.

Bigarren lauhilekoan klase gutxiago izango ditut, eta gainera klaseak apirilaren 12rako bukatuak egongo dira. Beraz, hor berriz ere denbora gehiago izango dut atzerapenik gertatzen baldin bada.

1.9 Kalitate-plana

Lortutako produktua hasieran pentsatu genuena bezalako dela neurtzeko modurik onena probatzea izango da. Modu horretan neurtu dezakegu proiektuak izandako arrakasta. Hori horrela den arren, produktuak kalitate minimo bat duela bermatzeko zenbait adierazle finkatuko ditugu eta horien kontrola egin.

1.9.1 Kalitatearen adierazleak

- **Zerbitzariak funtzioak prozesatzeko abiadura.**

- *Deskribapena:* Zerbitzariak funtzioak arintasunez prozesatzea. Modu horretan bezero aplikazioak itxaron beharreko denbora txikiagoa izango da, erabiltzailearen esperientzia hobetuz.
- *Neurtzeko modua:* Funtzioak prozesatzeko beharrezkoa duen denbora.
- *Maximo onargarria:* 5 segundo
- *Helburua:* Segundo bat edo gutxiago

- **Bezeroen fidagarritasuna.**

- *Deskribapena:* Bezeroaren aldetik hainbat erabilpen kasu egingo ditu bere saioetan, hainbat funtziori deituz eta emaitzak pantailaratu. Helburua denak ondo funtzionatzea.
- *Neurtzeko modua:* Gertatzen diren errore kopurua.
- *Minimo onargarria:* 100 deietatik 95ek ondo funtzionatzea.
- *Helburua:* Erabiltzaile guztiek saio guztietan arazorik ez gertatzea.

- **Kodearen ulergarritasuna.**

- *Deskribapena:* Proiektu bat garatzerakoan, kontuan eduki behar da kodearen implementazioaren argitasuna. Bereziki informatika arloan oso garrantzitsua den atal bat da, beste pertsona batek kodea irakurri edo aldatu behar baldin badu etorkizunean.

- *Neurtzeko modua*: Aldagaien izenak, anotazioak, espazioak kontrolatzea...
- *Minimo onargarria*: Funtzio guztien funtzionamendua anotazioen bidez argitzea.
- *Helburua*: Minimoaz gain, aldagaien izenak argiak izatea, lerro saltoak errespetatzea, ezker margina ondo eramatea... hau da, itxura elegantea izatea.

- **Interfazearen diseinua.**

- *Deskribapena*: Interfazea edozein gailutan funtzionala eta argia izatea, hau da, moldagarria. Erabilerraza izatea oso garrantzitsua da.
- *Neurtzeko modua*: Interfazeak moldaturik dauden gailu kopurua, bilaketak modu argian azaltzea eta klik kopurua gutxitzea.
- *Minimo onargarria*: Interfazeak gutxienez ordenagailuetarako pentsatuak egotea.
- *Helburua*: Ordenagailuez gain mugikorretan eroso izatea eta edozein erabilpen-kasu 3 kliketara egotea.

- **Memoriaren kalitatea.**

- *Deskribapena*: Formatu egokia, eduki argiak eta modu osatuan izatea.
- *Neurtzeko modua*: Proiektu guztia dokumentatzea prozesu guztia azalduz, modu argi eta antolatu batean.
- *Minimo onargarria*: Memoriak aurretik ezarritako oinarrizko egitura izatea.
- *Helburua*: Edukiak ondo antolatuta egoteaz gain, prozesu guztia modu oso batean azaltzea, beharrezkoa den media erabiliz (argazkiak, grafikoak, taulak ...).

1.9.2 Kalitatea ziurtatzea

Proiektu guztian zehar egingo den jarraipen eta kontrolaren bitartez, kalitate plana ondo betetzen ari dela egiaztatuko dugu. Modu horretan, gutxienez oinarrizko kalitatea edo kalitate minimoa duela jakingo dugu.

1.9.3 Kalitatearen kontrola

Zehaztu ditugun kalitatearen adierazleak betetzen direla kontrolatu beharko dugu. Gomendagarria da kontrol hau konstantea izatea, baina hala ere, kontrol zehatzena bukaeran egingo da. Honen helburua adierazleak betetzea izango da.

1.10 Arriskuen analisia

Proiektua aurrera modu egokian eramateko, ezinbestekoa da egon daitezkeen arriskuak aurre identifikatzea eta hauei konponbide bat planifikatzea. Modu horretan arazo bat bihurtzea ekiditeko ahalmena izango dugu. Arriskuak modu askotan sailkatzeko aukera dugu.

Alde batetik arrisku ezagunak eta ezezagunak ditugu. Arazoa ezaguna baldin bada, hori identifikatu eta erantzun bat planifikatzeko aukera izan dezakegu. Aldiz arazoa ezezaguna baldin bada, ezingo dugu modu proaktibo batean kudeatu, eta hor arriskua handitu egingo da.

Bestalde arriskua non dagoen multzokatu dezakegu. Hau da, arriskua produktuaren edo proiektuaren gainean dagoen finkatu.

Bi sailkapenak kontuan hartuta, ondorengoa da lortuko genukeen arriskuen analisia:

- **Zerbitzariak funtzionatzeari uztea.**

- *Arriskuaren deskribapena:* Produktuaren atal garrantzitsuena zerbitzaria beti martxan egotea da. Izan ere, zerbitzariak funtzionatzeari uzten baldin badio, bezero atalek ez dute ezertarako balio eta arazo larriak sortuko ziren. Arazoa konpondu arte aplikazioak ez lukeen funtzionatuko.
- *Probabilitatea:* Txikia
- *Eragina:* Oso handia (hondamendia)

- **Zerbitzariaren ekintzak motelak izatea.**

- *Arriskuaren deskribapena:* Bezero ugari aldi berean zerbitzariari eskaerak egiten saiatzen baldin badira, aukera dago zerbitzariak ekintzak modu motelean

egitea (bereziki datu-baseari dei ugari egiten baldin badizkio). Ondorioz, bezeroek modu astiroago batean ibiliko dira.

– *Probabilitatea*: Ertaina

– *Eragina*: Ertaina

- **Aplikazioaren interfazea zatarra izatea.**

– *Arriskuaren deskribapena*: Aukera dago interfazea (bereziki web orrikoa) gaur egungoekin konparatuz gero itsusiagoa izatea.

– *Probabilitatea*: Handia

– *Eragina*: Txikia

- **Produktuaren kodea galtzea.**

– *Arriskuaren deskribapena*: Produktuaren kodea ekipo bakar batean garatuko denez, ordenagailu horretan egongo da biltegitua. Makina horrek arazorik baldin badu kode guztia galtzeko aukera egongo zen. Hori saihesteko, GitHub edo Drive motako biltegitratzeak erabiltzea gomendatzen da.

– *Probabilitatea*: Txikia

– *Eragina*: Handia

- **Proiektuaren memoria galtzea.**

– *Arriskuaren deskribapena*: Memoria askotan irekiko dugun fitxategi bat da (bereziki egunak pasa ahala), horregatik beti existitzen da hau errorez ezabatzeko aukera (edo zatiren bat ezabatu eta gu ez ohartu). Aldi berean gailua apurtzen bada, memoria galtzeko aukera dago. Horregatik, kodearekin batera memoria beste biltegitratze batean gordetzea gomendatzen da.

– *Probabilitatea*: Ertaina

– *Eragina*: Galdu den zatiaren araberakoa (oso handia izan daiteke).

- **Denbora faltaren ondorioz, proiektuan atzerapenak izatea.**

- *Arriskuaren deskribapena:* Proiektuaren garapenean zehar hainbat motako arazoak gertatzeko aukera dago: teknikoak, garapenarekin arazoak, gaixotasunak, arazo pertsonalak, arazo akademionen ondorioz denbora gutxiago izatea...
- *Probabilitatea:* Ertaina
- *Eragina:* Arazoaren mailaren arabera. Agian hasierako plangintzan desbiderapen txiki bat suposatu dezake edo plangintza berriz egitea eta entrega atzeratzea.

1.11 Interesatuak

Kasu honetan, proiektua arrakastaz amaitzeko interesatu nagusia egilea bera izango da, hau da, ikaslea. Bera izango da proiektua aurrera eramateko erantzukizuna izango duena. Horrez gain, proiektuko zuzendaria interesatua izango, nire kasuan Jon Iturrioz. Beraren ardura nagusia proiektuak beharrezkoak dituen atalak azaltzea eta proiektuan zehar gidatzea izango da, sortzen diren zalantzak argituz.

Bukatzeko, Software Ingeniaritza espezialitateko Gradu Amaierako Lana denez, espezialitate horretako epaimahaia osatzen duten irakasleak izango dira azken interesatuak.

1.11.1 Komunikazio-plana

Alde batetik zuzendariarekin egingo den komunikazioa dugu. Proiektuaren garapenaren zehar agertzen diren zalantzak argitzeko berarekin harremanetan jarriko gara. Zalantza txikia baldin bada posta elektronikoa erabiliko dugu, aldiz arazoa handiagoa bada zehazturiko bilera bat egingo da. Bilerak adosteko eguna posta elektronikoa erabiliz egingo da.

2. KAPITULUA

Eskakizunen bilketa

Aplikazio on bat garatzeko, lehen urratsa beharrak ondo antzematea da. Kapitulu honetan proiektuaren testuingurua ulertu ondoren, sistemaren logika deskribatzen da. Hiru ataletan banatua dago: aurrekariak, erabilpen kasuak, domeinuaren eredia eta gertaera fluxuak.

2.1 Aurrekariak

2.1.1 Testuingurua

Aplikazioaren zergatia ulertzeko garrantzitsua da testuingurua azaltzea. Alde batetik ezinbestekoa izango da NBAren egitura ulertzea, eta bestetik erabiltzaileek gure moduko aplikazioekin horrelako txapelketetan duten parte-hartzea.

Hasteko NBA dugu, hau da, Amerikako Estatu Batuetako Saskibaloia liga ofiziala. Ez nahiz alderdi oso teknikoan sartuko eta soilik gure proiektuan eragina izango duten gauzak aztertuko ditut.

Liga 30 taldek osatzen dute, talde bakoitza gutxienez 13 jokalariz osatua dago. Aldi berean, talde guztiek jokalariei ordaindu dezaketen maximo bat dute, ondorioz jokalaria onenak talde desberdinetan banatzen dira ahal duten diru kantitate handiena irabaziz. Zergatik da hau garrantzitsua? Jokalari horrek duen urteko soldata izango da guk ezarriko diogun merkatu balioa. Modu horretan datuak automatikoki lortuko ditugu.

Talde bakoitzak 82 partidu jolastuko ditu denboraldi guztian zehar (playoff-ak kontatu ga-

be). Ez da egutegi bateraturik existitzen, eta ondorioz, ia egunero egongo dira partiduak. Modu horretan oso zaila da jardunaldiak zehaztea, beraz, asteak izango dira neurria. Aste bakoitzean jokaturako partiduen batezbestekotik lortuko dira jokalarien puntuak (eta ondorioz erabiltzailearena).

Komentatu berri ditudan datu guzti hauek NBA-ko API-tik eskuratzeko aukera izango dugu. Datuak *JSON* moduan irakurriko ditugu eta hauek dira URL-ak:

| Lortu nahi den datua | URL | Parametroak |
|----------------------------|---------------------------------------|--|
| | https://data.nba.net/10s/prod/v1/... | Adibideak... |
| Ligako jokalarien zerrenda | v1/{{year}}/players.json | year=2018 |
| Ligako taldeen zerrenda | v2/{{year}}/teams.json | year=2018 |
| Eguneko partidu zerrenda | {{gameDate}}/scoreboard.json | gameDate=20190201 |
| Partiduaren estadistikak | {{gameDate}}/{{gameId}}_boxscore.json | gameDate=20190201 gameId=0021800771 |

Bestalde erabiltzaileen aldea dugu. Horrelako aplikazioak elkarren artean txapelketak egiteko pentsatuak daude. Erabiltzaile bakoitzak jokalaria batzuk izango ditu, nahiz eta denboraldi guztian zehar aldaketak egiteko aukera izan. Modu horretan, jokalaria horien arabera erabiltzaileak puntuazio batzuk lortuko ditu eta gainontzeko erabiltzaileekin lehiatzeko aukera izango du.

Aplikazio honek sare sozial baten antzera funtzionatzen du. Gure kasuan, erabiltzaile bakoitzak soilik bere komunitateko erabiltzaileekin edukiko du harreman zuzena.

2.1.2 Antzeko aplikazioak

Merkatuan antzerakoak diren aplikazio ugari aurkitu ditzakegu. Adibidez, espainiar futbol ligarako hainbat aukera ditugu *Comunio*, *Futmondo*, *Marca Fantasy* besteak beste. Txirrindularitzarako ere *Tropela* izeneko aplikazioa dugu. Saskibaloirako (eta konkretuki NBA-ko ligarako) ere existitzen dira, horien artean aipagarrienak *AlleyoopManager*, *ESPN Fantasy Basketball* eta *Movistar Fantasy* espainiarra, baina besteekin alderatuz oso desberdina da.

Zergatik da beharrezkoa gure aplikazioa? Egia da merkatuan antzerakoak diren aplikazioak daudela, baina konkretuki NBA-ko aplikazioak gainontzekotik nahiko desberdinak

dira. Hau batez ere Ameriketako ohitura eta gustuengatik gertatzen den zerbait da. Bertan nahiago dute egunerokoak garrantzia izatea, eta aldiz hemen, nahiago dugu aste guztian zehar ondorengo astea prestatzea. Hori garbi ikusi dezakegu *Movistar Fantasy*-an. Soilik erderaz dagoen aplikazio bat da, eta nahiz eta NBAko ligaren txapelketa bat izan, puntuak asteka ematen dira.

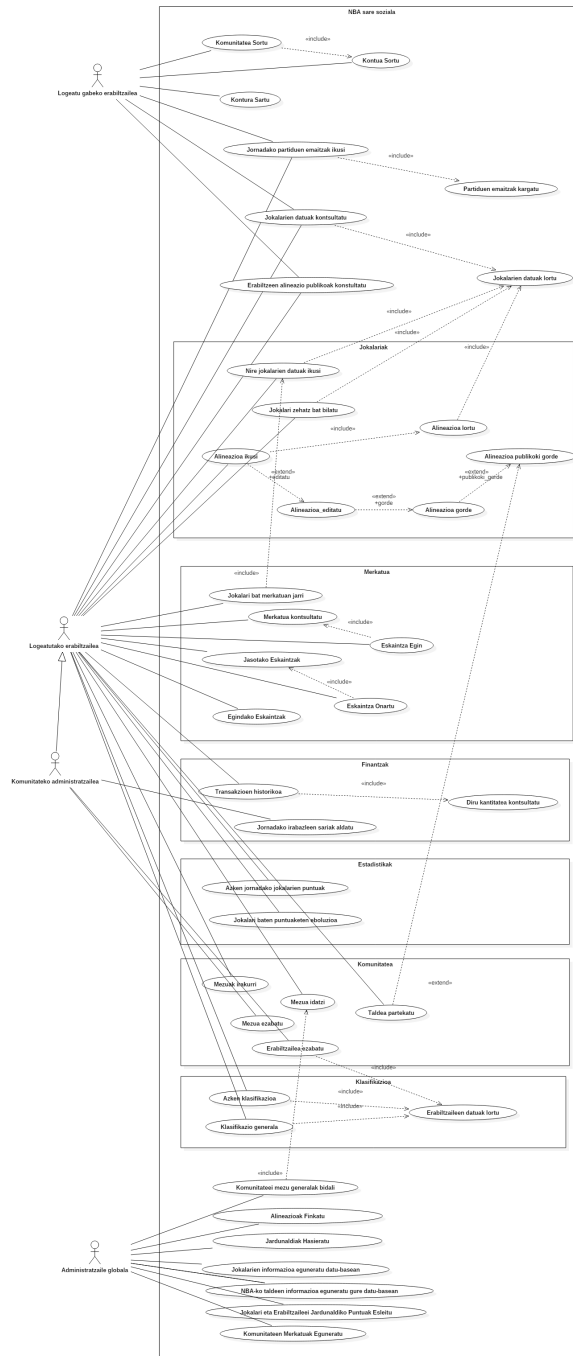
Orduan, zein da gure helburua? Europear kulturara egokitutako aplikazio bat sortzea, AEBko saskibalo ligan oinarritua. Horretarako hemen arrakasta duten aplikazioen zati interesgarrienak inplementatu eta bertakoekin konbinatuko ditugu.

| | Nire aplikazioa | Comunio | Alleyoop Manager | LaLiga Fantasy Marca | ESPN Fantasy Basketball | Movistar Fantasy |
|--|------------------------|----------------|-------------------------|-----------------------------|--------------------------------|-------------------------|
| Kirola | <i>NBA</i> | <i>Futbola</i> | <i>NBA</i> | <i>Futbola</i> | <i>NBA</i> | <i>NBA</i> |
| Partiduen emaitzak ikusi | X | | | | X | |
| Jokalarien estadistikak kontsultatu | X | X | X | X | X | X |
| Erabiltzaileek proposaturiko taldeak ikusi | X | | | X | | X |
| Koloreak aldatu | X | | | | | |
| Hizkuntza aldatu | X | X | | | | |
| Kontua sortu/sartu | X | X | X | X | X | X |
| Liga sortu | X | X | X | X | X | X |
| Jokalari zehatz bat bilatu | X | X | X | X | X | X |
| Alineazioa ikusi | X | X | X | X | X | X |
| Alineazioa editatu/gorde | X | X | X | X | X | X |
| Alineazioa publikoki gorde | X | | | | | |
| Merkatua kontsultatu | X | X | X | | X | |
| Jokalaria merkatuan jarri | X | X | X | | X | |
| Merkatuko jokalaria bati oferta bat egin | X | X | X | | X | |
| Saldu diren jokalarien informazioa | X | X | X | | X | |
| Traspaso bat ezeztatu | X | | | | | |
| Transakzioen historikoa | X | X | X | | X | |
| Jornadako irabazleen sariak aldatu | X | | | | | |
| Azken jardako jokalarien puntuak | X | X | X | X | X | X |
| Jokalari baten puntuen eboluzioa | X | X | X | X | X | X |
| Mezuak irakurri | X | X | | | | |
| Mezuak idatzi | X | X | | | | |
| Erabiltzailea ezabatu | X | | | | | |
| Mezua ezabatu | X | | | | | |
| Azken klasifikazioa | X | X | X | X | X | X |
| Klasifikazio generala | X | X | X | X | X | X |
| Komunitateei mezu generalak bidali | X | X | | | | |
| Komunitateak ikusi eta ezabatzeko aukera | X | | | | | |
| Jokalarien eta erabiltzaileen puntuak jarri | X | X | X | X | X | X |
| Jokalarien informazioa eguneratu datu-basean | X | X | X | X | X | X |

2.1 Taula: Antzeko produktuen konparaketa

2.2 Erabilpen Kasuak

Ondorengo irudian ikusi dezakegu erabilpen-kasu guztien irudi orokorra. Erabilpen-kasuak hobeto azaltzeko, aktore bakoitzarenak aztertuko ditugu independenteki.



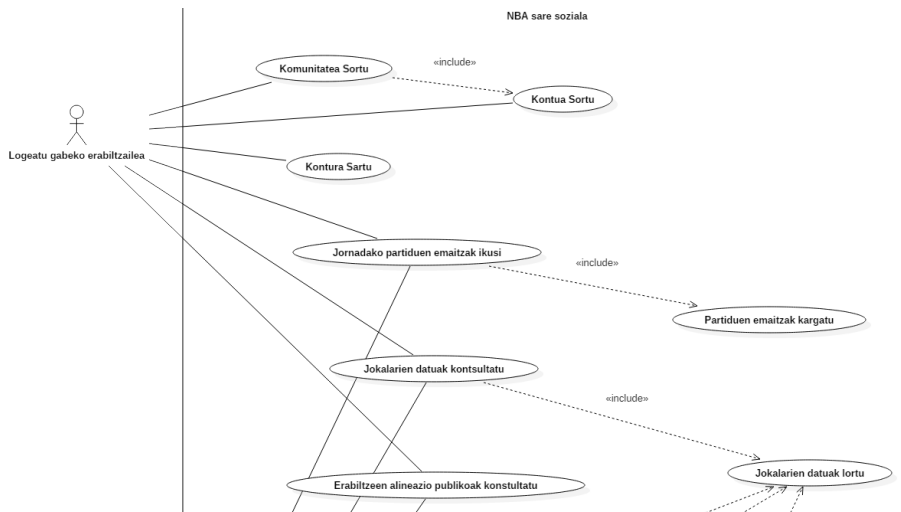
2.1 Irudia: Erabilpen Kasuak.

2.2.1 Logeatu gabeko erabiltzailea

Aplikazioan sartu berri den erabiltzailea. Normalean erabiltzaile berriak izaten dira, logika gutxi duelako ohiko erabiltzailea izatea eta konturik ez edukitzea.

Erabiltzaile mota honek hiru ekintza propio ditu *Komunitatea sortu* (eta ondorioz komunitate horretako administratzaile kontua), *Erabiltzaile kontua sortu* (SingUp) eta *Kontura sartu* (LogIn).

Horretaz gain, edozein erabiltzailek bezala *Partiduen emaitzak*, *Jokalarien datuak* eta *Erabiltzaileek publikaturiko boskoteak* ikusteko aukera edukiko du.

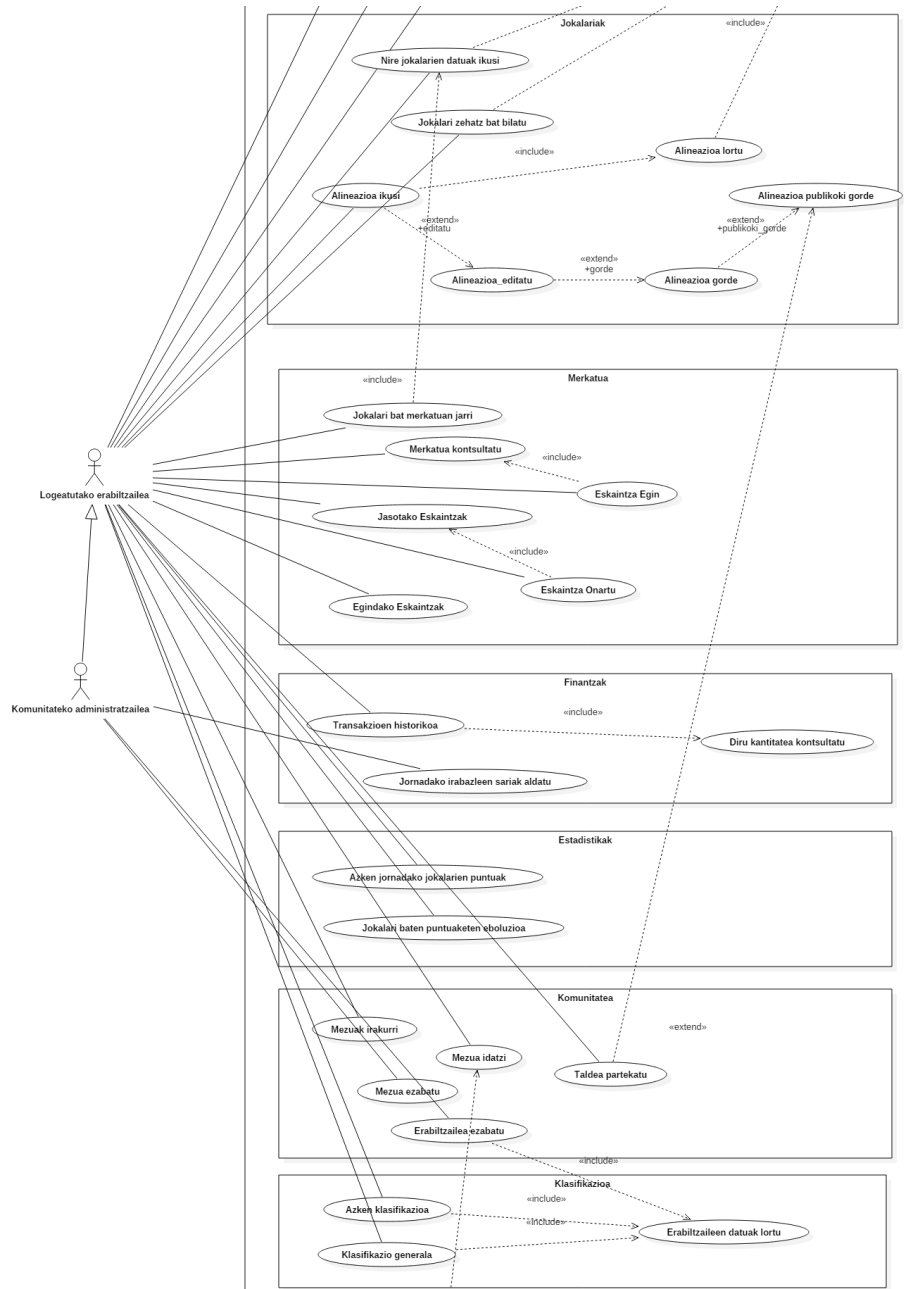


2.2 Irudia: Logeatu gabeko erabiltzailearen erabilpen kasuak.

2.2.2 Logeaturako erabiltzailea (baita komunitateko administratzailea)

Bi rol hauen arteko desberdintasuna, administratzaileak erabilpen kasu gutxi batzuk gehiago dituela da, zehazki *Komunitateko sariak aldatu*, *Mezuak ezabatu* eta *Erabiltzaileak ezabatu*.

Gainontzeko guztiak komunean dituzte (hauei lehen aipaturiko hiru erabilpen-kasu komunak gehitu): *Erabiltzailearen jokalarien kontsulta*, *Alineazioa ikusi*, *aldatu*, *orde eta publikatu*, *Jokalari bat merkatuan jarri*, *Merkatua kontsultatu*, *Eskaintzak egin*, *jaso*, *onartu eta kontsultatu*, *Finantzak ikusi*, *Jokalarien estatistikak aztertu*, *Mezuak irakurri eta idatzi* eta *Erabiltzaileek klasifikazioak ikusi*.

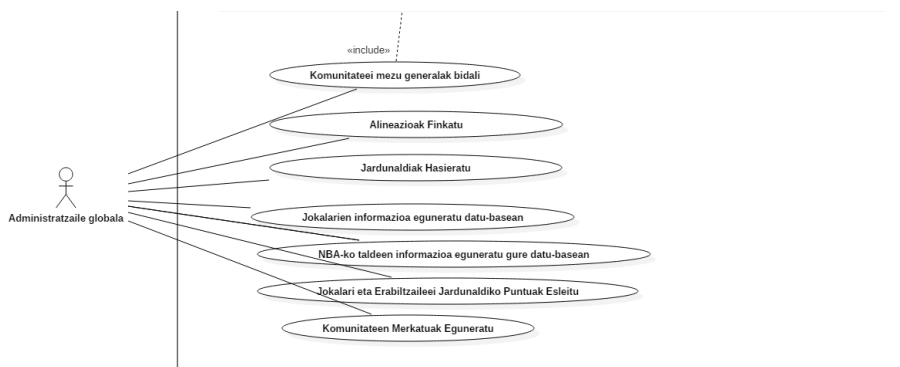


2.3 Irudia: Logeatutako erabiltzailearen erabilpen kasuak.

2.2.3 Administratzaile orokorra

Aplikazioa erabiliko duten publikotik kanpo dagoen rol-a. Erabilpen-kasu teknikoagoak dira, oso automatizatuak eta konplexutasun handikoak. Komunitateen egoera eguneratzeaz arduratzen da, modu horretan aplikazioak beti informazio berria edukiko du.

Funtzionalitateen artean *Mezu orokorrak bidali*, *Alineazioak finkatu*, *Jardunaldiak hasieratu*, *Jokalarien eta taldeen informazioa eguneratu*, *Jokalarien eta erabiltzaileek puntuazioak esleitu* eta *Komunitateen merkatuak eguneratu*.



2.4 Irudia: Administratzaile orokorraren erabilpen kasuak.

- **Erabiltzailea:** Bertan erabiltzailearen inguruko informazioa jasoko dugu. Erabiltzaile bakoitzak hainbat jokalariren jabea izango da, eta komunitate horretan beste inork ezingo du jokalaria hori eduki. Aldi berean, jokalaria merkatuan jartzeko eta merkatuko gainontzeko jokalariei eskaintzak egiteko aukera izango du. Erabiltzaile bakoitza komunitate bakar batean egongo da eta jardunaldi bakoitzean puntuazio bat esleituko zaio.
- **Eskaintza:** Erabiltzaile bakoitzak merkatuko jokalaria bati egindako eskaintzaren balioa gordeko da. Eskaintza bakoitza merkatuko jokalaria bati dagokio eta erabiltzaile bakoitzak nahi beste eskaintza egin ahal izango ditu.
- **Jardunaldia:** Jardunaldiak administratzaile orokorrak sortuko dituen objektu bat izango da. Liga hasten den lehen astetik bukaeraraino jardunaldiak egongo dira. Hasierako eguna jardunaldi hori hasten den astelehenari dagokio eta jardunaldiak astelehenetik igandera izango dira.
- **Jokalaria:** Informazio hau NBAko datu-basetik gure datu-basera pasako da. Bertan jokalarien datu pertsonalak gordeko dira. Jokalaria bakoitza gehienez talde bakar baten parte izan daiteke, eta jardunaldi bakoitzean puntuazio bat esleituko zaio.
- **Jokalariaren Partidua:** Objektu honek jokalaria batek partidu zehatz batean eduki dituen estatistikak gordetzen ditu. Hau soilik administratzaile globalak erabiliko du jardunaldiko puntuazioak ezartzerako orduan. Hori dela eta, *datu-basean gordeko ez den datu* bat izango da.
- **Komunitatea:** Komunitate bakoitzak hainbat erabiltzaile izango ditu (10 gehienez) eta horietatik bat komunitateko administratzailea. Aldi berean, komunitate bakoitzak merkatuko jokalaria propioak izango ditu.
- **Merkatuko Jokalaria:** Komunitate bakoitzak jokalarien merkatu propioa izango du, beraz, Merkatuko Jokalaria bakoitza komunitate bakar batean izango da. Bakoitzak Jokalaria bat errepresentatuko du, jabe bat edukiko du eta tramitatu ondoren jabe berria baldin badu, erabiltzaileari jokalaria hori ezarriko zaio jabe zaharrari kenduz. Gainera, Merkatuko Jokalaria bakoitzak hainbat eskaera izango ditu.
- **Mezua:** Erabiltzaile bakoitzari bere komunitatean mezuak jartzeko aukera eskainiko zaio. Mezu bakoitza erabiltzaile bakarrarena izango da eta komunitate bakar batean egongo da ikusgarri. Gainera, ezabagarriak izateko aukera dago.

- **Partidua:** Partiduaren informazio basikoa gordeko du. Jokalariaren partiduarekin gertatzen den bezala, partidua klasea *ez da datu-basean gordea izango*. Aplikazioak partiduen emaitzak denbora errealean emateko aukera eskaintzen duenez, ez du zentzurik partiduak datu-basean gordetzea, hobeto da beharrezkoa den kasuetan NBAkoari eskatzea.
Partidu bakoitzean bi taldek hartuko dute parte beti (etxeakoak eta kanpokoak), eta bietako jokalariek (Jokalariaren Partidua).
- **Puntuazioa Erabiltzailea Jardunaldia:** Jardunaldi horretan ezarritako alineazioan zeuden jokalariek lortutako puntuen batuketa, izango dira erabiltzaileak lorturiko puntuak.
- **Puntuazioa Jokalaria Jardunaldia:** Jardunaldi horretan jokalaria egindako partiduen arabera, algoritmo baten bidez puntuazio bat lortuko da. Jokaturiko partiduen batezbestekoaren bidez ezarriko dira puntuak.
- **Taldea:** Jokalariekin gertatzen den bezala, informazio hau NBA-ko datu-basetik gurera pasako da. Bertan 30 frankizien informazioa gordeko dugu.
- **Transakzioa:** Posible da domeinu guztian inportantzia gutxien duen klasea izatea. Ez da eragiketa bat bera egiteko erabiltzen, erabiltzaileak beraren mugimenduak ikusteko soilik balio du. Bertan eguna, transakzioaren kantitatea eta mezu pertsonalizatu bat izango du.

2.4 Gertaera Fluxuak

2.4.1 Logeatu gabeko erabiltzailea

Komunitatea sortu (eta administratzailea)

- *Fluxu arrunta:*

1. **Erabiltzaileak** komunitateko datuak (komunitateko nick-a, izen-osoia, ongie-torri mezua, pasahitza (x2) eta jardunaldiko irabazleen sariak (x3)) sartzen ditu.
2. **Sistemak** komunitate nick hori aurretik erregistratua ez dagoela konprobatu ondoren, sartutako datuekin Komunitate objektu bat sortuko du.
3. **Erabiltzaileak** erabiltzailearen nick, email, izen osoa, pasahitza (x2) sartzen ditu
4. **Sistemak** nick hori duen beste erabiltzailerik ez dagoela konprobatu ondoren, Erabiltzaile objektua sortzen du (aurreko komunitatearekin erlazionatuz). Ondoren, erabiltzaile horri hainbat Jokalari esleituko zaizkio ausaz. Bukatzeko aldaketa guztiak datu basean gauzatuko dira.

- *Kasu alternatiboak:*

- Sartutako komunitateko nick-a existitzen da .
- Komunitateko pasahitzak ez datoz bat.
- Erabiltzailearen nick-a aurretik existitzen da.
- Erabiltzaileak pasahitzak sartzean hauek ez ziren berdinak.
- Komunitateko nick eta pasahitza ondo sartu ondoren, bertan behera utzi eragiketa.

Kontua sortu

- *Fluxu arrunta:*

1. **Erabiltzaileak** komunitateko nick-a eta pasahitza sartzan ditu
2. **Sistemak** sartutako komunitateko nick-a duen beste komunitate bat existitzen dela eta pasahitz hori duela frogatuko du. Orduan Komunitate objektu hori lortuko du.
3. **Erabiltzaileak** erabiltzailearen nick, email, izen osoa, pasahitza (x2) sartzan ditu
4. **Sistemak** nick hori duen beste erabiltzailerik ez dagoela konprobatu ondoren, Erabiltzaile objektua sortzen du (dagokion komunitatearekin erlazionatuz). Ondoren, erabiltzaile horri hainbat Jokalari esleituko zaizkio ausaz. Bukatzeko aldaketa guztiak datu basean gauzatuko dira.

- *Kasu alternatiboak:*

- Sartutako komunitateko nick-a ez da existitzen.
- Nick hori duen komunitatea existitzen da, baina pasahitza ez da hori.
- Erabiltzailearen nick-a aurretik existitzen da.
- Erabiltzaileak pasahitzak sartzean hauek ez ziren berdinak.
- Komunitateko nick eta pasahitza ondo sartu ondoren, bertan behera utzi eragiketa.

Kontura sartu

- *Fluxu arrunta:*

1. **Erabiltzaileak** erabiltzailearen nick-a eta pasahitza sartzan ditu.
2. **Sistemak** sartutako erabiltzailearen nick-a duen beste erabiltzaile bat existitzen dela eta pasahitz hori duela frogatuko du. Ondoren, komunitatearen eta erabiltzaile objektuak lortuko ditu eta logeaturiko pantailara joango da.

- *Fluxu alternatiboak:*

- Sartutako nick eta pasahitzarekin ez da erabiltzailerik lortzen.

2.4.2 Logeatutako erabiltzailea

Jokalariak

Erabiltzailearen jokalarien datuak ikusi

- *Fluxu arrunta:*

1. **Erabiltzaileak** bere jokalariak ikustea aukeratu.
2. **Sistemak** erabiltzailearen Jokalari objektu guztiak eskuratu eta hauek bistaratu.

- Ez da *kasu alternatiborik* existituko, erabiltzaile orok jokalariak izango dituelako. Beti ere zerbitzariarekin konexioa ondo ezarrita egon, jokalariak bistaratuko dira.

Alineazioa ikusi

- *Fluxu arrunta:*

1. **Erabiltzaileak** alineazioa ikusi aukeratu
2. **Sistemak** erabiltzailearen alineazio tenporala lortu, ondoren jokalari bakoitzaren datu basikoak eskuratu eta hau erakutsiko du.

- *Kasu alternatiboak:*

- Erabiltzaileak ez du aurretik gordetako alineaziorik.

Alineazioa editatu

- *Fluxu arrunta:*

1. **Erabiltzaileak** alineazioa editatu aukeratu.
2. **Sistemak** erabiltzailearen jokalaria guztiak lortzen ditu eta bakoitzak datu basikoak lortu ondoren erabiltzaileari erakutsi.
3. **Erabiltzaileak** alineazioan dagoen jokalaria zaharra eta kanpo dagoen jokalaria berria aukeratzen ditu.
4. **Sistemak** jokalaria berria jarriko du alineazioan. Ondoren berriz 3. puntura pasako da.

- *Kasu alternatiboak:*

- Sorturiko alineazio berria posible ez izatea: jokalariren bat falta da.

Alineazioa gorde

- *Fluxu arrunta:*

1. **Erabiltzaileak** alineazioa aldatu ondoren gorde aukeratu.
2. **Sistemak** momentu horretan alineazio tenporal bezala gordeta dagoena datu basean gorde.

- *Kasu alternatiboak:*

- Sorturiko alineazio berria posible ez izatea: jokalariren bat falta da.

Alineazioa publikoki gorde

- *Fluxu arrunta:*

1. **Erabiltzaileak** alineazioa publikoki publikatu edo gordetzea aukeratu.
2. **Sistemak** datu basetik lortuko du azken gorderiko alineazio tenporala, eta hau alineazio publiko bezala esleituko du. Ondoren, datu basean aldaketak gordeko dira.

- *Kasu alternatiboak:*

- Sorturiko alineazio berria posible ez izatea: jokalariren bat falta da.

Merkatua

Jokalari bat merkatuan jarri

- *Fluxu arrunta:*

1. **Erabiltzaileak** Jokalari bat eta hasierako prezioa izango den zenbaki bat sartu.
2. **Sistemak** *MerkatukoJokalaria* objektua sortuko du.

- *Kasu alternatiboak:*

- Jokalari hori aurretik merkatuan jarrita egotea.
- Jarritako hasierako prezioa zero edo negatiboa izatea.

Merkatua kontsultatu

- *Fluxu arrunta:*
 1. **Erabiltzaileak** merkatura kontsultatu aukeratu.
 2. **Sistemak** komunitate horretako, MerkatukoJokalaria objektu guztiak lortu non onartua=*false* den. Hauek bistaratu.

- Administratzaile orokorrak egunero eguneratuko ditu merkatuak, eta modu horretan beti egongo dira jokalaria merkatuan, *kasu alternatiboak* saihestuz.

Merkatuko jokalaria bati oferta bat egitea

- *Fluxu arrunta:*
 1. **Erabiltzaileak** MerkatukoJokalaria eta diru kantitate bat sartu.
 2. **Sistemak** diru kantitate hori eskaintza handiena baino handiagoa dela konprobatu. Horrela bada eskaintza handienaren jabea bezala erabiltzailea jarri eta gordeko du.

- *Kasu alternatiboak:*
 - Erabiltzaileak duen diru kantitatea baino eskaintza balio handiagoa egitea.

Erabiltzaileak egindako eskaintzak kontsultatu

- *Fluxu arrunta:*
 1. **Erabiltzaileak** egindako eskaintzak kontsultatu.
 2. **Sistemak** erabiltzaile horrek egin dituen eskaintza guztiak lortu eta pantailan erakutsi.

- *Kasu alternatiboak:*

- Erabiltzaileak oraindik eskaintzarik ez egin izana.

Erabiltzaileak jasotako eskaintzak kontsultatu

- *Fluxu arrunta:*

1. **Erabiltzaileak** jasotako eskaintzak kontsultatu.
2. **Sistemak** erabiltzaile horrek jaso dituen eskaintza guztiak lortu eta pantailan erakutsi.

- *Kasu alternatiboak:*

- Erabiltzaileak oraindik eskaintzarik jaso ez izana.

Erabiltzaileak jasotako eskaintzak erantzun

- *Fluxu arrunta:*

1. **Erabiltzaileak** jasotako eskaintzak kontsultatu.
2. **Sistemak** erabiltzaile horrek jaso dituen eskaintza guztiak lortu eta pantailan erakutsi.
3. **Erabiltzaileak** jasotako eskaintzeri erantzuna eman.
4. **Sistemak** erantzunak jaso eta eskaintzeri dagokion erantzuna ezarri.

- *Kasu alternatiboak:*

- Erabiltzaileak oraindik eskaintzarik jaso ez izana.

Estadistikak

Jokalari guztien puntuak

- *Fluxu arrunta:*
 1. **Erabiltzaileak** jokalaria guztien puntuak aukeratu.
 2. **Sistemak** jokalaria bakoitzaren puntu totalak lortu eta ordenaturik inprimatu.

- *Kasu alternatiboak:*
 - Oraindik jardunaldirik ez jolastu izana eta guztiek 0 puntu izatea.

Jokalari baten puntuaketen eboluzioa

- *Fluxu arrunta:*
 1. **Erabiltzaileak** jokalaria bat bidali.
 2. **Sistemak** Jokalari objektu hori lortu eta dagozkion Jokalariaren puntuazioak erakutsi.

- *Kasu alternatiboak:*
 - Oraindik jardunaldirik ez jolastu izana, eta ondorioz puntuen eboluziorik ez izatea.
 - Jardunaldiak ezarrita ez egotea.

Finantzak

Transakzioen historikoa

- *Fluxu arrunta:*
 1. **Erabiltzaileak** aukeratu.
 2. **Sistemak** erabiltzaile horren Transakzio objektu guztiak lortu eta modu kronologikoan bistaratu.

- Erabiltzaile sortu orduko, sistema ongiatorri bonusa esleituko dionez, beti egongo da transakzioaren bat. Beraz, ez da posible izango *kasu alternatiborik* egotea.

Komunitatea

Mezuak irakurri

- *Fluxu arrunta:*
 1. **Erabiltzaileak** aukeratu.
 2. **Sistemak** Komunitateko Mezua objektu guztiak lortu eta egunaren arabera erakutsi

- *Kasu alternatiboak:*
 - Komunitatean mezurik ez egotea (oso gauza zaila, administratzaile orokorrak mezuak bidaltzen dituelako).

Mezuak idatzi

- *Fluxu arrunta:*

1. **Erabiltzaileak** mezu bat (String) bat bidali.
2. **Sistemak** Mezua izeneko objektu bat sortu eta Komunitateko mezuak-en gehitu.

- *Kasu alternatiboak:*

- Jarritako mezuak karaktere desegokiak edukitzea.
- Mezuak lengoaia ez-egokia erabiltzea.

Klasifikazioa

Jardunaldiko klasifikazioa

- *Fluxu arrunta:*
 1. **Erabiltzaileak** aukeratu.
 2. **Sistemak** data hori hartu eta dagokion Jardunaldia objektua lortu. Horrekin erabiltzaile horren komunitateko Erabiltzaile guztien Puntuazioa lortu eta hauek ordenaturik erakutsi.

- *Kasu alternatiboak:*
 - Oraindik jardunaldirik ez jolastu izana, eta ondorioz erabiltzaileek punturik ez izatea.
 - Jardunaldiak ezarrita ez egotea.

Klasifikazio generala

- *Fluxu arrunta:*
 1. **Erabiltzaileak** aukeratu.
 2. **Sistemak** komunitateko erabiltzaile guztien Puntuazioa lortu eta guztien batuketara egin. Ondoren hau erakutsi.

- *Kasu alternatiboak:*
 - Oraindik jardunaldirik ez jolastu izana, eta ondorioz erabiltzaileek punturik ez izatea.

2.4.3 Logeatu gabeko edo logeaturiko erabiltzailea

Eguneko partiduen emaitzak

- *Fluxu arrunta:*
 1. **Erabiltzaileak** egun bat aukeratu.
 2. **Sistemak** egun horretako partiduen emaitzak inprimatu.
- *Kasu alternatiboak:*
 - Egun horretan partidurik jolastu ez izana.

Jokalarien datuak kontsultatu

- *Fluxu arrunta:*
 1. **Erabiltzaileak** jokalaria baten izenaren zati bat idatzi.
 2. **Sistemak** izenean zati hori duten jokalarien horren datu desberdinak erakutsi.
- *Kasu alternatiboak:*
 - Ez da existitzen izen hori duen jokalaririk.

Erabiltzaileen alineazio publikoak kontsultatu

- *Fluxu arrunta:*
 1. **Erabiltzaileak** aukeratu.
 2. **Sistemak** alineazio publiko guztiak lortu eta irudikatu.
- *Kasu alternatiboak:*
 - Alineazio publikorik ez egotea.

2.4.4 Komunitateko administratzailea

Sariak

- *Fluxu arrunta:*
 1. **Erabiltzaileak** aukeratu.
 2. **Sistemak** momentu horretan komunitateak dituen sariak irudikatu.
 3. **Erabiltzaileak** kantitate berriak jarri.
 4. **Sistemak** kantitate berri horiek ezarri Komunitateari.

- *Kasu alternatiboak:*
 - Sarien kantitateak onargarriak ez izatea (negatiboak, etab.).

Mezua ezabatu

- *Fluxu arrunta:*
 1. **Erabiltzaileak** Mezua objektua bidali.
 2. **Sistemak** mezu hori ezabagarria bada ezabatu. Ondo joan dela abisatu.

Erabiltzailea ezabatu

- *Fluxu arrunta:*
 1. **Erabiltzaileak** Erabiltzailea bidali.
 2. **Sistemak** objektu hori eta erlazionaturiko guztiak ezabatu.

2.4.5 Administratzaile globala

Alineazioak finkatu

- *Fluxu arrunta:*
 1. **Administratzaileak** alineazioak finkatzea aukeratu.
 2. **Sistemak** datu-basetik momentuko alineazio guztiak hartu eta finkatu bezala kopiatu.

Jardunaldiak hasieratu

- *Fluxu arrunta:*
 1. **Administratzaileak** hasierako data bat sartuko du.
 2. **Sistemak** data horretatik hurrengo ekainaren azken asteraino, asteko jardunaldi bat sortuko du.

Jokalari eta erabiltzaileei jardunaldiko puntuak esleitu

- *Fluxu arrunta:*
 1. **Administratzaileak** jardunaldiaren hasiera data sartuko du.
 2. **Sistemak** Jardunaldia objektua lortuko du hasiera data horri dagokiona. Ondoren egun horietako Partiduak eta JokalariarenPartidua osatu, eta jokalaria bakoitzari jardunaldi horretako JokalariarenPuntuazioa objektua osatuko du. Ondoren, erabiltzaile bakoitzak alineazioFinkatua-n dituen jokalariaren puntuak batu eta Puntuazioa objektua osatuko du.
- *Kasu alternatiboak:*
 - Jardunaldia oraindik jolastu ez izana.
 - Jardunaldi horretan partidurik ez egotea.

NBA-ko jokalarien informazioa eguneratu gure datu-basean

- *Fluxu arrunta:*

1. **Administratzaileak** jokalarien informazioa eguneratzea aukeratu.
2. **Sistemak** NBA-ko API-tik datuan lortu, hauek Jokalari objektuetan gorde eta datu basean gordeko ditu.

NBA-ko taldeen informazioa eguneratu gure datu-basean

- *Fluxu arrunta:*

1. **Administratzaileak** taldeen informazioa eguneratzea aukeratu.
2. **Sistemak** NBA-ko API-tik datuan lortu, hauek Taldeak objektuetan gorde eta datu basean gordeko ditu.

Komunitateei mezu generalak bidaltzeko aukera

- *Fluxu arrunta:*

1. **Administratzaileak** mezua sartuko du.
2. **Sistemak** admin nick-a duen Erabiltzaile objektua lortuko du. Ondoren, komunitate guztieri Mezua objektu bat gehituko die administratzailearen erabiltzailearen izenean.

- *Kasu alternatiboak:*

- Jarritako mezuak karaktere desegokiak edukitzea.
- Mezuak lengoaia ez-egokia erabiltzea.

Komunitateei merkatuak eguneratu

- *Fluxu arrunta:*

1. **Administratzaileak** komunitateei merkatuak eguneratzea aukeratu.
2. **Sistemak** komunitate guztiak lortu eta bakoitzaren merkatuan beharrezkoak diren aldaketak egin.

3. KAPITULUA

Diseinua

Atal honetan, izenak dioen bezala, aplikazioaren diseinua aztertuko dugu. Lehendabizi erabili ditzakegun teknologiaak aztertu eta ondoren aukeratu egingo ditugu.

Behin erabiliko ditugun tresnak argi eduki, beharrezkoa izango dugun datu-basearen diseinua egin beharko dugu.

Bukatzeko erabilpen-kasu bakoitzaren diseinua dugu, hau da, erabiliko dugun interfazea eta sekuentzia-diagrama. Azken ataza hori oso garrantzitsua izango da inplementazioan lana modu automatikoago batean egiteko.

3.1 Teknologia posibleak

Egin nahi dugun proiektua oso handia eta hainbat zati oso desberdin ditu. Horregatik, teknologia ugari erabili beharko ditugu. Posibilitateak ondo aztertzeko egokiena proiektuaren zatiak ondo desberdintzea izango da.

3.1.1 Zerbitzaria

Agian produktuaren zati garrantzitsuena. Zerbitzariak proiektuaren negozio-logika guztia kontrolatuko du, web-zerbitzua publikatu eta dagozkion deiak prozesatu. Atal hau garatzeko ditugun aukerak zenbatezinak dira. Gaur egun, ia edozein programazio lengoaiak eskaintzen digu zerbitzari bat garatzeko aukerak. Aukeratzen dugun lengoaiaren arabera, programazio ingurunea IDE-a ere aldatu beharko dugu.

3.1.2 Datu-Basea

Gordeko ditugun datuak biltegitzeko erabiliko dugun modua. Oso garrantzitsua izango da datu-basearen mota aukeratzea, izan ere aukera anitz ditugu:

- Datu-base hierarkikoak
- Sare datu-baseak
- Datu-base erlazionalak
- Datu-base transakzionalak
- Datu-base multidimentsionalak
- Objektuetara bideratutako datu-baseak
- Datu-base dokumentalak
- Datu-base deduktiboak edo datu-base logikoak

Aukeratzen dugun motaren arabera, dagokion kudeatzeko sistema eta kontsultak egiteko modua ere asko aldatuko da

3.1.3 Zerbitzaria eta Datu-Basearen arteko konexioa

Zerbitzariak NBA-ko API-az gain, guk sortutako Datu-Basearekin konexioa oso jarraia izango da. Beraz, oso garrantzitsua da hau azkarra izatea, baina aldi berean txukuna. Aukeratzen dugun datu-basearen arabera asko aldatuko da konexioa ezartzeko era.

3.1.4 Web-Bezeroa eta Zerbitzariaren konexioa

Zerbitzariari deiak egiteaz arduratuko dena. Horrez gain erantzunak jaso eta interpretatu beharko ditu.

Historikoki ohikoena PHP erabiltzea dena arren, gaur egun alternatiba ugari agertzen joan dira. Beraz kontuan hartu beharko ditugu Python, Ruby, Go etab. moduko lengoaiak erabakia hartu aurretik.

3.1.5 Web-Bezeroaren Interfazeak

Zerbitzaritik jasotako datuak interpretatu ondoren, hauek erabiltzailearentzako pantaila-ratu beharko dira.

Aurreko kasuan bezala, oso ohikoa da PHP eta HTML5 konbinatzea. Hala ere HTML5 ez denez programazio lengoaia bat, baizik maketazio-lengoaia, ia ia guztiekin konbinatzeko aukera eskaintzen digu. Gainera, itxura hobetuzko lortzeko hainbat liburutegi erabiltzeko aukera eskaintzen digu: Bootstrap, ZURB Foundation, Pure.CSS, Semantic UI, UIKit, Materialize...

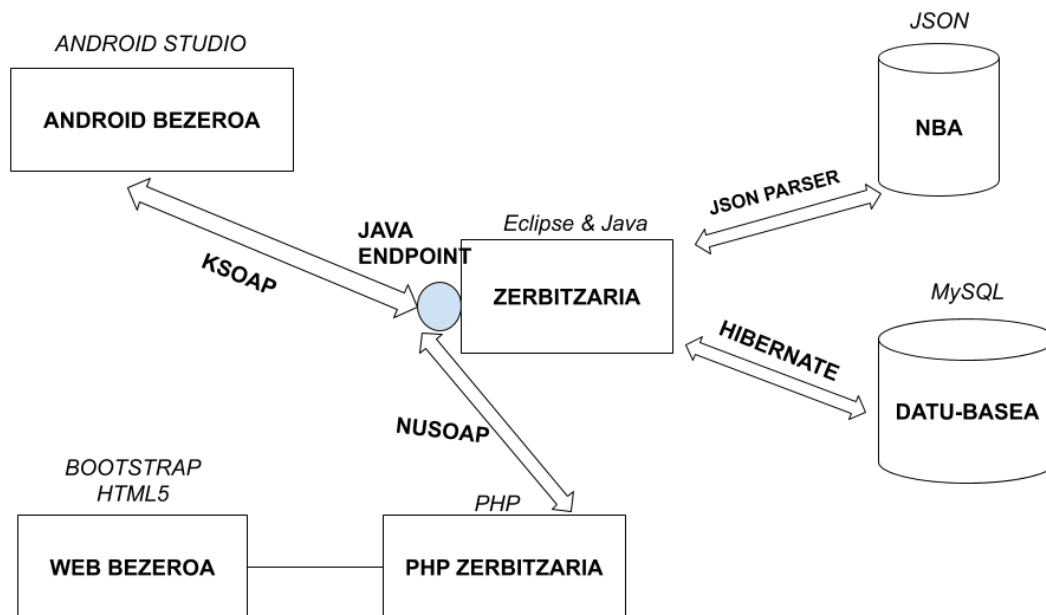
3.1.6 Android-Bezeroa, Zerbitzariaren Konexioa eta Interfazeak

Web-Bezeroarekin gertatzen den bezala, zerbitzariari deiak egin eta hauek prozesatzeaz arduratuko da. Baina kasu honetan, horretaz gain, datuak jaso eta interpretatu ondoren, hauei forma emateaz arduratuko da.

Kasu honetan guztia batera egiteko aukera dugu barne logika ezartzeko posibilitatea dugulako. Gainera, Android garapenerako IDE guztiek (edo ia guztiek) interfaze grafikoekin lan egiteko aukera eskaintzen digutenez, guztia leku bakar batean egin dezakegu. Hainbat aukera azaltzen dira: Android Studio (Google-ek sortua eta gaur egun merkatuko garrantzitsuenak), Xamarin, AIDE, B4A, Unity eta beste zenbait aukera gehiago.

3.2 Erabilitako tresna eta teknologiak

Ikerkuntza sakon bat egin ondoren, hauek izango dira proiektua aurrera eramateko aukeraturako teknologiak eta hemen ikusi dezakegu dagokion arkitektura diseinua.



3.1 Irudia: Erabilitako teknologien arkitektura.

3.2.1 Eclipse eta Java

Gradu guztian zehar gehien erabili izango dugun erreminta izan denez, ezagunena eta erosoena da. Atal honetan ez dugu zalantzarik izan. Gainera, negozio logika diseinatzerako orduan oso eroso egiten zaigu interfazeak sortzea eta hauek *Endpoint*-en bidez web-zerbitzuak sortzea.

Modu horretan zerbitzaria martxan izango genuke.

3.2.2 MySQL

Datu-base erlazionalak izan dira gehien erabili ditugunak gradu guztian zehar, eta apartatu honetan gehien gustatu zaidana beti *MySQL* izan da. Egia da gaur egun tendentzia aldatzen dagoela, bereziki Google-ek eskaintzen duen Firebase Realtime Database-arekin, non *NoSQL* motako datu basea eskaintzen du.

Hala ere, nahiago izan dugu MySQL erabiltzea.

3.2.3 Hibernate

Datu-basea nolakoa zen aukeratu ondoren, ez nuen zalantzarik izan Hibernaterekin. Erreminta honen bitartez Javako klaseen mapeaketak modu automatikoan egiteko aukera eskaintzen digu. Horrela guk soilik mapeaketa horiek sortzeaz eta konfiguratzea arduratu behar gara.

3.2.4 PHP eta Nusoap

Web-bezeroaren logikaz arduratuko den zatia. Bezeroak php deiak egingo dizkio, eta honen *Nusoap* liburutegiaren bidez zerbitzariarekin konexioa ezarriko du. Ondoren jasotako emaitzak prozesatu eta itzuliko ditu bezeroaren nabigatzaileak pantailaratzeko.

3.2.5 HTML5, JavaScript eta Bootstrap

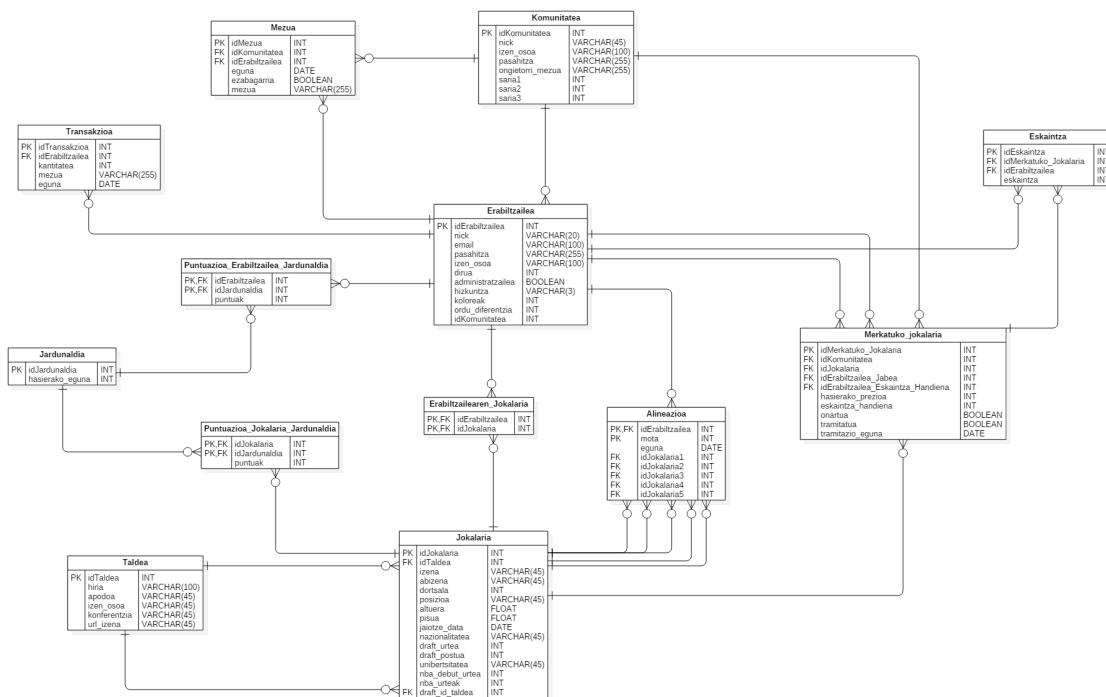
Datuak pantailaratzeko arduratuko da. Bootstrap-en bidez web-orriak *responsive* izateaz lortzeaz gain, itxura txukunagoa modu errazago batean ematea lortuko dugu. Hau guztia noski HTML5 maketazio lengoaiaren bidez egingo dugu. Atal batzuetan ere JavaScript kodeak txertatu beharko ditugu.

3.2.6 Android Studio eta Ksoap

Lehen azaldu dugun bezala, Android bezeroan atal guzti hau batera joango da. Android aplikazioak bere logika propioa izango du Java-z programatua, eta zerbitzariarekin kontaktuan jarriko da *Ksoap* liburutegia erabiliz. Aldi berean datuak pantailaratzeko Android Studiok eskaintzen dituen *widget*-ak erabiliko dira.

3.3 Datu-basearen diseinua

Domeinuaren Eredua kontuan harturik, hau izango litzateke berari dagokion datu-basearen diseinua. *Partidua* eta *Jokalariaren Partidua* klaseak ez dira datu-basean gordeko. Hauei dagokion informazioa oso aldakorra denez, hobe izango da datu horiek beharrezkoak direnean NBA-ko datu-basetik lortzea (API-aren bitartez).



3.2 Irudia: Datu-basearen diseinua.

3.4 Logeatu gabeko erabiltzailea

Logeatu gabeko erabiltzaileak eta logeaturikoak hainbat ekintza partekatu egiteko aukera izan arren, atal honetan berak soilik egin ditzakeen atazak aztertuko ditugu *Komunitatea sortu*, *Erabiltzailea sortu* eta *Kontuan sartu*.

3.4.1 Komunitatea sortu (eta administratzailea)

Komunitate berria sortzean, horrekin batera erabiltzaile kontu bat sortu beharra dago. Modu horretan, komunitate guztiek barne administratzaile bat dutela bermatzen dugu eta ez direla komunitate hutsak egongo.

The image shows two side-by-side screenshots of a mobile application interface for creating a community. The left screenshot, labeled (a) Web, shows a form titled 'Erregistratu' (Register) with the subtitle 'Komunitate berria sortu.' (Create new community). The form contains several input fields: 'Komunitateko nick-a' (Community nickname), 'Komunitateko izen-osoia' (Community name), 'Komunitateko pasahitza' (Community password), 'Komunitateko errepikatu' (Repeat community password), and 'Ongietorri mezua' (Welcome message). Below these are three fields for 'Saria 1', 'Saria 2', and 'Saria 3' with values 5000000, 2500000, and 1500000 respectively. A green button at the bottom is labeled 'Komunitatea sortu' (Create community). The right screenshot, labeled (b) Android, shows the same form on a mobile device. The title bar at the top says 'NBA SareSoziala'. The form fields are the same as in the web version, but the 'Saria' fields are pre-filled with the same values. A grey button at the bottom is labeled 'KOMUNITATEA SORTU'.

(a) Web

(b) Android

3.3 Irudia: Komunitatea sorturen interfazea.

The image shows two side-by-side screenshots of a mobile application interface for creating a user. The left screenshot, labeled (a) Web, shows a form titled 'Erregistratu' (Register) with the subtitle 'Erabiltzaile berria sortu.' (Create new user). The form contains several input fields: 'Erabiltzailearen nick-a' (User nickname), 'Erabiltzailearen izen-osoia' (User name), 'Erabiltzailearen email-a' (User email), 'Erabiltzailearen pasahitza' (User password), and 'Pasahitza errepikatu' (Repeat user password). A green button at the bottom is labeled 'Erabiltzailea sortu' (Create user). The right screenshot, labeled (b) Android, shows the same form on a mobile device. The title bar at the top says 'NBA SareSoziala'. The form fields are the same as in the web version. A grey button at the bottom is labeled 'ERABILTZAILA SORTU'.

(a) Web

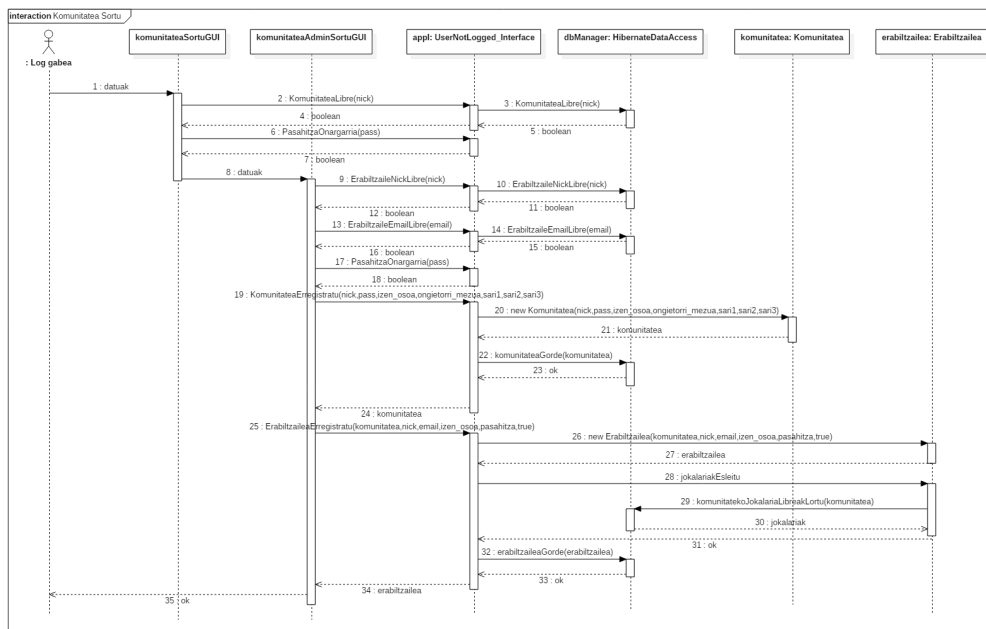
(b) Android

3.4 Irudia: Erabiltzailea sorturen interfazea.

Lehendabizi erabiltzaileak komunitate berria sortzeko beharrezkoak diren datuak sartuko ditu. Orduan *KomunitateaLibre(komunitateNick)* eta *PasahitzaOnargarria(komunitatePass)* funtzioen emaitzak baliozkoak badira, erabiltzailea sortzera pasako da.

Bertan, erabiltzailearen datuak sartuko ditu eta ondo dauden konprobatuko dira *ErabiltzaileNickLibre(nick)*, *ErabiltzaileEmailLibre(email)* eta *PasahitzaOnargarria(pass)* funtzioen bidez.

Dena ondo baldin badoa, komunitatea eta erabiltzailea sortuko dira, bata bestearen segidan *KomunitateaErregistratu(nick,pass,izenOsoa,ongietorriMezua,sari1,sari2,sari3)* eta *ErabiltzaileaErregistratu(komunitatea,nick,email,izenOsoa,pasahitza,true)*. Administratzailean *true* balioa joango da, hurrengo kasuetan ikusiko dugu hori ez dela horrela izango.



3.5 Irudia: Komunitatea sortu (eta administratzailea) sekuentzia-diagrama.

3.4.2 Kontua sortu

Erabiltzaile kontu berria sortzeko, lehendabizi komunitatea kokatu beharko dugu. Horretarako, erregistratu nahi duen komunitatearen *nick* eta *pasahitza* modu egokian jarri beharko ditu.

Hori ondo egin ondoren erabiltzailea sortzeko aukera izango du.

The web interface for joining a community features a green circular icon with a white square inside. Below the icon, the text "Komunitatea aukeratu" is displayed. There are two input fields: "Komunitateko nick-a" and "Komunitateko pasahitza". A green button labeled "Komunitatea aukeratu" is positioned at the bottom.

(a) Web

The Android interface for joining a community has a dark header with the text "NBA SareSoziala". Below the header is a red and black logo. The text "Komunitateko nick-a" is followed by a red underline. Below that is the text "Komunitateko pasahitza" followed by a grey underline. A grey button labeled "KOMUNITATEA AUKERATU" is at the bottom.

(b) Android

3.6 Irudia: Komunitatea aukeratu interfazea.

The web interface for user registration has a header with the text "Erregistratu" and the subtitle "Erabiltzaile berria sortu.". There are five input fields: "Erabiltzailearen nick-a", "Erabiltzailearen izen-osoa", "Erabiltzailearen email-a" (highlighted with a red border), "Erabiltzailearen pasahitza" (highlighted with a green border), and "Pasahitza errepikatu" (highlighted with a green border). A green button labeled "Erabiltzailea sortu" is at the bottom.

(a) Web

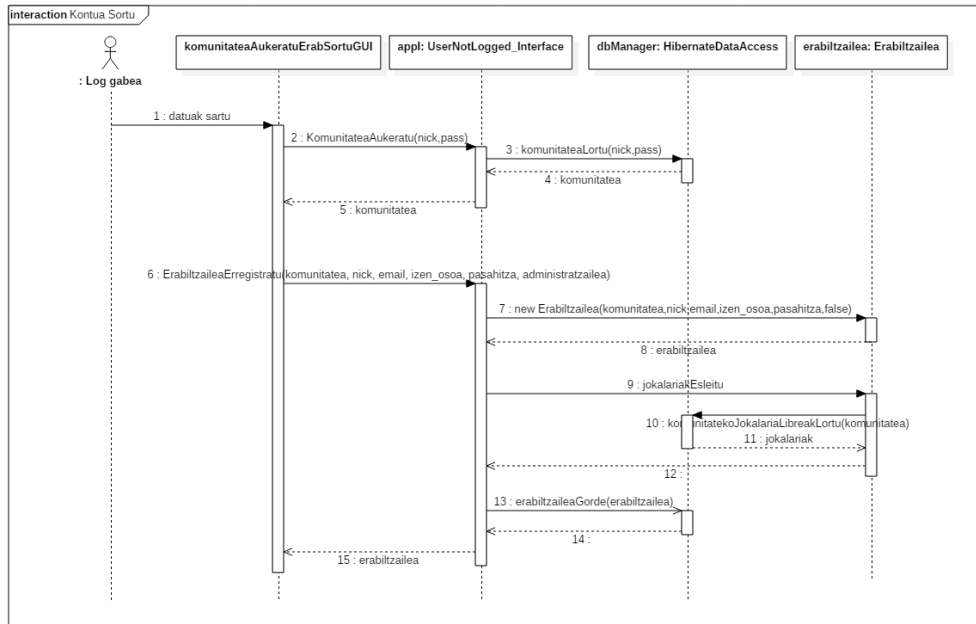
The Android interface for user registration has a dark header with the text "NBA SareSoziala". Below the header is a red and black logo. The text "Erabiltzaile nick-a" is followed by a red underline. Below that are input fields for "Izen osoa", "Email", "Pasahitza", and "Pasahitza errepikatu", each followed by a grey underline. A grey button labeled "ERABILTZAILEA SORTU" is at the bottom.

(b) Android

3.7 Irudia: Erabiltzailea sortu interfazea.

Erabiltzaileak *nick* eta *pasahitza* jarriko ditu eta honek zerbitzariari dei egingo dio *KomunitateaAukeratu* funtzioaren bidez. Egokia baldin bada hurrengo urratsera joango da, bestela errore mezu baten bidez bertan behera utziko du prozesua.

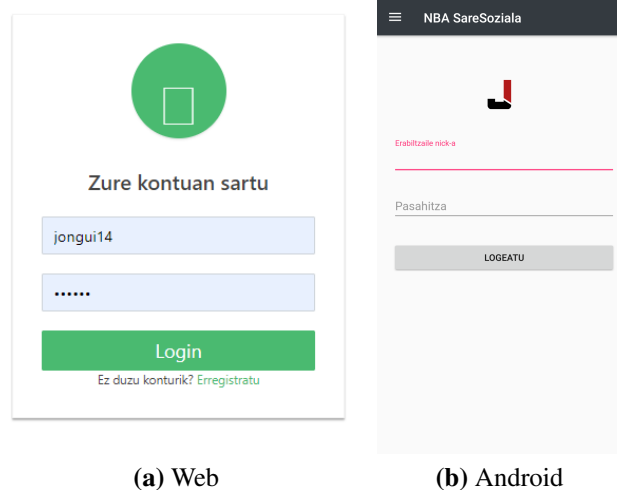
Dena ondo joan bada, erabiltzailea sortzeko interfaze berria azalduko da. Aurreko atalean bezala sortuko genuke erabiltzailea, baina kasu honetan zerbitzariari deitzean *administra-tzaile* balioa ezezkoa izango da.



3.8 Irudia: Kontua sortu sekuentzia-diagrama.

3.4.3 Kontura sartu

Erabiltzailea kontura sartzeko nick-a eta pasahitza modu egokian jarri beharko ditu. Behin kontura sartzerakoan, ikusiko dituen menuak desberdinak izango dira.



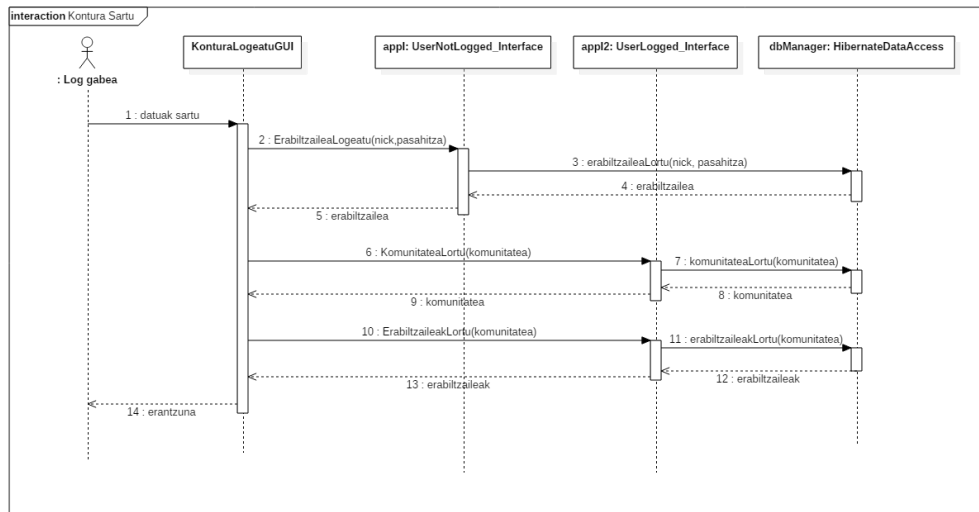
(a) Web

(b) Android

3.9 Irudia: Mezuak irakurri interfazea.

Erabiltzaileak *nick* eta *pasahitza* jarriko ditu eta honek zerbitzariari dei egingo dio *ErabiltzaileaLogeatu* funtzioaren bidez. Egokia baldin bada hurrengo urratsera joango da, bestela errore mezu baten bidez bertan behera utziko du prozesua.

Dena ondo joan baldin bada *KomunitateaLortu* eta *ErabiltzaileakLortu* funtzioen datuak jasoko dira askotan errepikatzen diren deiak direlako.



3.10 Irudia: Mezuak irakurri sekuentzia-diagrama.





















3.5 Logeatutako erabiltzailea









Logeatutako erabiltzaileak egiteko gauza asko dituenek, multzoka sailkatu ditut ekintzak. Multzo horiek erabiltzaileak menuan ikusiko dituen aukera izango dira: *Jokalariak*, *Merkatua*, *Estatistikak*, *Finantzak*, *Klasifikazioa* eta *Komunitatea*.

Jokalariak

3.5.1 Erabiltzailearen jokalarien datuak ikusi

Erabiltzaile bakoitzak jokalaria propio batzuk izango ditu, eta jokalaria horiek bakarrak izango dira komunitate horretan. Modu honetan, erabiltzaileak bere jokalarien inguruko informazio sakonagoa ikusteko aukera izango du.

| Jokalariak | Soldata | Taldea | Posizioa | Dorsala | Altuera | Pisua | Jaiotze-data | Nazionalitatea |
|--|----------|---|----------|---------|---------|-------|--------------|----------------|
|  Treveon Graham | 1512601 |  | F | 21 | 1.96 | 102.1 | 1993-10-28 | USA |
|  Tony Snell | 10607143 |  | G | 21 | 2.01 | 96.6 | 1991-11-10 | USA |
|  Damian Jones | 1512601 |  | C | 15 | 2.13 | 111.1 | 1995-06-30 | USA |
|  Zach LaVine | 19500000 |  | G | 8 | 1.96 | 90.7 | 1995-03-10 | USA |
|  Thon Maker | 2799720 |  | F | 7 | 2.16 | 100.2 | 1997-02-25 | South Sudan |
|  Kemba Walker | 12000000 |  | G | 15 | 1.85 | 83.5 | 1990-05-08 | USA |
|  Iman Shumpert | 11011234 |  | G | 1 | 1.96 | 97.5 | 1990-06-26 | USA |
|  Deonte Burton | 500000 |  | G | 30 | 1.96 | 111.1 | 1994-01-31 | USA |
|  Kyle Lowry | 31200000 |  | G | 7 | 1.85 | 88.9 | 1986-03-25 | USA |
|  Jabari Parker | 20000000 |  | F | 12 | 2.03 | 111.1 | 1995-03-15 | USA |

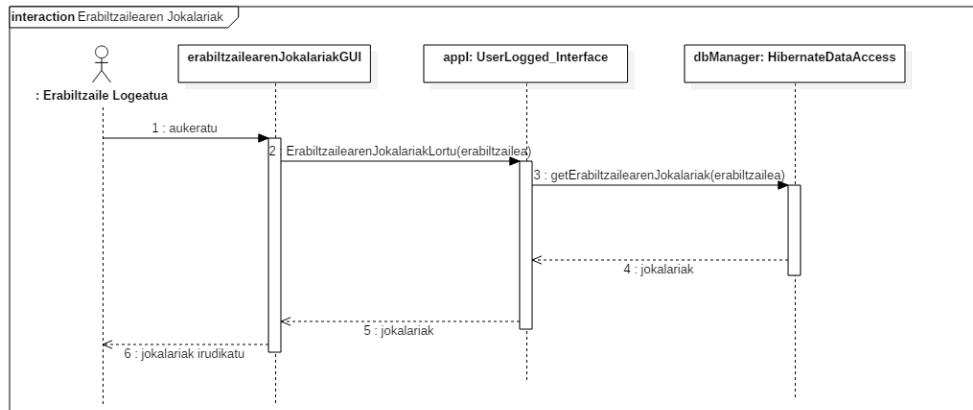
| NBA SareSoziala | |
|--|---|
|  Iman Shumpert | Altuera 1.96m, Jaiotze data 1990-06-26, Posizioa G, #1, Soldata \$11,234,011 |
|  Deonte Burton | Altuera 1.96m, Jaiotze data 1994-01-31, Posizioa G, #30, Soldata \$500,000 |
|  Jabari Parker | Altuera 2.03m, Jaiotze data 1995-03-15, Posizioa F, #12, Soldata \$20,000,000 |
|  Thon Maker | Altuera 2.16m, Jaiotze data 1997-02-25, Posizioa F, #7, Soldata \$2,720,799 |
|  Kemba Walker | Altuera 1.85m, Jaiotze data 1990-05-08, Posizioa G, #15, Soldata \$12,000,000 |
|  Tony Snell | Altuera 2.01m, Jaiotze data 1991-11-10, Posizioa G, #21, Soldata \$10,143,607 |
|  Treveon Graham | Altuera 1.96m, Jaiotze data 1993-10-28, Posizioa F, #21, Soldata \$1,601,512 |
|  Damian Jones | Altuera 2.13m, Jaiotze data 1995-06-30 |

(a) Web

(b) Android

3.11 Irudia: Erabiltzailearen jokalarien interfazea.

Prozesua oso sinplea da, erabiltzaileak *Nire Jokalariak* atala aukeratzen duenean, zerbitzariari dei egiten zaio *ErabiltzailearenJokalariakLortu* erabiltzailea objektua bera bidalitzeko. Bertan, datu basetik lortuko dira erabiltzaile horrek dituen jokalaria guztiak eta bezeroari bidaliko zaizkie. Bezeroak ondoren soilik hauek pantailaratu beharko ditu.



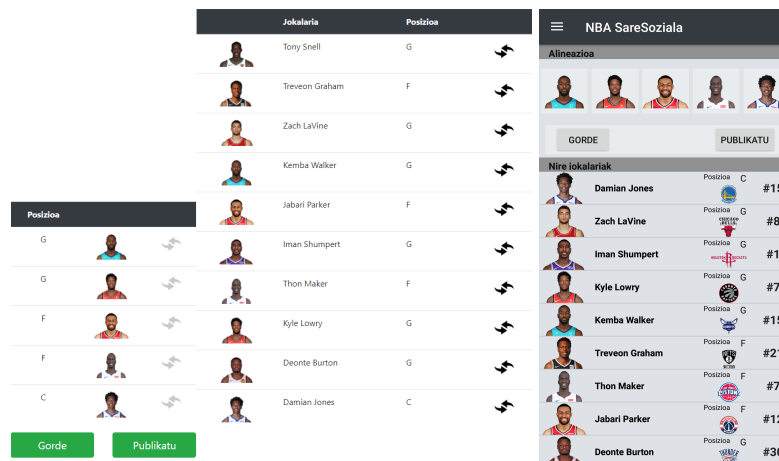
3.12 Irudia: Erabiltzailearen jokalariren sekuentzia-diagrama.

3.5.2 Alineazioa ikusi, editatu, gorde eta publikatu

Nahiz eta hainbat erabilpen-kasu independente izan, guztiak interfaze berdinetik egiteko aukera dugunez, guztiak aldi berean aztertuko ditugu.

Alde batetik alineazioa dugu eta bestalde erabiltzailearen jokalariak. Jokalari bakoitzak postu konkretu batzuk bakarrik eduki ditzake.

Ondoren, gordetzeko eta publikatzeko aukera izango dugu.

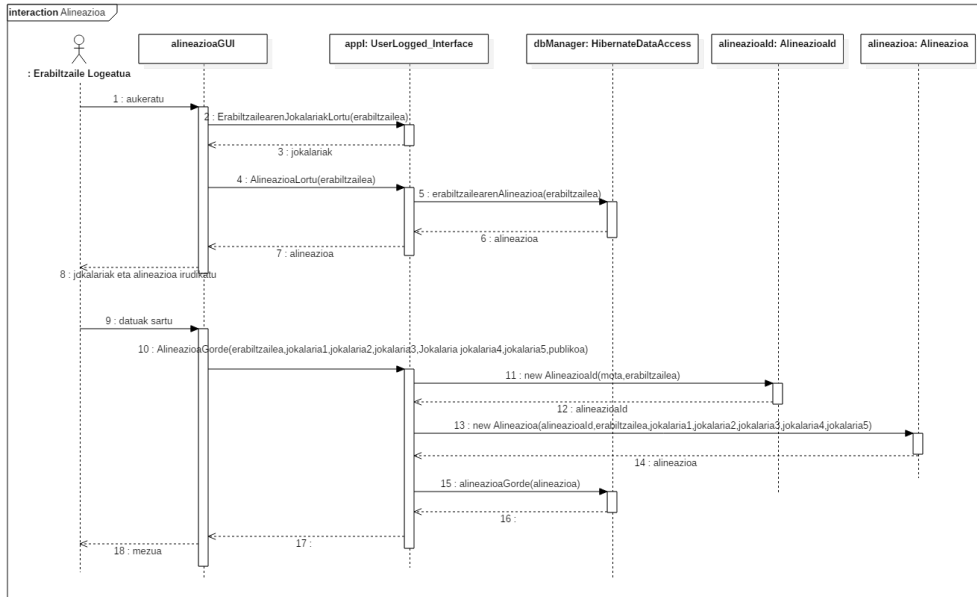


(a) Web

(b) Android

3.13 Irudia: Alineazioaren interfazea.

Interfazea kargatzerakoan, erabiltzailearen jokalaria eta alineazioa lortu beharko ditugu, *ErabiltzailearenJokalariaLortu* eta *AlineazioaLortu* funtzioen bidez, erabiltzailea parametro bezala pasaz.



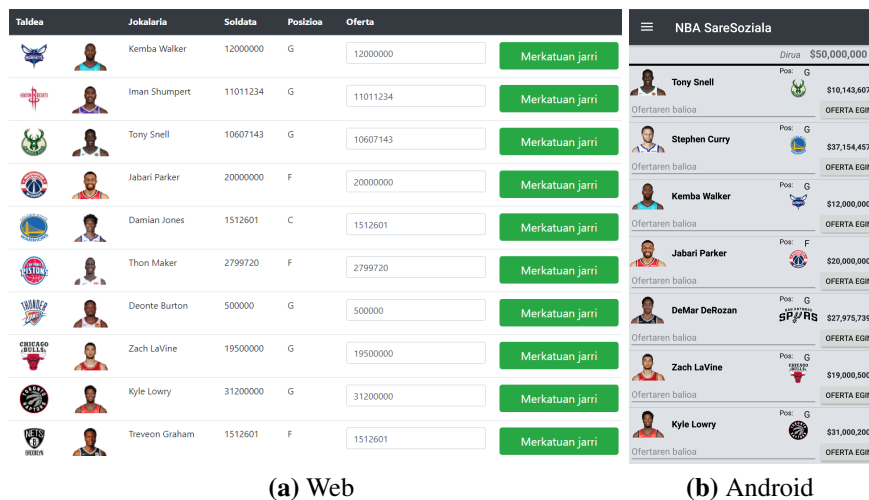
3.14 Irudia: Alineazioa ikusi, editatu, gorde eta publikaturen sekuentzia-diagrama.

Merkatua

3.5.3 Jokalari bat merkatuan jarri

Erabiltzaileak bere jokalaria saltzea nahi baldin badu, hauek merkatuan jarri beharko ditu. Horretarako hasierako prezio minimo baten bidez, hauek merkatuan eskuragarri jartzeko aukera izango du.

Ondoren, merkatuko jokalaria bakoitzak gainontzeko erabiltzaileen eskaintzak jasotzeko aukera izango du, baita "root"erabiltzailearena ere. Modu horretan erabiltzaileari beti aukera bat eskainiko zaio. *Root* honen eskaintza merkatuan jartzerakoan gertatuko da, bere soldataren %95-a eskainiz.

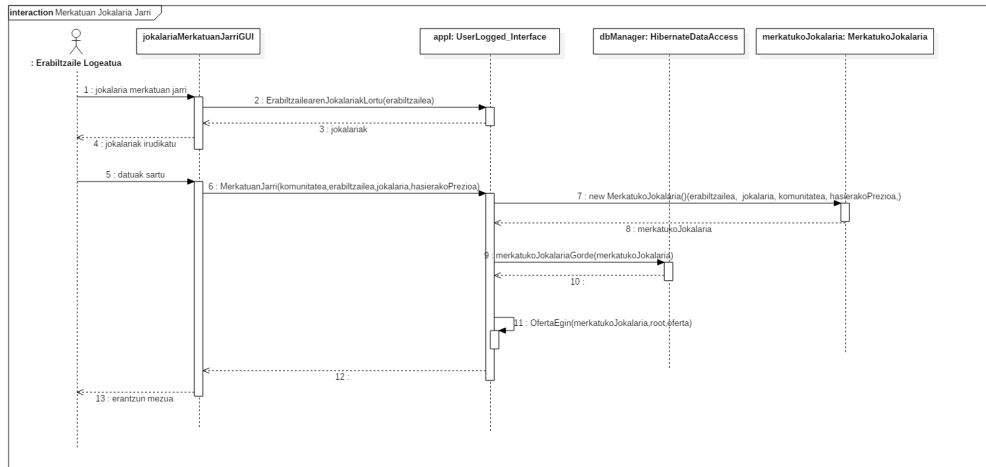


3.15 Irudia: Jokalari bat merkatuan jarriren interfazea.

Lehendabizi, erabiltzaileak dituen jokalariai lortu behar ditugu (aurrean azaldu dugu zelan), baina kasu honetan bezeroaren interfazea desberdina izango da. Ondoren merkatuan jarri nahi dugun jokalariai hasierako prezio bat jarri, merkatuan jartzeko aukera izango dugu *MerkatuanJarri* funtzioa erabiliz. Bertan komunitatea, erabiltzailea, jokalaria eta hasierako-prezioa bidaliko ditugu parametro gisa.

Hasteko, komunitate horretako merkatuan dagoen konprobatu beharko dugu, eta horrela baldin bada, bukatutzat emango dugu prozesua. Gero *MerkatukoJokalaria* objektu berria sortuko da pasatako datuekin eta hori datu-basean gorde. Gainera, eskaintza bat egin behar diogunez, *OfertaEgin* funtzioari deituko diogu (ondoren ikusiko dugu honen egitura).

Modu horretan jokalaria merkatuan jarrita egongo da eta erabiltzaileak gutxienez eskaintza bat jasako du.



3.16 Irudia: Jokalari bat merkatuan jarriren sekuentzia-diagrama.

3.5.4 Merkatua kontsultatu

Erabiltzaile bakoitza komunitate baten parte izango da, eta komunitate bakoitzak bere merkatu propioa izango du. Merkatu honetan jokalaria egongo dira hasierako prezio batekin. Edozein erabiltzaileek egin dezake merkatuan dagoen jokalari bati eskaintza bat.

| Jokalaria | Taldea | Soldata | Posizioa | Has. Prezioa | Eskaintza | |
|-----------|--------|----------|----------|--------------|----------------------|-------------|
| | | 10607143 | G | 10607143 | <input type="text"/> | Oferta egin |
| | | 37457154 | G | 37457154 | <input type="text"/> | Oferta egin |
| | | 12000000 | G | 12000000 | <input type="text"/> | Oferta egin |
| | | 20000000 | F | 20000000 | <input type="text"/> | Oferta egin |
| | | 27739975 | G | 27739975 | <input type="text"/> | Oferta egin |
| | | 19500000 | G | 19500000 | <input type="text"/> | Oferta egin |
| | | 31200000 | G | 31200000 | <input type="text"/> | Oferta egin |
| | | 11011234 | G | 11011234 | <input type="text"/> | Oferta egin |

(a) Web

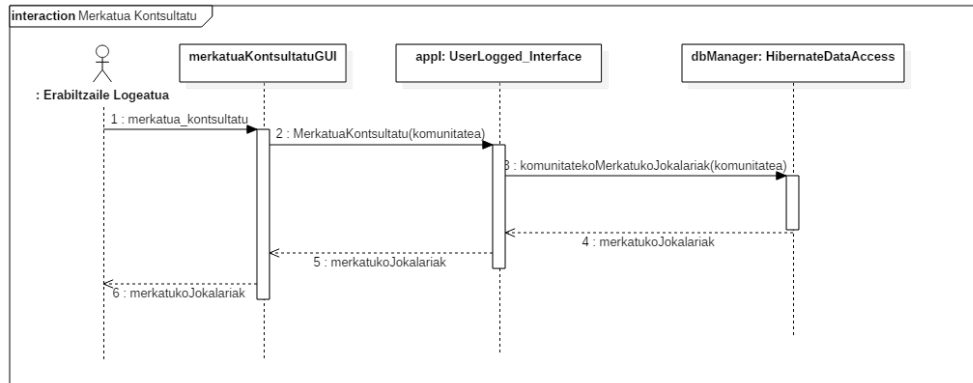
| NBA SareSoziala | |
|------------------|---------------------|
| | Dirus \$50,000,000 |
| Tony Snell | Pos: G \$10,143,607 |
| Ofertaren balioa | OFERTA EGIN |
| | Pos: G \$37,154,457 |
| Ofertaren balioa | OFERTA EGIN |
| | Pos: G \$12,000,000 |
| Ofertaren balioa | OFERTA EGIN |
| | Pos: F \$20,000,000 |
| Ofertaren balioa | OFERTA EGIN |
| | Pos: G \$27,975,739 |
| Ofertaren balioa | OFERTA EGIN |
| | Pos: G \$19,000,500 |
| Ofertaren balioa | OFERTA EGIN |
| | Pos: G \$31,000,200 |
| Ofertaren balioa | OFERTA EGIN |

(b) Android

3.17 Irudia: Merkatua kontsultatu interfazea.

Erabiltzaileak *Merkatua kontsultatu* atala aukeratzen duenean, zerbitzariari dei egiten zaio *MerkatuaKontsultatu* erabiltzailearen komunitatea objektua bidaliz. Bertan, datu basetik lortuko dira komunitate horrek une horretan merkatuan dituen jokalariren lista eta hau be-

zeroari pasako zaio. Ondoren bezero bakoitzak, bere moduan, datu horiek pantailaratuko ditu.



3.18 Irudia: Merkatua kontsultatu sekuentzia-diagrama.

3.5.5 Merkatuko jokalari bati eskaintza bat egitea

Aurreko erabilpen kasuaren garapena. Merkatua kontsultatu ondoren jokalariengatik eskaintza bat egiteko aukera dago.

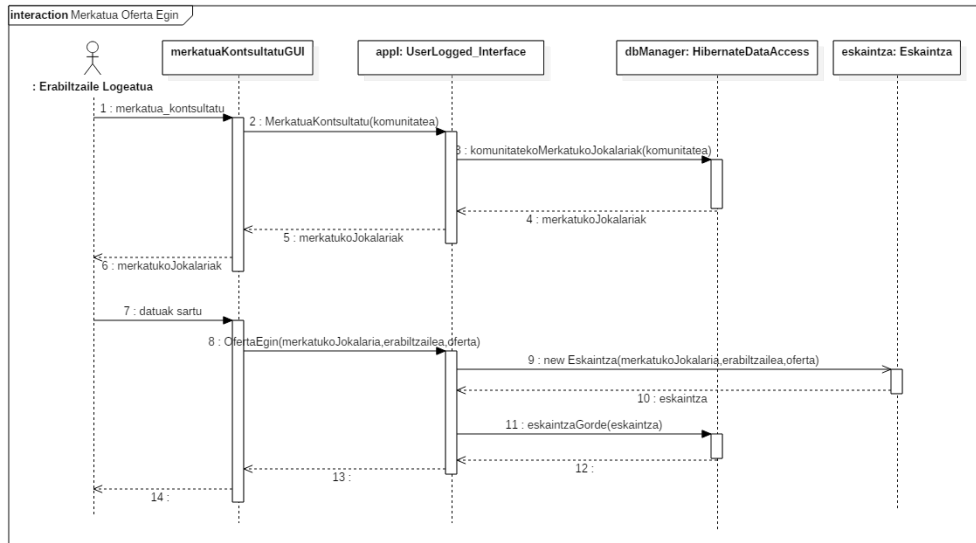
| Jokalaria | Taldea | Soldata | Posizioa | Has. Prezioa | Eskaintza | |
|-----------|--------|----------|----------|--------------|--------------------------------------|-------------|
| | | 10607143 | G | 10607143 | <input type="text"/> | Oferta egin |
| | | 37457154 | G | 37457154 | <input type="text"/> | Oferta egin |
| | | 12000000 | G | 12000000 | <input type="text"/> | Oferta egin |
| | | 20000000 | F | 20000000 | <input type="text"/> | Oferta egin |
| | | 27739975 | G | 27739975 | <input type="text" value="3000000"/> | Oferta egin |
| | | 19500000 | G | 19500000 | <input type="text"/> | Oferta egin |
| | | 31200000 | G | 31200000 | <input type="text"/> | Oferta egin |
| | | 11011234 | G | 11011234 | <input type="text"/> | Oferta egin |
| | | 12000000 | G | 20000000 | <input type="text"/> | Oferta egin |
| | | 6000000 | F | 8000000 | <input type="text"/> | Oferta egin |

(a) Web

(b) Android

3.19 Irudia: Merkatuko jokalari bati eskaintza egitearen interfazea.

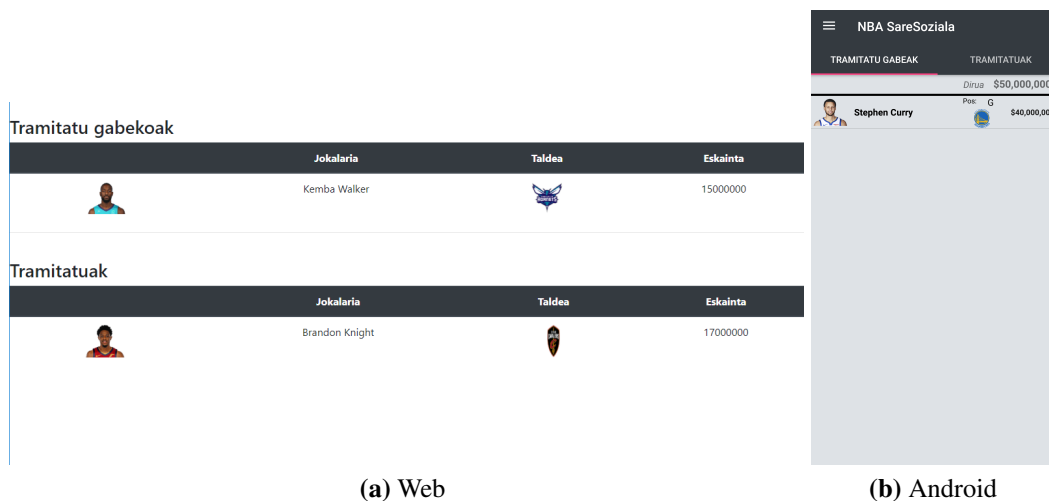
Merkatuan dauden jokalari bat aukeratu, eta diru kantitate bat ezarri ondoren, zerbitzarian eskaintza objektu berri bat sortuko da eta ondoren hori datu-basean gorde. Horretarako, zerbitzariaren *OfertaEgin* funtzioa erabiliko da.



3.20 Irudia: Merkatuko jokalaria bati eskaintza egitearen sekuentzia-diagrama.

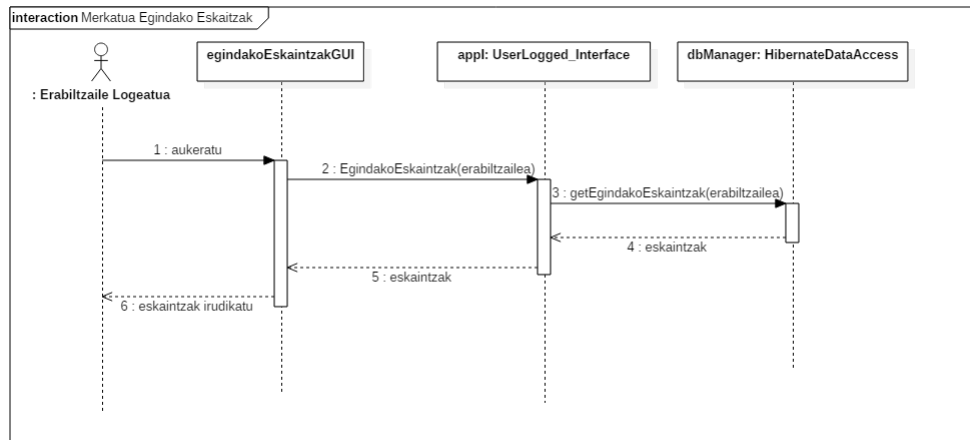
3.5.6 Erabiltzaileak egindako eskaintzak kontsultatu

Oso ohikoa izaten da egindako eskaintzen historiko bat izatea horrelako aplikazioetan.



3.21 Irudia: Erabiltzaileak egindako eskaintzak kontsultaturen interfazea.

Hau lortzeko, bezeroak *EgindakoEskaintzak* funtzioaren bidez erabiltzailea objektua bidaliko du. Bertan, egin dituen eskaintza guztiak lortu eta itzuliko ditu.



3.22 Irudia: Erabiltzaileak egindako eskaintzak kontsultaren sekuentzia-diagrama.

3.5.7 Erabiltzaileak jasotako eskaintzak kontsultatu, onartu edo ezeztatu

Guztia interfaze berdinetik egiteko aukera dago. Merkatuko jokalaririk bakoitzak jaso dituen eskaintzen listatu bat egongo da.

Horietako bat onartzea nahi baldin badugu, hori aukeratu eta *Onartu* botoiaren bidez egingo dugu.

Aldiz, onartu ondoren damututa sentitzen baldin bagara, *Ezeztatzeko* aukera existitzen da.

Tramitatu gabekoak

| Jokalaria | Taldea | Eskatariko balioa | Eskaintzak |
|------------------|--------|-------------------|--|
| Jonathon Simmons | | 8000000 | *5700000 5700000 <input type="button" value="Onartu"/> <input type="button" value="Ezeztatu"/> |

Tramitatuak

| Jokalaria | Taldea | Eskatariko balioa | Eskaintzak |
|---------------|--------|-------------------|------------|
| Anthony Davis | | 18787123 | 18787123 |
| Serge Ibaka | | 20583333 | *20583333 |
| Thon Maker | | 0 | 0 |

NBA SareSoziala

TRAMITATU GABEAK TRAMITATUAK

Dirua: \$50,000,000

Kemba Walker Pos: G

\$11,000,400 \$20,000,000

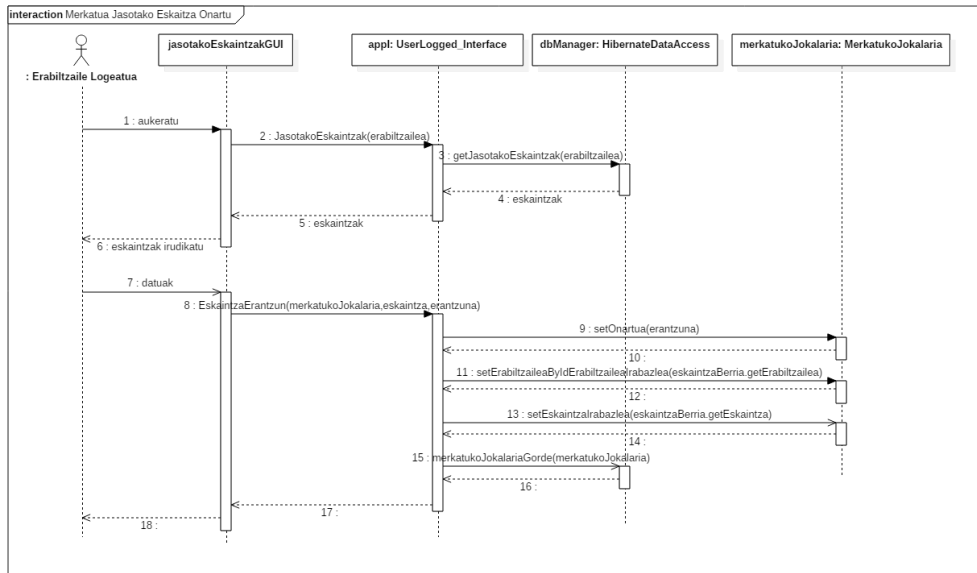
\$11,000,400

(a) Web
(b) Android

3.23 Irudia: Erabiltzaileak jasotako eskaintzak kontsultatu eta onarturen interfazea.

Lehendabizi erabiltzaileak jasotako eskaintza guztiak lortu behar ditugu eta hauek pantailan jarri.

Ondoren erabiltzaileak erantzun bat emateko aukera izango du. Positiboa den kasuan eskaintza onartzeko izango da, aldiz negatiboa den kasuan, eskaintza ezeztatu eta ondorioz irabazlea ezabatu.





3.24 Irudia: Erabiltzaileak jasotako eskaintzak kontsultatu eta onarturen sekuentzia-diagrama.





















Estadistikak

3.5.8 Jokalari baten puntuen eboluzioa

Jokalari bakoitzari jardunaldi puntuazio bat ezarriko zaio. Jokalari baten azterketa sakonagoa egitea nahi baldin bada, jardunaldi bakoitzean lorturiko puntuen kontsulta egiteko aukera dago.

| Jokalari | Soldata | Taldea | Posizioa | Dorsala | Altuera | Pisua | Jaiotza-data | Nazionalitatea |
|--|--------------|----------|--|---------|---------|-------|--------------|----------------|
|  | Kemba Walker | 12000000 |  | | | | | |
| Jardunaldia | Puntuak | | | | | | | |
| 32 | 0 | | | | | | | |
| 31 | 0 | | | | | | | |
| 30 | 0 | | | | | | | |
| 29 | 0 | | | | | | | |
| 28 | 0 | | | | | | | |
| 27 | 0 | | | | | | | |
| 26 | 0 | | | | | | | |
| 25 | 0 | | | | | | | |
| 24 | 0 | | | | | | | |
| 23 | 0 | | | | | | | |
| 22 | 0 | | | | | | | |
| 21 | 0 | | | | | | | |
| 20 | 0 | | | | | | | |

(a) Web

| Jokalari | Taldea | Posizioa | Puntuak | Reboundak |
|---|--|----------|---------|-----------|
|  |  | G | 12 | 0 |
|  |  | G | 11 | 0 |
|  |  | G | 10 | 0 |
|  |  | F | 9 | 0 |
|  |  | G | 8 | 0 |
|  |  | G | 7 | 0 |
|  |  | G | 6 | 0 |
|  |  | F | 5 | 0 |
|  |  | F | 4 | 0 |
|  |  | C | 3 | 0 |
| | | | 2 | 0 |
| | | | 1 | 18 |

(b) Android

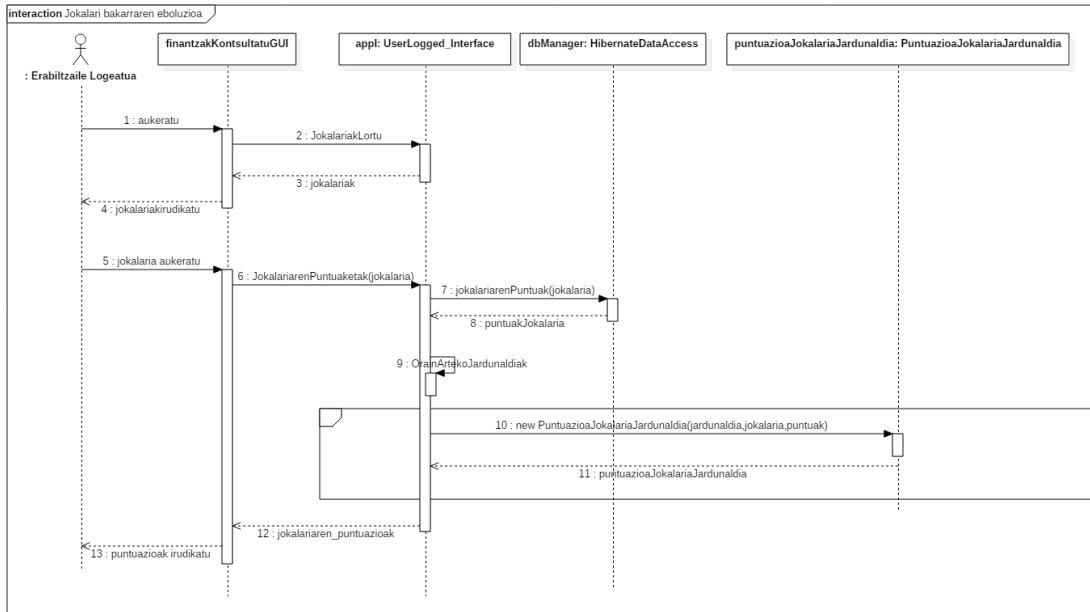
3.25 Irudia: Jokalari baten puntuen eboluzioaren interfazea.

Lehendabizi erabiltzailearen jokalaria lortuko ditugu. Ondoren horietako edozein aukeratzeko aukera emango zaio erabiltzaileari, baina bezeroak listako lehenengoa hartuko du defektuz bat kargatzeko.

Ondoren datu-basetik jokalari horren puntuazio guztiak jasoko ditugu. Hala ere, posible da jardunaldi batean jolastu ez izana, eta orduan aste horretako puntuazioa 0-koa izan beharko da.

Horretarako orain arteko jardunaldi guztiak lortuko dira eta bakoitzeko jasotako puntuak esleituko ditugu.

Azkenik bezeroak emaitza jaso eta irudikatuko du.



3.26 Irudia: Jokalari baten puntuaren eboluzioaren sekuentzia-diagrama.

3.5.9 Jokalari guztien puntuak

Aurreko kasuan jardunaldika desglosatzen bagenuen jokalari bakoitzaren puntuak, kasu honetan puntuazio orokorra bakarrik kontuan hartzen dugu. Modu horretan, ligako jokalaria guztien arteko konparaketa egiteko modurik sinpleena izango da.

| Jokalari | Taldea | Puntuak |
|---------------------|--------|---------|
| Nikola Jokic | | 58 |
| Anthony Davis | | 47 |
| Jimmy Butler | | 45 |
| Nikola Vucevic | | 45 |
| Blake Griffin | | 35 |
| Garrett Temple | | 32 |
| Kyle Lowry | | 30 |
| Steven Adams | | 29 |
| Nikola Mirotic | | 29 |
| Chris Paul | | 28 |
| Willie Cauley-Stein | | 27 |

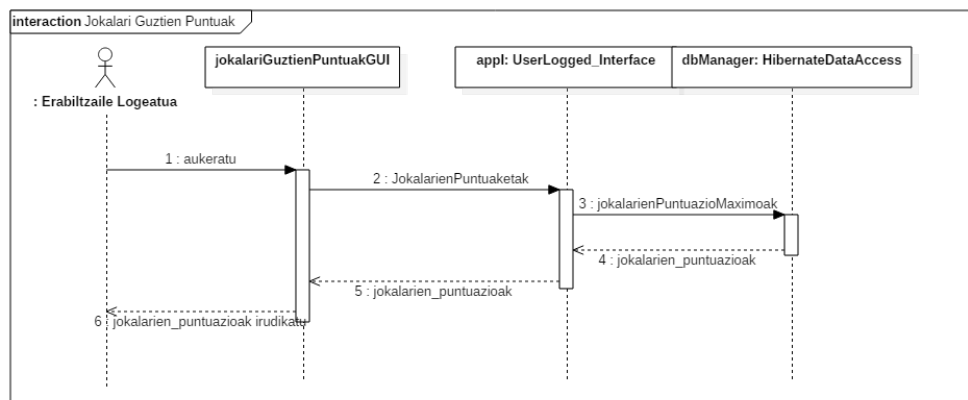
| NBA SareSoziala | | |
|---------------------|--|----|
| Nikola Jokic | | 58 |
| Anthony Davis | | 47 |
| Jimmy Butler | | 45 |
| Nikola Vucevic | | 45 |
| Blake Griffin | | 35 |
| Garrett Temple | | 32 |
| Kyle Lowry | | 30 |
| Steven Adams | | 29 |
| Nikola Mirotic | | 29 |
| Chris Paul | | 28 |
| Willie Cauley-Stein | | 27 |
| Joel Embiid | | 27 |
| Jayson Tatum | | 26 |

(a) Web

(b) Android

3.27 Irudia: Jokalari guztien puntuaren interfazea.

Sekuentzia-diagrama hau oso sinplea da. Datu-baseari dei egiten diogu jokalari guztien puntuazio maximoak lortuz. Ondoren negozio logikak hauek ordenatuko ditu puntuen arabera eta bezeroari pasako dizkio honek pantailaratzuz. Ez da beharrezkoa izango inolako parametririk pasatzea funtzioen artean.



















3.28 Irudia: Jokalari guztien puntuen sekuentzia-diagrama.

Finantzak

3.5.10 Transakzioen historikoa

Erabiltzaileek bere diruarekin egin diren transakzioak ikusteko ohitura handia dago. Batez ere, mugimenduak ondo joan direla ikusteko, jardunaldiko saria ondo sartu dela etab. Horretarako finantzak atala dugu transakzio guztiekin.

| Jokalaria | Taldea | Soldata | Posizioa | Has. Prezioa | Eskaintza | |
|---|---|----------|----------|--------------|----------------------|-------------|
|  |  | 10607143 | G | 10607143 | <input type="text"/> | Oferta egin |
|  |  | 37457154 | G | 37457154 | <input type="text"/> | Oferta egin |
|  |  | 12000000 | G | 12000000 | <input type="text"/> | Oferta egin |
|  |  | 20000000 | F | 20000000 | <input type="text"/> | Oferta egin |
|  |  | 27739975 | G | 27739975 | <input type="text"/> | Oferta egin |
|  |  | 19500000 | G | 19500000 | <input type="text"/> | Oferta egin |
|  |  | 31200000 | G | 31200000 | <input type="text"/> | Oferta egin |
|  |  | 11011234 | G | 11011234 | <input type="text"/> | Oferta egin |

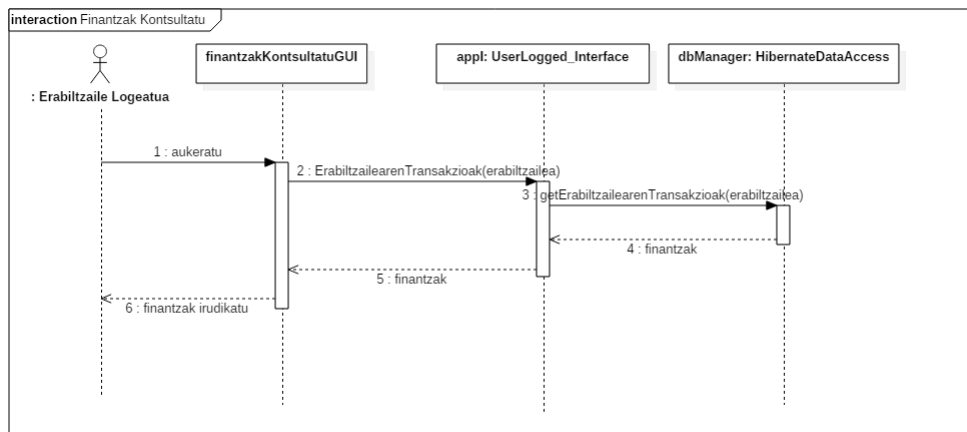
(a) Web

| NBA SareSoziala | |
|-----------------|---|
| Dirua | \$50,000,000 |
| \$40,000,000 | Ongietorri bonusa 2019-4-16 |
| \$5,000,000 | Jardunaldian 1. postua lortzeagatik 2019-4-16 |
| \$5,000,000 | Jardunaldian 1. postua lortzeagatik 2019-4-16 |

(b) Android

3.29 Irudia: Transakzioen historikoen interfazea.

Erabiltzaileak *Finantzak* atala aukeratzen duenean, zerbitzariari dei egiten zaio *Erabiltzailearen Transakzioak* parametro gisa erabiltzailea objektua bidaliz. Bertan, datu basetik lortuko dira erabiltzaileak egin dituen transakzio guztiak eta ondoren bezeroak dagokion forma emango dio.

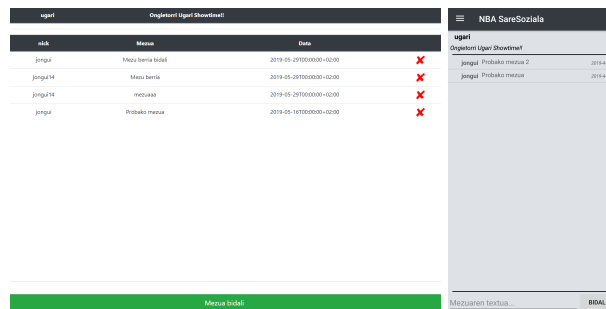


3.30 Irudia: Transakzioen historikoen sekuentzia-diagrama.

Komunitatea

3.5.11 Mezuak irakurri

Mezuak irakurtzeko, erabiltzailea komunitatea atalean sartu behar da. Bertan komunitatean bidali diren mezu guztiak azalduko dira, egunaren arabera ordenaturik.

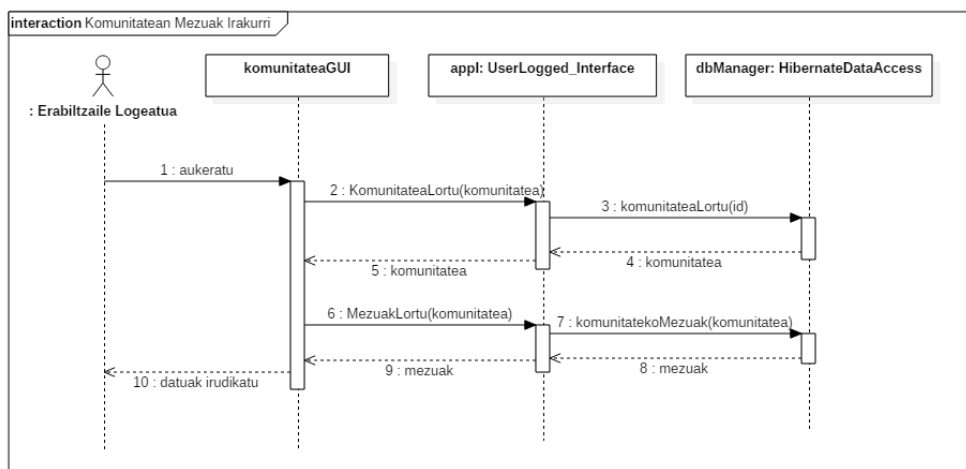


(a) Web

(b) Android

3.31 Irudia: Mezuak irakurriri dagokion interfazea.

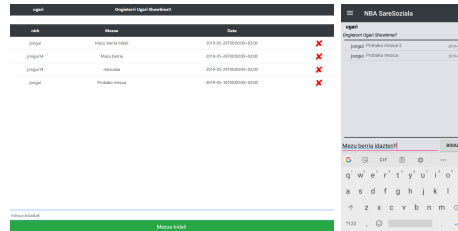
Erabiltzaileak *Komunitatea* atala aukeratzen duenean, zerbitzariari dei egiten dio erabiltzailearen komunitatea parametro gisa pasaz. Honek komunitatea objektu berria pasako digu komunitate horren datu-gehiagorekin (deskribapena, izena...). Aldatu daitezkeen parametroak direnez, nahiago dugu horrela egitea. Ondoren berriz komunitatea objektua pasako diogu dagozkion mezuak lortuz eta pantailan irudikatuz.



3.32 Irudia: Mezuak irakurriri dagokion sekuentzia-diagrama.

3.5.12 Mezuak idatzi

Mezuak idazteko, erabiltzailea komunitatea atalean sartu behar da. Ondorioz, mezu guztiak kargatu eta pantailaratuko dizkio bezero aplikazioak. Ondoren, erabiltzaileak karaktere-kate bat idatziz, mezua bidaltzeko aukera izango du.



(a) Web

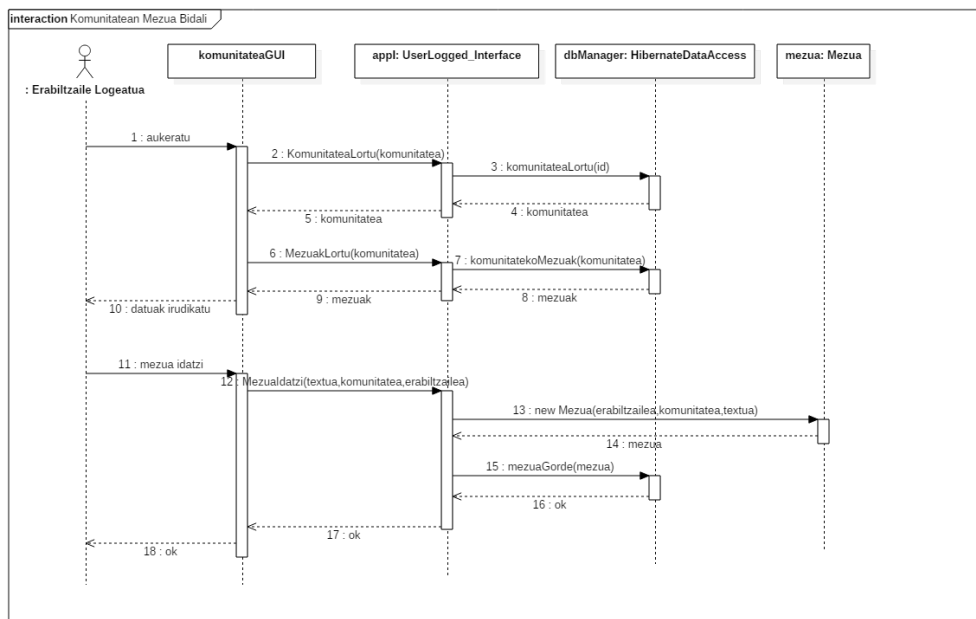
(b) Android

3.33 Irudia: Mezuak idazteari dagokion interfazea.

Erabiltzaileak *Komunitatea* atala aukeratzen duenean, zerbitzariari dei egiten dio mezu guztiak lortuz eta bezeroak hauek irudikatzen ditu (lehen azaldu dugun bezala).

Ondoren, erabiltzaileak mezu bat idazten du eta *Bidali* botoiarekin zerbitzariari dei egiten dio *MezuaIdatzi* funtzioaren bidez, parametro gisa erabiltzailea, komunitatea eta mezua-ren testua bidaliz.

Zerbitzariak mezua objektua sortu eta datu-basean gordeko du.



3.34 Irudia: Mezuak idazteari dagokion sekuentzia-diagrama.

Klasifikazioa

3.5.13 Jardunaldiko klasifikazioa

Erabiltzaileak jardunaldi bat aukeratzen du (defektuz jokatu den azkena), eta bezeroak jardunaldi horretako klasifikazioa irudikatuko dio.

| Postua | Erabiltzailea | Puntuak |
|--------|---------------|---------|
| 1 | jongui14 | 89 |
| 2 | jongui | 69 |

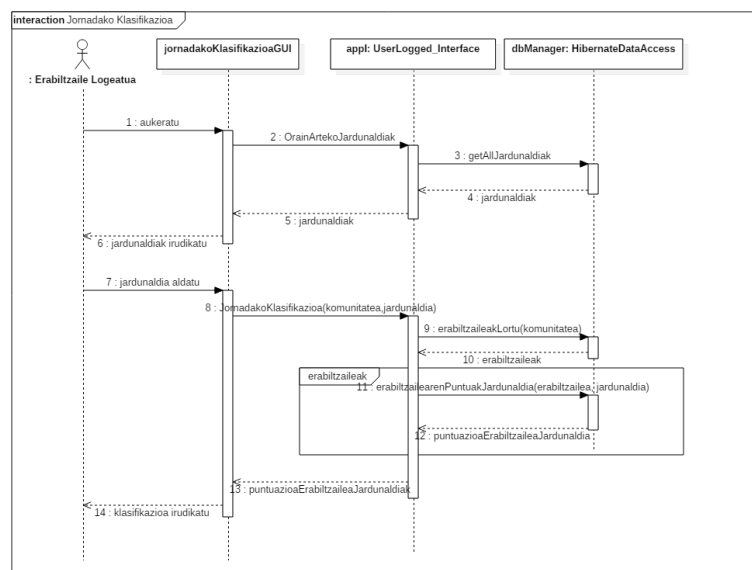
(a) Web

| NBA Sarekoziak | |
|--------------------|-------------|
| Erabiltzaile nik-o | |
| 1 | jongui14 89 |
| 2 | jongui 69 |

(b) Android

3.35 Irudia: Jardunaldiko klasifikazioaren interfazea.

Lehendabizi orain arteko jardunaldi guztiak lortu beharko ditugu *OrainArtekoJardunaldiak* eta ondoren hauek pantailan irudikatu. Jardunaldia aukeratzean (defektuz azkenengo), zerbitzariari dei egingo zaio komunitatea eta jardunaldia bidaliz *JornadakoKlasifikazioa* funtzioan. Honek komunitate erabiltzaileak lortu eta erabiltzaile bakoitzak jardunaldi horretan lorturiko puntuak itzuliko digu. Bukatzeko bezeroak klasifikazioa irudikatuko du.



3.36 Irudia: Jardunaldiko klasifikazioaren sekuentzia-diagrama.

3.5.14 Klasifikazio orokorra

Komunitateko erabiltzaile bakoitzak denboraldi guztian zehar lorturiko puntu guztien klasifikazio orokorra izango da.

| Postua | Erabiltzailea | Puntuak |
|--------|---------------|---------|
| 1 | jongui14 | 89 |
| 2 | jongui | 69 |

(a) Web

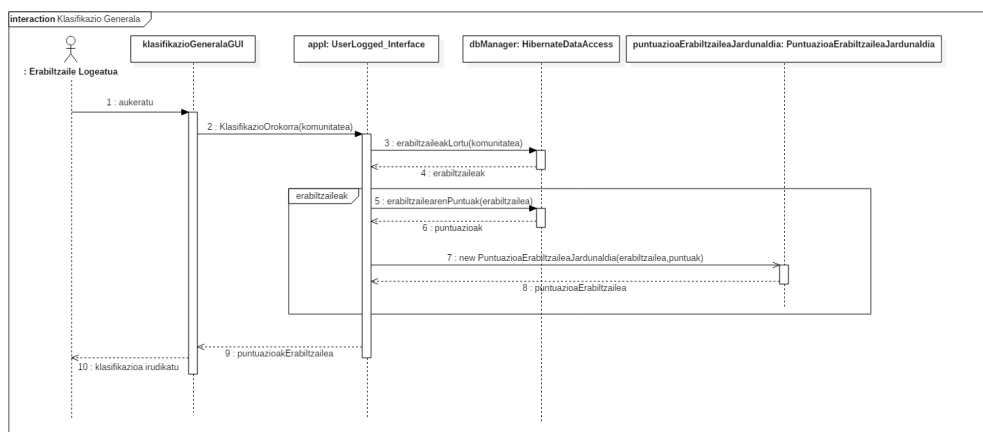
| NBA SareSoziala | | |
|---------------------|----------|----|
| Erabiltzaile nick-a | | |
| 1 | jongui14 | 89 |
| 2 | jongui | 69 |

(b) Android

3.37 Irudia: Klasifikazio orokorraren interfazea.

Aurreko kasua baino nahiko sinpleagoa. Oraingoan komunitatearekin nahikoa izango dugu. Bertan dauden erabiltzaileak lortu, eta erabiltzaile bakoitzak dituen puntu totalak itzuliko ditu.

Ondoren bezeroak aurrekoan bezalako interfaze baten bidez irudikatuko ditu emaitzak.













3.38 Irudia: Klasifikazio orokorraren sekuentzia-diagrama.











3.6 Logeatu gabeko edo logeaturiko erabiltzailea

Logeaturiko zein logeatu gabeko erabiltzaileek hainbat erabilpen-kasu dituzte komunean. Horiek *Jardunaldiko partiduen emaitzak ikustea*, *Jokalarien datuak* eta *Erabiltzaileen alineazio publikoak kontsultatzea* izango ziren.

3.6.1 Eguneko partiduen emaitzak

Partiduen emaitzak kontsultatzeko aukera eskaintzen da aplikazioan, eta hori edozein erabiltzaileek egin dezke. Bertan, egunean jokaturiko partiduen emaitzak zerrendatuko dira.

| 2019-01-31 | 2019-02-01 | | 2019-02-02 | |
|---|------------|-------|------------|---|
| Etxekoa | | | Kanpokoa | |
|  | CHA 100 | Final | MEM 92 |  |
|  | NYK 99 | Final | BOS 113 |  |
|  | MIA 102 | Final | OKC 118 |  |
|  | UTA 128 | Final | ATL 112 |  |
|  | DEN 136 | Final | HOU 122 |  |

| NBA SareSoziala | | | | |
|---|------------|-------|------------|---|
| Ostirala, 1 Otsaila | | | | |
|  | CHA 100 | Final | MEM 92 |  |
|  | NYK 99 | Final | BOS 113 |  |
|  | MIA 102 | Final | OKC 118 |  |
|  | UTA 128 | Final | ATL 112 |  |
|  | DEN 136 | Final | HOU 122 |  |

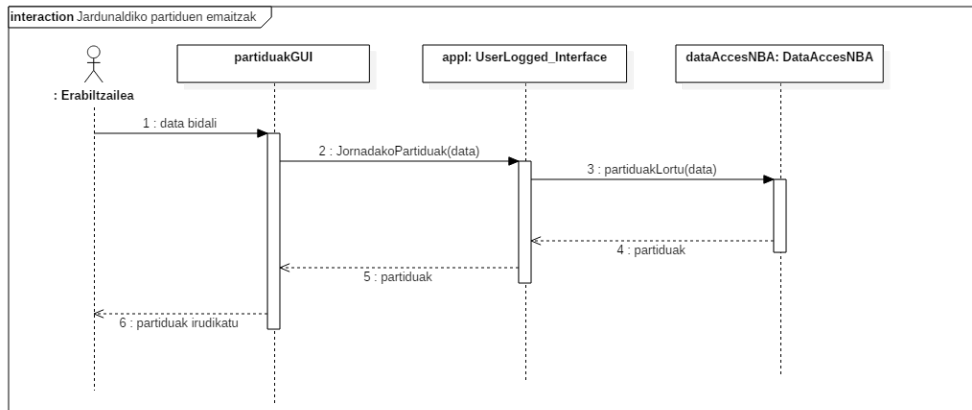
(a) Web

(b) Android

3.39 Irudia: Partiduen emaitzen interfazea.

Berezitasun txiki bat dago, izan ere, datu hauek NBA-ko datu-basetik irakurtzen dira. Modu horretan, momentuan jolasten ari diren partiduen emaitzak denbora errealean jarraitzeko aukera eskaintzen digu.



Bezeroak data bat pasako dio eta zerbitzariak egun horretako partiduak bidaliko dizkio.



3.40 Irudia: Jardunaldiko partiduen emaitzen sekuentzia-diagrama.

3.6.2 Jokalarien datuak kontsultatu

Edozein erabiltzailek ligako jokalaria kontsultatzeko aukera izango du. Modu horretan horien inguruko informazio basikoa jasotzeko aukera izango du.

| Jokalaria | Soldata | Taldea | Posizioa | Dorsala | Altuera | Pisua | Jaiotze-data | Nazionalitatea |
|---|----------|---|----------|---------|---------|-------|--------------|----------------|
|  | 23114066 |  | F | 2 | 2.01 | 104.3 | 1991-06-29 | USA |

Search: kaw

NBA SareSoziala

Kawhi Leonard

Kawhi Leonard

Altuera 2.01m Jaiotze-data 1991-06-29

Posizioa F Pisua 104.3kg Nazionalitatea USA

#2 Soldata 14.530.00\$

Leonar Leonardo

q' w' e' r' t' y' u' i' o' p'

a s d f g h j k l ñ

↑ z x c v b n m

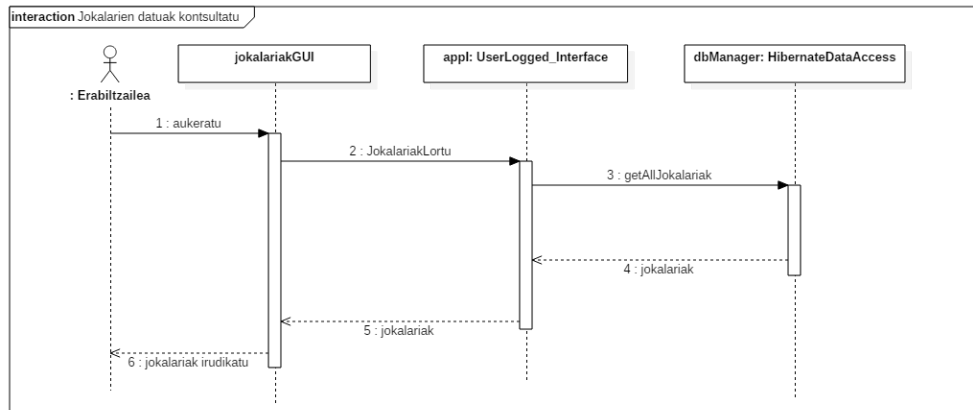
7123 , @ . ←

(a) Web

(b) Android

3.41 Irudia: Jokalarien datuen interfazea.

Prozesua oso sinplea da, erabiltzaileak bere jokalaria kontsultatzean egiten den prozesuaren oso parekoa. Kasu honetan ez da parametririk pasa behar eta jokalaria guztiak lortzen ditugu datu-baseari *getAllJokalariaik* deiaren bidez.

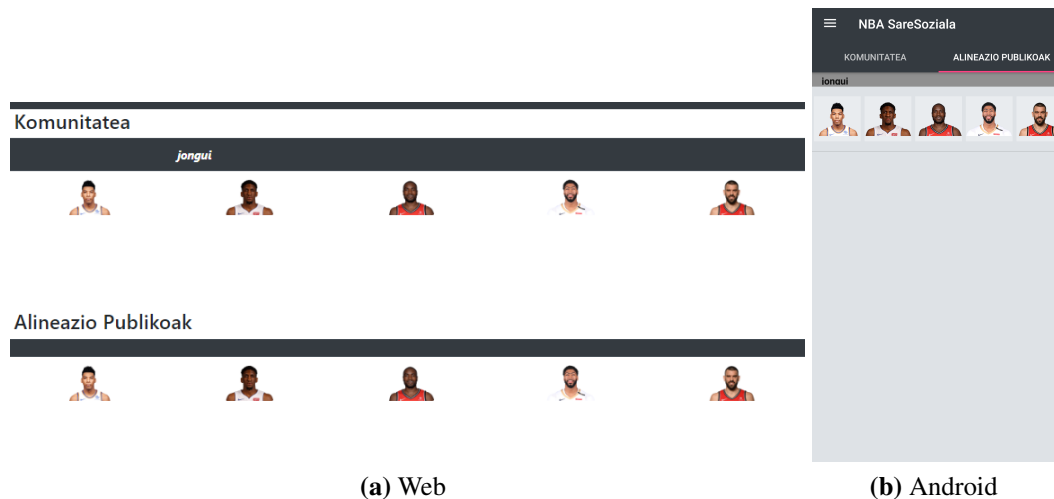


3.42 Irudia: Jokalarien datuak kontsultatu sekuentzia-diagrama.

3.6.3 Erabiltzaileen alineazio publikoak kontsultatu

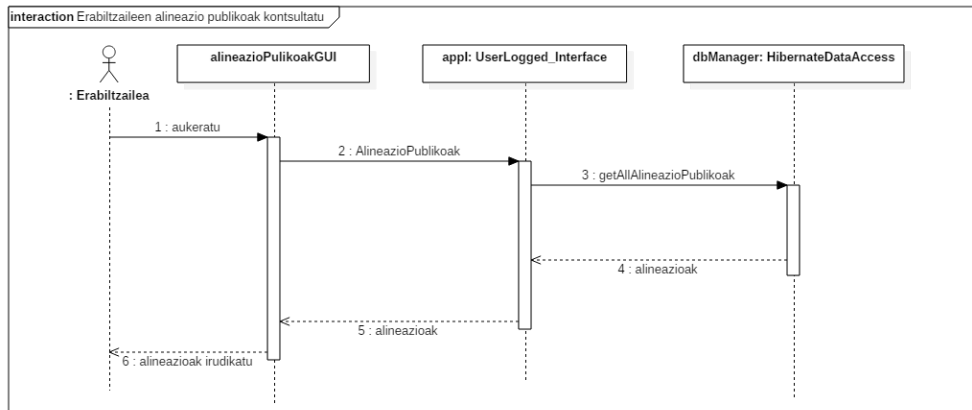
Nahiz eta alineazioak erabiltzaileek soilik eduki, edozeinek ikusi ditzake zeintzuk diren erabiltzaileek publikatzen dituzten alineazioak.

Logeatu-erabiltzaileek aldi berean beraien komunitatean publikoki dauden alineazioak ikusteko aukera izango dute.



3.43 Irudia: Erabiltzaileen alineazio publikoen interfazea.

Bezero aplikazioak *AlineazioPublikoak* funtzioari dei egingo dio, eta honek datu-basetik finkaturik dauden alineazioak lortuko ditu. Modu horretan bezeroari bidali eta honek dagokion moduan irudikatuz.

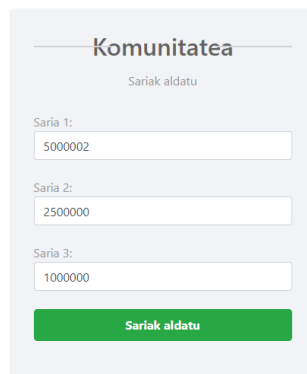


3.44 Irudia: Erabiltzaileen alineazio publikoak kontsultaturen sekuentzia-diagrama.

3.7 Komunitateko administratzailea

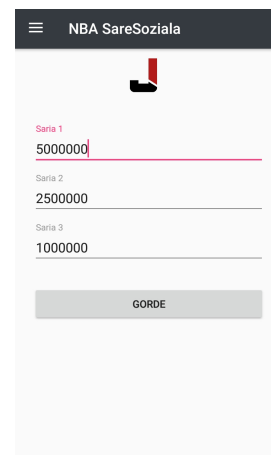
3.7.1 Sariak aldatu

Sariak komunitatea sortzean ezartzen diren balio batzuk dira, eta posible da denboraldia aurrera joan ahala, balio horiek desfasatuak geratzea. Horregatik, komunitateko administratzaileak aldatzeko aukera izango du.



The screenshot shows a web form titled "Komunitatea" with the subtitle "Sariak aldatu". It contains three input fields for prize amounts: "Saria 1:" with the value "5000002", "Saria 2:" with "2500000", and "Saria 3:" with "1000000". A green button labeled "Sariak aldatu" is at the bottom.

(a) Web



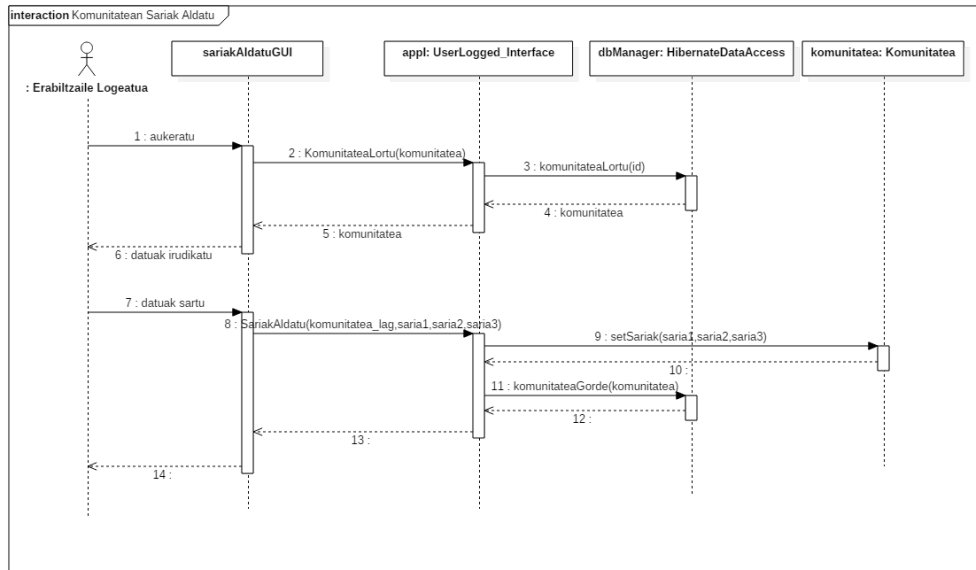
The screenshot shows an Android app interface titled "NBA SareSoziala". It displays three prize amounts: "Saria 1" (5000000), "Saria 2" (2500000), and "Saria 3" (1000000). A grey button labeled "GORDE" is at the bottom.

(b) Android

3.45 Irudia: Sariak aldaturi dagokion interfazea.

Lehendabizi balio zaharrak kargatzen dira eta hauek irudikatu. Gero, erabiltzaileak balio berriak jarri eta gordeko ditu.

Balio horiek posible baldin badira gordeko dira. Balio negatiboak edo 0-a ez dira onartuko eta sarien balioak ordenaturik egon behar dira, handienetik txikienera.



3.46 Irudia: Sariak aldatu dagokion sekuentzia-diagrama.

3.7.2 Mezua ezabatu

Bidalitako mezuren bat desegokia baldin bada, administratzaileak hori ezabatzeko aukera izan beharko du.

| ugari | Ongietorri Ugari Showtime!! | | |
|----------|-----------------------------|---------------------------|---|
| nick | Mezua | Data | |
| jongui | Mezu berria bidali | 2019-05-29T00:00:00+02:00 | ✘ |
| jongui14 | Mezu berria | 2019-05-29T00:00:00+02:00 | ✘ |
| jongui14 | mezuaaa | 2019-05-29T00:00:00+02:00 | ✘ |
| jongui | Probako mezua | 2019-05-16T00:00:00+02:00 | ✘ |

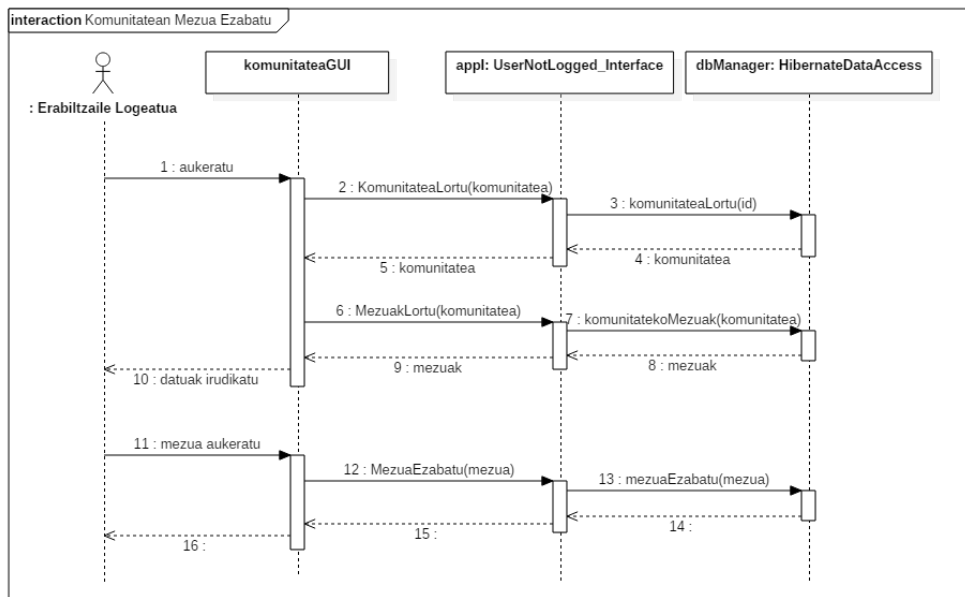
(a) Web



(b) Android

3.47 Irudia: Mezua ezabatzeari dagokion interfazea.

Komunitatean, mezuak azaltzen diren tokian, administratzailearen interfazea desberdina izango da. Mezuen alboan ezabatzeko aukera azalduko zaio. Modu horretan, mezuak modu erosoan ezabatuko ditu.



3.48 Irudia: Mezua ezabatzeari dagokion sekuentzia-diagrama.

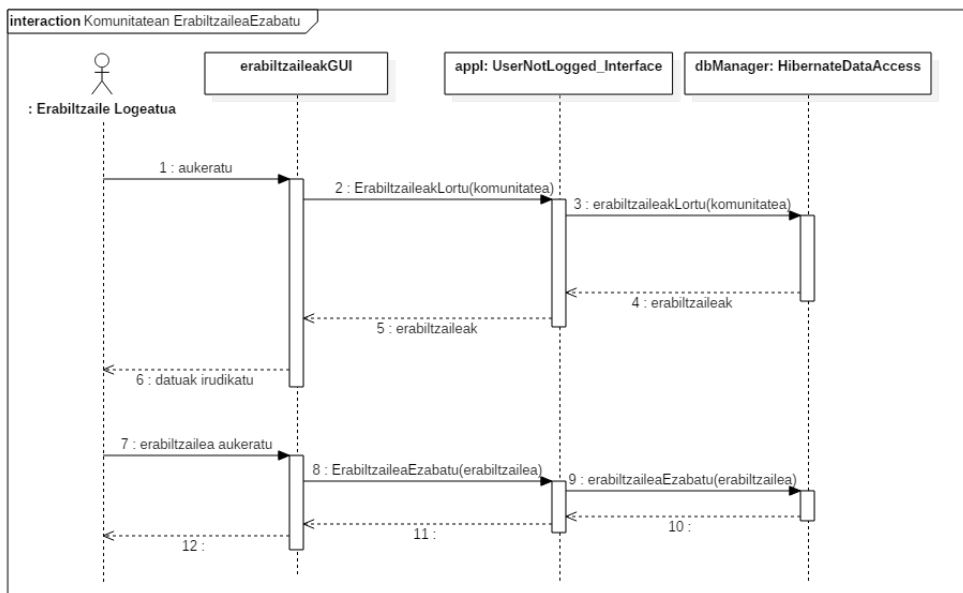
3.7.3 Erabiltzailea ezabatu

Posible da, denboraldian zehar erabiltzaileren batek jolasteari uztea, eta bere jokalariaik beste batzuek nahi izatea. Hori dela eta, administratzaileak erabiltzaile hori ezabatzeko aukera izan beharko du.



3.49 Irudia: Erabiltzailea ezabatzeari dagokion interfazea.

Lehendabizi erabiltzaile guztiak irudikatuko zaizkio. Ondoren, ezabatu nahi duen erabiltzailea aukeratu eta sistemak hori ezabatuko du, bere jokalaria libre utziz, mezuak ezabatuz etab.



3.50 Irudia: Erabiltzailea ezabatzeari dagokion sekuentzia-diagrama.

3.8 Administratzaile globala

Administratzaile globalak soilik bi interfaze izango ditu. Azken finean, erabiltzaileak ezingo dira hemen sartu eta administratzeko erosoagoa da dena batera egotea.

Lehenengo interfazean ekintza gehienak daude. Bakoitza botoi baten bidez dago irudikatua, eta pare-bat ezik, gainontzeko guztiek ez dute sarrerako daturik behar.

Bigarrenean jardunaldiak azaltzen dira bata bestearen segidan. Eskuineko botoiaren bidez jardunaldi horretarako puntuak eguneratzeko aukera izango dugu. Dena oso sinplea.

The screenshot shows a web interface with a list of blue buttons on the left and a table on the right. The buttons are labeled as follows:

- Jokalariak eguneratu
- Soldatak eguneratu
- Taldeak eguneratu
- Alineazioak eguneratu
- Merkatuak eguneratu
- Mezua bidali
- Jardunaldiak Hasieratu

Below the buttons, there are input fields for:

- Mezuren edukia...
- Urtea: 2019
- Hilabetea: 10
- Eguna: 10

On the right, a table is partially visible with the following headers:

| Id | Hasiera | Bukaera |
|----|---------|---------|
|----|---------|---------|

(a) Nagusia

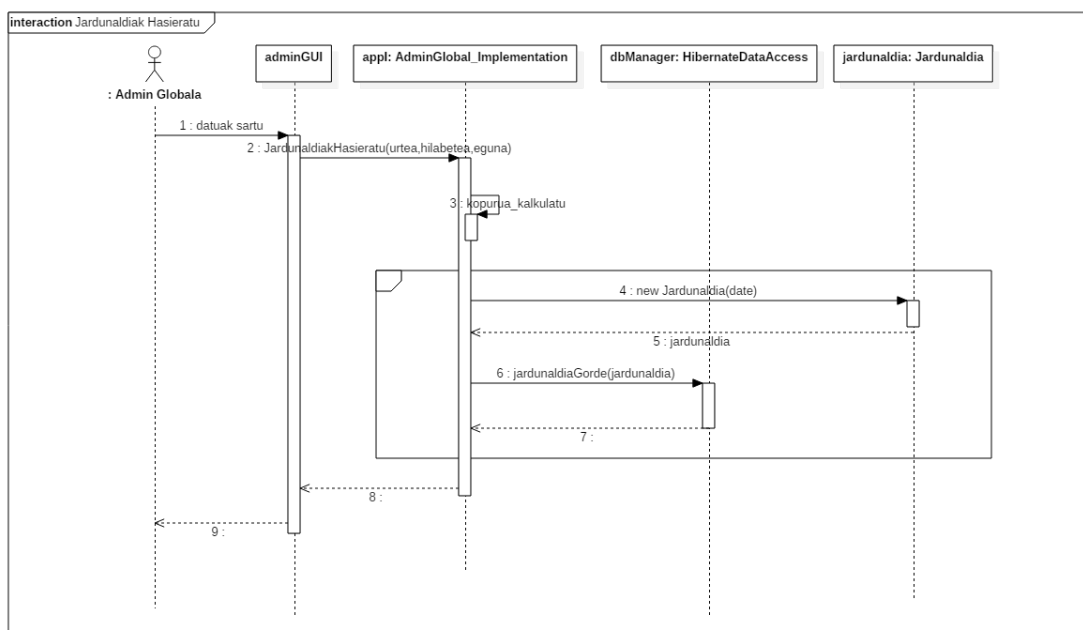
(b) Puntuak eguneratu

3.51 Irudia: Administratzaile globalaren interfazea.

3.8.1 Jardunaldiak hasieratu

Administratzaileak tenporada hasiko den eguna zein den zehatuko du, eta egun horretarik ekainaren azken asteraino, jardunaldiak sortuko ditu astelehenetik ostiralera. Nahiz eta seguruena azkenengoak ez erabiltzea, hobe da sortuak egotea hasieratik. Gainera, erabiltzaile arruntak ez du ezer jakingo.

Ondoren sortutako jardunaldi bakoitza datu-basean gordeko da.

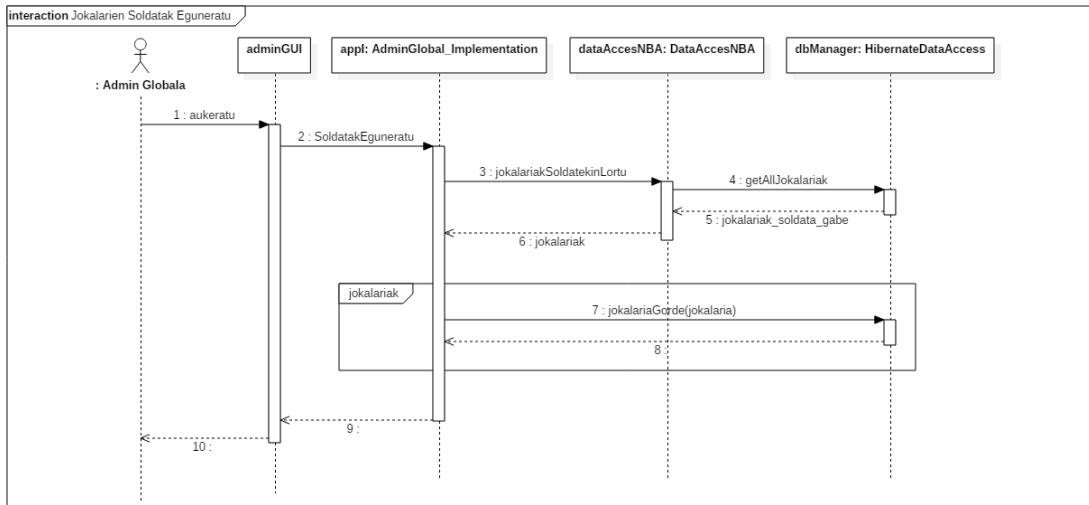


3.52 Irudia: Jardunaldiak hasieraturen sekuentzia-diagrama.

3.8.2 Soldatak eguneratu

NBA-ren datu-basetik irakurriko ez den datu bakarra izango da. Ondoren azalduko dugun bezala, datu hau jasotzeko *web scraping* teknika erabiliko dugu, baina hori beste atal bat da. Hau aipatzearen arrazoi bakarra, *Hibernate*-en klasea erabili beharrean beste klase bat erabiliko dugu.

Klase horrek datu-basetik jokalaria kargatu, soldatak ezarri eta itzuli egingo ditu. Ondoren, banan-banan jokalaria objektu bakoitza gure datu-basean gordeko dugu, aurretik gordetakoak eguneratuz.



3.53 Irudia: Jokalarien soldatak eguneratzearen sekuentzia-diagrama.

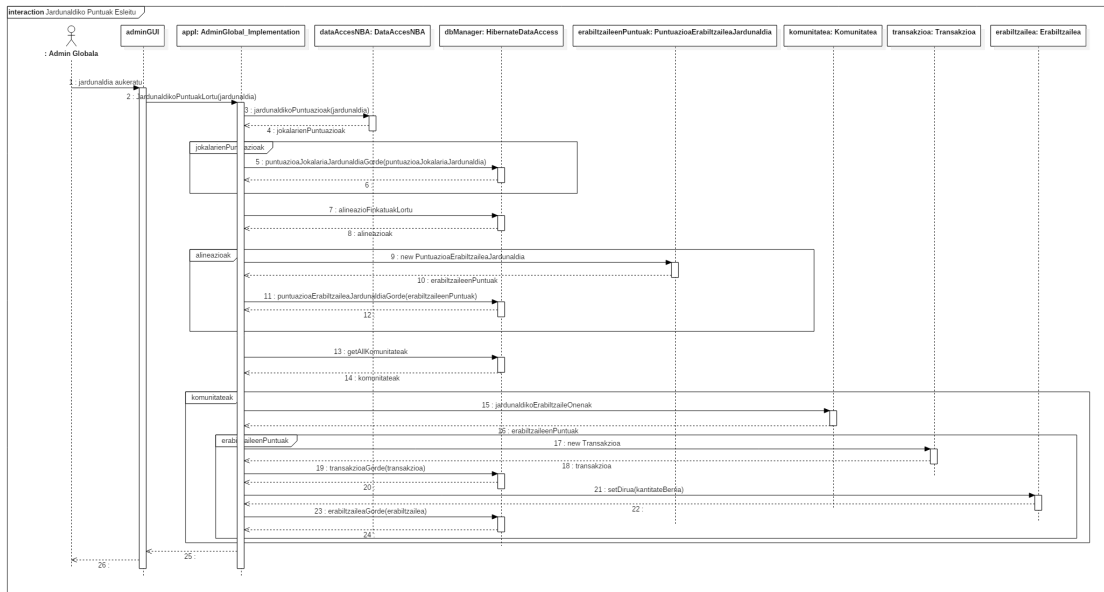
3.8.3 Jokalari eta erabiltzaileei jardunaldiko puntuak esleitu

Administratzaileak jardunaldi bat aukeratu du. Jardunaldi horrek hasiera egun bat du (astelehena) eta aste guztia ordezkaten du, beraz, aste horretan jolastu diren partidu guztien datuak irakurriko dira NBA-ko datu-basetik eta algoritmo baten bidez jokalarien puntuak lortu.

Ondoren, lortutako puntuazio bakoitza gordeko dugu.

Gero, finkaturik dauden alineazio guztiak kargatuko ditugu eta bertan dauden jokalarien puntuak batuz, erabiltzaileei puntuak esleituko zaizkie.

Bukatzeko, komunitate guztiak hartuko ditugu, eta komunitate horretan jardunaldiko lehen sailkatuei saria emango zaie, diru kantitate berria esleituz eta diru transferentzia hori irudikatuko duen transakzioarekin.

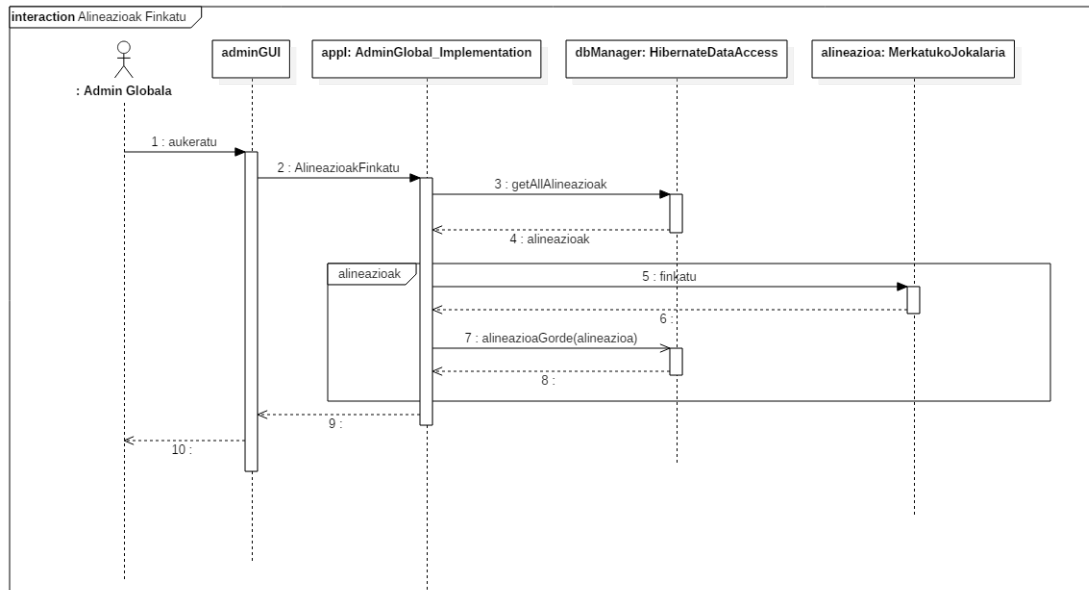


3.54 Irudia: Jokalari eta erabiltzaileei jardunaldiko puntuak esleitu sekuentzia-diagrama.

3.8.4 Alineazioak finkatu

Erabiltzaileei dagozkien puntuak kalkulatzeko, alineazioak finkatu behar dira. Erabiltzaileek badakite jardunaldi horretan puntuak lortzeko, astelehen goizerako (ordua finkatzear) alineazioak egin behar dituztela. Momentu horretan alineazio horiek datu-basean gordeko dira eta ezingo dute horiek aldatu hurrengo asteraino. Hala ere, alineazioa aldatzeko aukera dute, baina alineazio horrek ez die punturik emango.

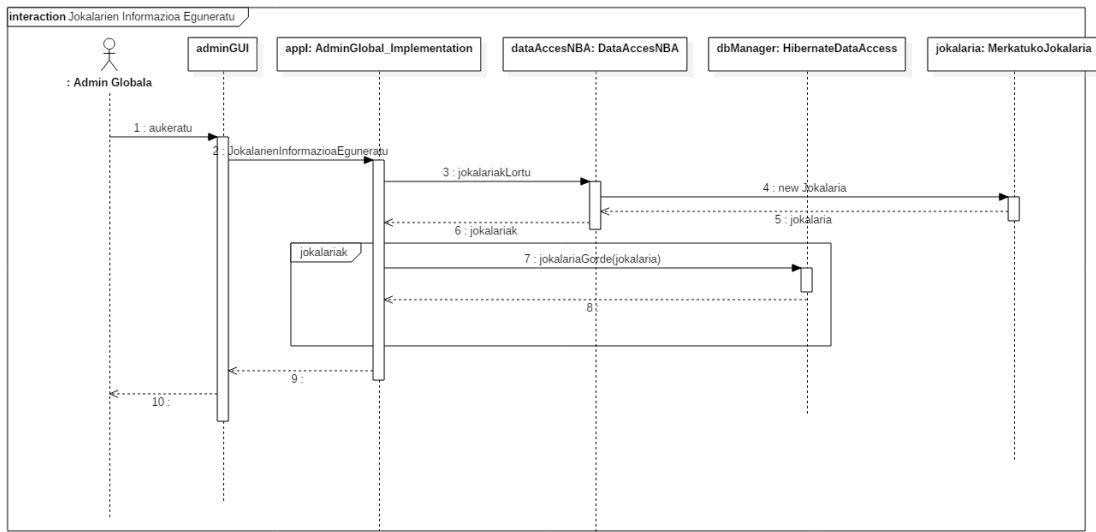
Hori horrela izanik, *AlineazioakFinkatu*-ren bidez datu-basetik lortzeko ditugu alineazio guztiak eta hauek finkatzen ditugu, hau da, $mota == 0$ duten alineazioei 1-ekoa esleituko zaie.



3.55 Irudia: Alineazioak finkaturen sekuentzia-diagrama.

3.8.5 NBA-ko jokalarien informazioa eguneratu gure datu-basean

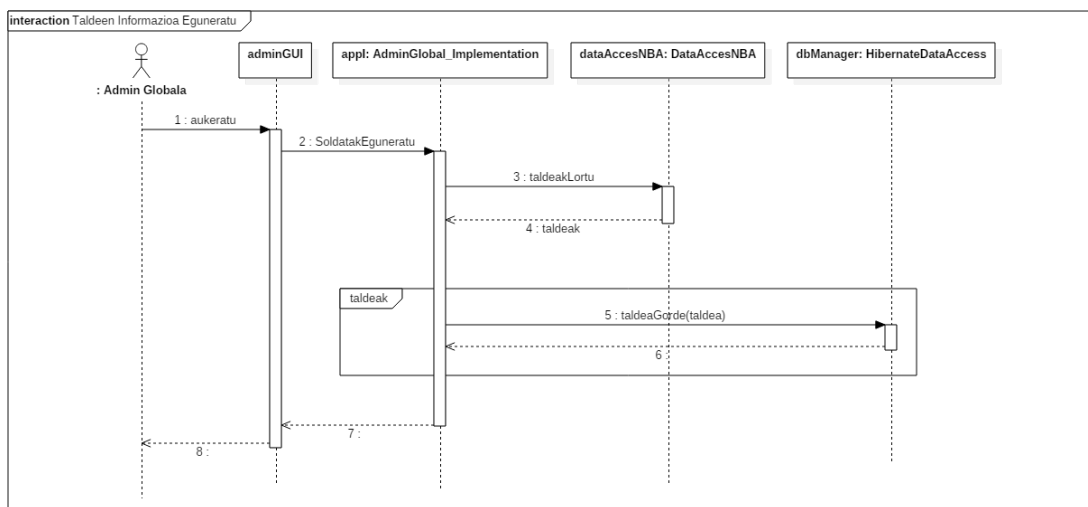
Jokalarien informazioa NBA-ko datu-basetik jaso beharko dugu eta horretarako beraien API-a erabili. Jokalari bakoitzeko objektu bat sortuko dugu eta ondoren objektu horietako bakoitza datu-basean gorde edo ordezkatu dugu. Jokalariak ez dira errepikatuak egongo, erabiltzen dugun identifikadorea NBAREN berdina delako. Modu horretan jokalarien argazkiak etab. lortzea errazagoa izango da.



3.56 Irudia: NBA-ko jokalarien informazioa eguneratu gure datu-basean sekuentzia-diagrama.

3.8.6 NBA-ko taldeen informazioa eguneratu gure datu-basean

Taldeen informazioa NBA-ko datu-basetik jaso beharko dugu eta horretarako beraien API-a erabili. Jokalariekin bezala, talde bakoitzeko objektu bat sortuko dugu eta ondoren horietako bakoitza datu-basean gorde.



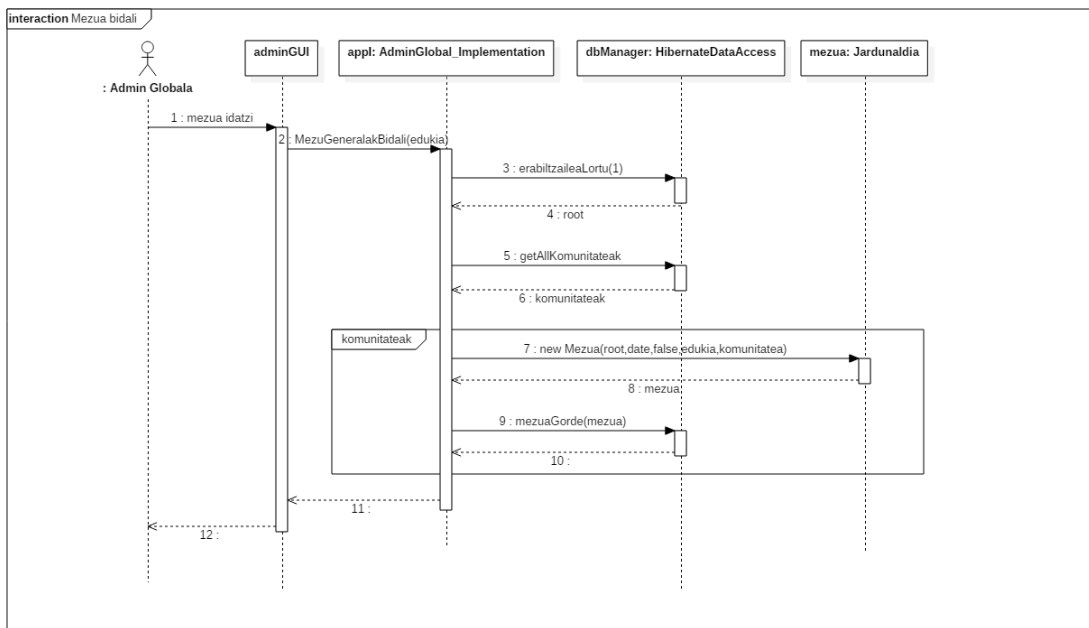
3.57 Irudia: NBA-ko taldeen informazioa eguneratu gure datu-basean sekuentzia-diagrama.

3.8.7 Komunitateei mezu orokorrak bidali

Lehendabizi administratzaileak testuaren edukia jarri beharko du. Testu hori izango da komunitate guztietan ikusiko den mezua.

Mezu guztiak erabiltzaile batek idazten dituenez, *root* erabiltzailea lortu behar dugu.

Ondoren komunitate guztiak lortu eta komunitate bakoitzeko *Mezua* objektua sortu eta datu-basean gorde.

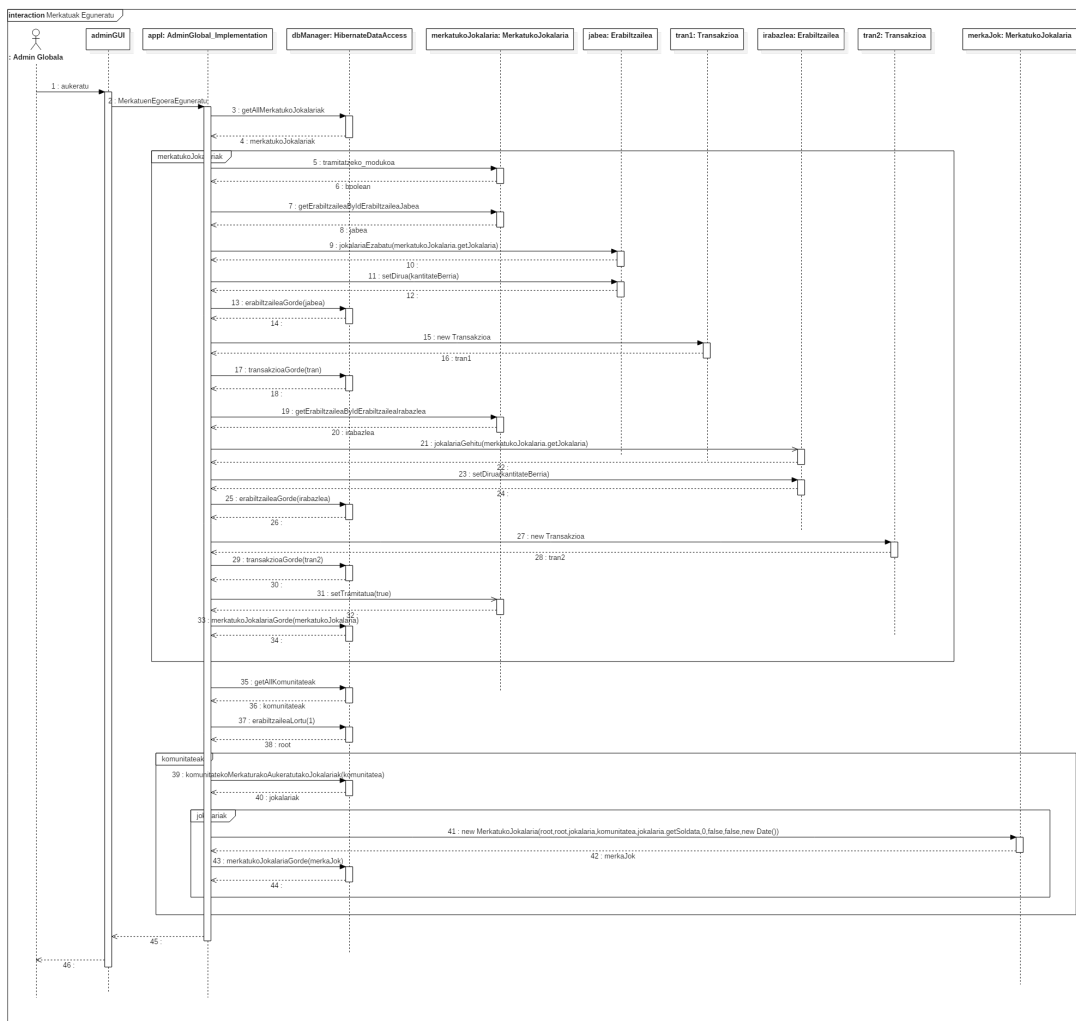


3.58 Irudia: Komunitateei mezu orokorrak bidaliri dagokion sekuentzia-diagrama.

3.8.8 Komunitateen merkatuak eguneratu

Komunitateen merkaturan dauden jokalarien egoera egunero eguneratu beharra dago. Modu horretan, merkaturan dauden jokalariai jabe berriei esleitu, diru transakzioak eta jokalarri berriak merkaturan jarri beharra dago.

Lehendabizi merkatuko jokalarri guztiak lortu beharko ditugu. Ondoren, tramitatugarriak direnak aukeratu. Jabeari jokalaria eta dirua aldatuko diogu, ondoren jabea gorde eta egin den diru mugimenduaren transakzioa, eta hau ere datu-basean gorde. Jabearekin egin dugun bezala, irabazlearekin egingo dugu, jokalaria gehituz eta beste transakzio bat eginez. Ondoren komunitate guztiak jasoko ditugu. Komunitate bakoitzean libre dauden eta gure parametroak betetzen dituzten zenbait jokalarri jasoko ditugu, eta horiekin merkatuko-jokalariai sortuko ditugu, ondoren datu-basean gordez.



3.59 Irudia: Komunitateen merkatuak eguneratzen sekuentzia-diagrama.

4. KAPITULUA

Garapena

Kapitulu honetan garapenean erabilitako tresna edo teknologia bakoitzaren urrats garrantzitsuenak zeintzuk diren erreparasatuko ditugu.

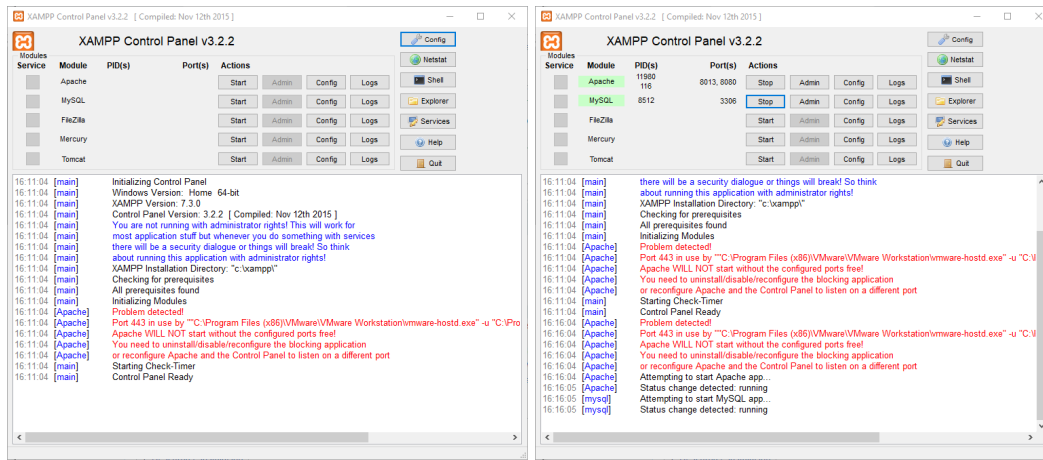
4.1 Zerbitzaria

4.1.1 MySQL, PhpMyAdmin, Apache eta Xampp

Zerbitzaria garatzen hasi aurretik, datu-basea sortu behar dugu. Horretarako *MySQL* eta *phpMyAdmin* erabiliko dugu.

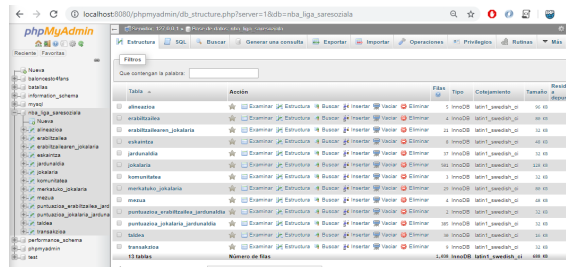
- *MySQL* datu-baseak kudeatzeko sistema bat da.
- *phpMyAdmin* PHP gainean idatzita dagoen tresna bat da. Web nabigatzaile baten bitartez *MySQL* administrazioa egiteko aukera eskaintzen digu.

Hau guztia martxan jartzeko, **Xampp** software libre programa erabiliko dugu. Datu-basea funtzionamenduan ipintzeko eta administratzeko balioko digu.



4.1 Irudia: Xampp-en kontrol panela.

Martxan jarri ondoren, nabigatzailea erabiliz aurretik diseinaturiko datu-basea inplementatzeko ordua izango da. Hau da lorturiko emaitza:



4.2 Irudia: phpMyAdmin-en interfazean gure datu-basea.

Datu-basearen egitura bukatutzat eman dezakegu, baina oraindik konexioa ezarri ahal izateko azken urratsa eman behar dugu, *erabiltzaileen* kudeaketa.

Guk nahi dugun pasahitz bat ezarri beharko diogu *root* erabiltzaileari konexioa ezarri ahal izateko. Bigarren zutabea dei egingo diogun zerbitzariaren izena izango da.

| | | | | | | | |
|--------------------------|------|-----------|----|----------------|----|--|--|
| <input type="checkbox"/> | root | 127.0.0.1 | Si | ALL PRIVILEGES | Si | | |
| <input type="checkbox"/> | root | :::1 | No | ALL PRIVILEGES | Si | | |
| <input type="checkbox"/> | root | localhost | Si | ALL PRIVILEGES | Si | | |

4.3 Irudia: Erabiltzaileen kudeaketa phpMyAdmin.

4.1.2 Hibernate

Hibernate-en bidez datu-base erlazional tradizional baten eta objektuei orientaturiko aplikazio baten arteko komunikazioa errazteko erabiltzen den mapeaketa erreminta bat da. Honen bidez, gure *Java*-ko domeinua eta datu-basean *SQL* bidez ditugun taulen arteko komunikazioa ezarriko dugu.

Hau horrela izateko, *.java* fitxategiez gain, *.hbm.xml* mapeaketa fitxategiak beharrezkoak izango dira. Fitxategi hauek sortzea oso astuna eta konplexua izan daiteke, beraz, *Eclipse*-k eskaintzen digun erreminta bat erabiliko dugu bai *java* klaseak zein mapeaketa fitxategiak **datu-basetik** modu automatikoa sortzeko.

1. *Konexioa ezarri*: Lehen esan dugun bezala, konexioa ezarri beharko dugu zerbitzaria programatzen dugun tokitik datu-basera. Ondoren, bi fitxategi lortuko ditugu *hibernate.cfg.xml* konfigurazio fitxategia eta *hibernate.reveng.xml*. Azken honen bidez, datu-basean ditugun taulak klase bihurtuko ditugu alderantzizko ingeniariatza eginez.



4.4 Irudia: Hibernate konexioa ezartzen.

2. *Alderantzizko ingeniariatza*: Datu-basetik automatikoki klaseak sortzeko aukera eskaintzen digu *Hibernate*ek. Horrela, errore asko saihestez gain, prozesua automatizatu eta asko azkartzen digu. Hala ere, badira zenbait klase eskuz sortu beharko ditugunak: datu-basean agertzen ez diren taulen objektuak (*Partidua* adibidez).

arteko erreferentziek zikloak sortzeko arriskua dute.

Hori saihesteko, *XML etiketak* erabili beharko ditugu.

- Klaseak direla adierazteko.

```

1      @SuppressWarnings("serial")
2      @XmlAccessorType(XmlAccessType.FIELD)
3      @Entity
4

```

- Objetuaren identifikadorea.

```

1      @XmlID
2      @XmlJavaTypeAdapter(IntegerAdapter.class)
3      @Id
4      @GeneratedValue
5

```

Kasu hauetan, beharrezkoa izango da klase laguntzaile bat erabiltzea. Identifikadorea zenbaki oso bat denean ez dugu arazo handirik izango, baina identifikadorea hainbat atributuren arteko konbinaketa denean horietako bakoitzarentzat klase laguntzaile bat sortu beharko dugu.

Hau izango litzateke *AlineazioId* klaseari dagokion adaptatzailea:

```

1      class AlineazioIdAdapter extends XmlAdapter<String, AlineazioId> {
2
3      public AlineazioId unmarshal(String s) {
4      return new AlineazioId(Integer.parseInt(s.split("_")[0]), Integer.parseInt(s.split("_")
5      "[1]));
6
7      }
8
9      @Override
10     public String marshal(AlineazioId v) throws Exception {
11     if (v == null) return "";
12
13     return v.getIdErabiltzailea()+"_"+v.getMota();
14     }
15
16     }

```

- Atributu sinpleek ez dute etiketa berezirik behar. Hau da, interferentziarik sortuko ez duten atributuek ez dute tratamendu berezirik behar.

- *Erreferentzia* egiten duten atributuak. Erreferentzia bi objektuen artean egiten diren kasuetan, horietako bati etiketa berezi bat jarri behar zaio. Etiketa horren bitartez web-zerbitzuak ez du objektu "guztia" bidaliko, soilik dagokion identifikadorea. Modu horretan erredundantzia saihestuko dugu.

| | |
|---|------------|
| 1 | @Xml.IDREF |
| 2 | |

4.1.4 Datuak JSON formatutik irakurri

NBA-ren estatistika eta datu ofizialak JSON formatuan eskuragarri ditugu, aurrerago esan bezala. Ondorioz, datu horiekin lan egiteko irakurri egin beharko ditugu.

Horretarako *json-simple* liburutegi erabiliko dugu, ondorengo klaseak erabiltzeko aukera eskainiz:

- JSONParser
- JSONObject
- JSONArray

4.1.5 Web scraping

Nahiz eta NBAren datu-basea oso handia izan, badira zenbait aspektu kontenplazten ez direnak. Horien artean gure proiektuan oso garrantzitsua den soldata dago.

Jokalarien soldatak publikoak dira eta kirol horretako egunkari digital ia guztiek dituzte datu horiek. Ondorioz, *Web scraping*-a egitera behartuta ikusi nintzen.

- Zer da? Software-programen bidez web guneetatik informazioa ateratzeko teknika bat. Hau lortzeko, *HTTP* protokoloa irakurriko duen *script* bat diseinatu da.
- Segurua da? Web-orriaren *html*-ak aldaketarik jasotzen baldin badu, posible da *script*-ak egoki funtzionatzeari uztea.
- Txukuna da? Nire ustetan oso teknika zikina da, baina kasu honetan oso beharrezkoa. Ligan 500 jokalaria inguru daude, eta bakoitzari soldata banaka ezartzea oso lan astuna izan daiteke.

- Funtzionatzen du? Egia da, zenbait jokalariren izenekin arazoak eduki ditudala (informazioa oso modu desberdinean gordetzen baita toki bakoitzean), baina orokorrean prozesua egoki joan. Gainera, script-a nahiko ahaltsua da eta martxoan web-orriak jaso zuen diseinu aldaketa arazorik gabe jasan zuen.

4.2 Web-Bezeroa

4.2.1 Apache

Web-aplikazioa azken finean web-orri bat izango da, eta horretarako zerbitzari bat behar dugu. Gure kasuan, zerbitzari nagusiaren datu-basea martxan jartzeko *Apache* zerbitzari bat erabili dugunez, zerbitzari bera erabiliko dugu.

Ondorioz, ez dugu erreminta gehigarririk instalatu behar izango, zerbitzari nagusian *Xampp*-a martxan jartzen dugunean, web-aplikazioaren zerbitzaria ere martxan jarriko da.

Zerbitzari honek nabigatzaileei eskaturiko orriak bidaltzeaz enkargatuko da.

Oharra: Lehen datu-basea martxan jartzeko *Apache*-ere martxan jarri behar dela esan dut, baina hori ez da egia. *Apache*-a beharrezkoa izango da *phpMyAdmin*-etik lan egin nahi baldin badugu. Orduan garrantzitsua izango da *Apache* martxan dagoela ziurtatzea.

4.2.2 Nusoap

Php erabiliz web-zerbitzuak kontsumitzeko ahalmena eskaintzen digun liburutegia da. Deskarga egin ondoren, erabilera oso sinplea du.

1. Lehendabizi inportatu egin behar genuke.
2. Ondoren bezero objektu bat sortu *wSDL* helbidea jarritz.
3. Bukatzeko deituko dugun funtzioaren izena eta jarraian array baten bidez parametroak.

Adibide bezala erabiltzailearen e-maila libre dagoen funtzioari deia:

```

1 <?php
2     $email = $_GET['email'];
3
4     include '../datuak/domainua.php';
5     include '../datuak/datuak.php';
6     $wsdl = $usernot_wsdl;
7     //session_start();
8
9     error_reporting(E_ERROR | E_PARSE);
10    require_once('../lib/nusoap/nusoap.php');
11
12    //Create Client & set WSDL file path
13    $client = new nusoap_client($wsdl, 'wsdl');
14
15    // Check for any error(s)
16    $err = $client->getError();
17    if ($err){
18        echo '
19        <h2>Constructor error</h2>
20        <pre>' . $err . '</pre>
21        ' . $err . '</pre>
22        ' . $err . '</pre>
23        ' . $err . '</pre>
24        ' . $err . '</pre>
25        ' . $err . '</pre>
26        ' . $err . '</pre>
27        ' . $err . '</pre>
28        ' . $err . '</pre>
29        ' . $err . '</pre>
30        ' . $err . '</pre>
31        ' . $err . '</pre>
32        ' . $err . '</pre>

```

4.7 Irudia: Adibidea: Nusoap liburutegiaren erabilera e-maila libre dagoen kontsultatzeko.

4.2.3 Bootstrap

Webguneak eta aplikazioak diseinatzeko kode irekiko tresna multzo edo liburutegi (framework) plataforma-anitza da. Aurretik definiturik dauden diseinu ugari ditu: letrak, botoiak, formularioak, menuak ...

4.2.4 JQuery, JavaScript, CSS eta HTML5

Web-aplikazioak garatzerako orduan teknologia erabilienak. Hauen artean interfaze grafikoei itxura eta funtzionaltasuna ematen diote.

Jquery, *CSS*-ko estilo orriak eta *Bootstrap* erabil ahal izateko, hauek inportatu beharko ditugu:

```

1 <link rel="stylesheet" href="../css/styles.css">
2 <link rel="stylesheet" href="../lib/bootstrap/css/bootstrap.min.css">
3 <script src="../lib/jquery/jquery-3.3.1.min.js"></script>
4 <script src="../lib/bootstrap/js/bootstrap.min.js"></script>

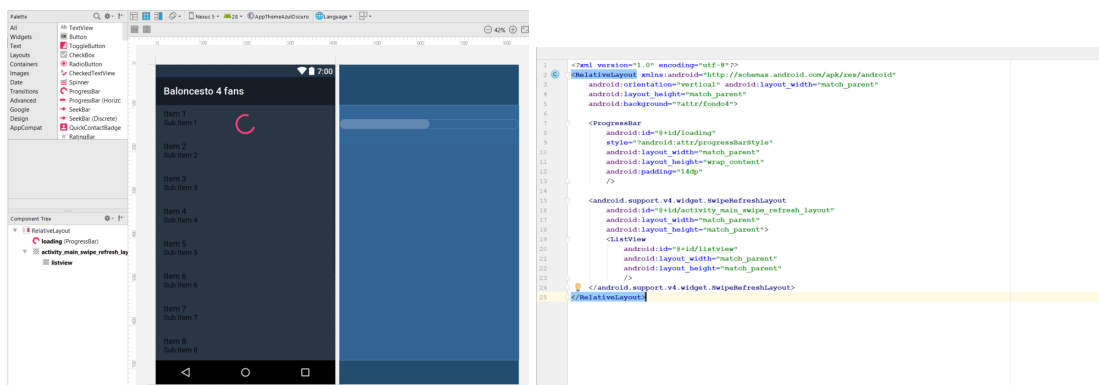
```


4.3 Android-Bezerao

4.3.1 Android-Studio

Android-erako aplikazioa garatzea erabaki nuen unetik, ez nuen zalantzarik eduki erabiliko nuen garapen erremintaren inguruan.

Java lengoaiaren bidez garatuko dugu programazio atal guztia, eta interfazeak garatzeko eskaintzen dituen *widget*-ak. Aipatu beharra dago interfaze grafikoak *xml* fitxategi baten bidez errepresentatzen direla.



4.8 Irudia: Interfaze grafikoak Android Studion.

Horrez gain, mugikorra erabiltzea nahi ez baldin badugu, aplikazioa emulatzeko aukera eskaintzen digu gailu birtualen bidez, bertsio kudeaketa egiteko sistema barneratu bat du *VCS*, *Local History* eta beste hainbat abantaila gehiago.

4.3.2 Picasso

Interfazeek jokalarien eta taldeen argazki ugari dituzte interfaze bakoitzean. Horren ondorioz, interfazeen lan-karga handia da eta gainera irudi horiek zerbitzarian baldin badaude, karga denbora asko handitzen da.

Hori saihesteko, *Picasso* liburutegia erabili dugu. Modu horretan, irudien karga asinkrono egiten dugu erabiltzailearen esperientzia hobetuz.

Hainbat modutan karga ditzakegu liburutegiak:

- *.jar*-a deskargatu eta kokalekua adierazi

- liburutegia zein den adierazi eta aplikazioa bera arduratzea.

Ondoren azalduko dugun *kSoap* deskargaren bidez egin genuen, beste biak aldiz erreferentzia erabiliz.

```
compile files('libs/ksoap2-android-2.5.2.jar')

compile 'com.squareup.picasso:picasso:2.5.2'
compile 'com.google.firebase:firebase-messaging:10.0.1'
```

4.9 Irudia: *build.gradle* app-ren zati bat.

4.3.3 Ksoap

Bezerao java lengoaiaren bidez idatzia zegoenez, hasieran *Service.create* funtzioa erabili nahi nuen interfazeak modu erosoan erabili ahal izateko. Ezinezkoa izan zen.

Horregatik, kanpo liburutegi bat erabiltzeko beharra izan genuen, *kSOAP 2*. Honen bidez, web-zerbitzuak kontsumitzeko aukera genuen.

Parametro konplexuak bidali

Domeinuko klase bat bidaltzerako, hainbat funtzio extra inplementatu behar dira, *KvmSerializable* interfazearenak:

- *public Object getProperty(int i)*
- *public int getPropertyCount()*
- *public void setProperty(int i, Object o)*
- *public void getPropertyInfo(int i, Hashtable properties, PropertyInfo info)*

```

@Override
public Object getProperty(int i) {
    switch (i) {
        case 0: return idMezua;
        case 1: return eguna;
        case 2: return ezabagarria;
        case 3: return mezua;
    }
    return null;
}

@Override
public int getPropertyCount() { return 4; }

@Override
public void setProperty(int i, Object o) {
    switch (i) {
        case 0: idMezua=Integer.parseInt(o.toString());break;
        case 1: eguna=(Date) o;break;
        case 2: ezabagarria=Boolean.parseBoolean(o.toString());break;
        case 3: mezua=o.toString();break;
    }
}

@Override
public void getPropertyInfo(int i, Hashtable properties, PropertyInfo info) {
    switch (i) {
        case 0:info.name="idMezua";info.type = PropertyInfo.INTEGER_CLASS;break;
        case 1:info.name="eguna";info.setType(Date.class);break;
        case 2:info.name="ezabagarria";info.type = PropertyInfo.BOOLEAN_CLASS;break;
        case 3:info.name="mezua";info.type = PropertyInfo.STRING_CLASS;break;
    }
}

```

4.10 Irudia: Adibidea: Mezua klasean KvmSerializable-ren funtzioen implementazioa.

Konexioa ezarri eta datuak jaso

Konexioa ezartzeko, *URL*, *portua*, *namespace*, *wSDL*-a eta *funtzioaren-izena* beharko ditugu.

Lehendabizi *Soap* objektua sortuko dugu, *namespace* eta *funtzioaren-izena* pasaz. Ondoren, beharrezkoak diren parametroak gehitu *PropertyInfo* klasearen bidez.

Hori guztia serializatua izango den objektu bateri gehitu, *http*-a prestatu eta deia egingo dugu.

Zerbitzariaren erantzuna beste *SoapObject* batean jasoko dugu, eta bertatik interesatzen zaizkigun datuak lortu.

```

String funtzio_izena="ErabiltzailearenJokalariakLortu";
SoapObject soapObject = new SoapObject(NAMESPACE, funtzio_izena);
PropertyInfo propertyInfo = new PropertyInfo();
propertyInfo.setName("arg0");
propertyInfo.setValue(Datuak.erabiltzailea);
propertyInfo.setType(Erabiltzailea.class);
soapObject.addProperty(propertyInfo);

SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
new MarshalDate().register(envelope);
envelope.dotNet=false;
envelope.setOutputSoapObject(soapObject);

HttpTransportSE httpTransportSE = new HttpTransportSE(NSDL_UserLogged);
httpTransportSE.call(NAMESPACE+funtzio_izena, envelope);
httpTransportSE.debug = true;

SoapObject result = (SoapObject) envelope.bodyIn;

```

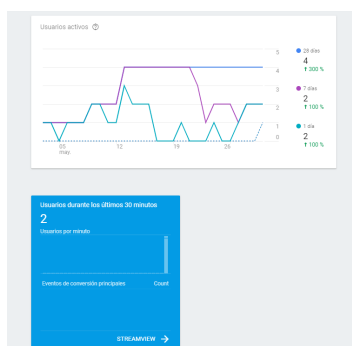
4.11 Irudia: Adibidea: Erabiltzailearen jokalariak lortzeko deia.

4.3.4 Firebase

Bukatzeko *Firebase* dugu. Oso tresna ahaltua eta konplexua da, izan ere web edo mugikorretarako aplikazioak garatzeko ahalmena du. 2014. urtean Google-ek erosi zuenetik, erabilera guztiz handitu da, abantaila ikaragarriak eskaintzen ditu eta.

Hala ere, gure kasuan detaile txiki batzuk besterik ez ditugu erabiltzen:

- *Erabiltzaileen estatistikak*. Oso modu erraz baten bidez, gure aplikazioa erabiltzen duten erabiltzaile kopurua kontrolatu dezakegu.



4.12 Irudia: Erabiltzaileen estatistikak.

- *Notifikazioak bidali*. Erabiltzaileei berriren bat komunikatzeko oso baliagarria izan daiteke notifikazio baten bidez egitea.

4.13 Irudia: Notifikazioak bidali.

Hori guztia erabili ahal izateko, aplikazioa hasieratzean beharrezkoa izango da *FirebaseAnalytics* objektua hasieratzea.

```
1 FirebaseAnalytics mFirebaseAnalytics = FirebaseAnalytics.getInstance(this);
```

4.3.5 Locale

Mugikorretarako bertsioan ordu-aldaketa, koloreak eta hizkuntza aldatzeko aukerak eskaintzen dira. Bereziki azken honen funtzionamendua gainontzekoetatik oso desberdina izan da. Izan ere, *Android Studio*-k karaktere-kate guztiak fitxategietan gordetzeko aukera ematen digu. Modu horretan, hizkuntza bakoitzarentzat azpi-karpeta bat sortuko da dagozkien balioekin: *values-en* (ingelesezkoa), *values-eu* (euskera) eta *values-es* (gaztelazkoa).

Mugikorrak automatikoki defektuz ezarritako lengoaiaren balioak hartzen saiatuko da, eta horiek existitzen ez badira aplikazioak defekturik jarritakoak *values* karpetakoa. Hala ere, guk hori aldatzeko aukera eskaintzen diogu erabiltzaileari eta hori lortzeko, konfigurazioa aldatu beharko dugu eta beste *Locale* bat ezarri.

```
1 Locale myLocale = new Locale(Hizkuntza.hizkuntza);
2 Resources res = getResources();
3 DisplayMetrics dm = res.getDisplayMetrics();
4 Configuration conf = res.getConfiguration();
5 conf.locale = myLocale;
6 res.updateConfiguration(conf, dm);
```

4.3.6 Interfazeen karga asinkronoa

Web-orrietan oso ohikoa egiten da orria kargatu aurretik segundo batzuk itxarotea, baina telefono mugikorretan ezinbestekoa da erabiltzaileak une oro aplikaziotik *feedback*-a jasotzea. Gainera, normalean ordenagailuek mugikorrek baino konexio abiadura handiagoak izaten dituzte.

Hori guztia dela eta, zerbitzariarekin egindako konexioak modu asinkronoan egin ditut. Horretarako datuak beharrezkoak diren harian, beste bati dei egin diot eta datuak lortu bitartean progresio-barra bat ezarri dut (ez du egoera erreala irudikatzen, soilik mugimendua zegoela adierazten du).

Bi kasu nagusi izan ditut:

- *Activity*-a aldatzean, hau da, interfaze batetik bestera salto egitean. Kasu hauetan hari berri bat sortu eta zerbitzaritik datuak jasotzen ditut. Ondoren, interfazean aldaketak egiteko modu *sinkronoa* ezarri behar da, beraz haritik akribitate (edo bere kontextua) lortzen dut eta modu sinkronoan interfazea aldatu. Gainera, ikusgarri zegoen progresio-barra desgaitzen dut.

```

1     new Thread(new Runnable() {
2         public void run() {
3             datuak_lortu()
4             getActivity().runOnUiThread(new Runnable() {
5                 @Override
6                 public void run() {
7                     (rootView.findViewById(R.id.loading)).setVisibility(View.GONE);
8                     interfazea aldatu...
9                 }
10            });
11        }
12    }).start();

```

- *Activity* baten barruan, zerbitzariari dei bat egin eta horren arabera aldaketa bat egin behar dugunean.

Hau adibidez kontuan logeatzean gertatzen da. Zerbitzariari galdetu behar zaio ia erabiltzailearen datuak zuzenak diren, baina ezin dugu aktibitatez aldatu ez dakigulako nora joan. Orduan, emaitza jaso arte erabiltzaileari progresio-mezu bat bistaratuko zaio *ProgressDialog* erabiliz.

Aurreko kasuan moduan hari berri bat sortuko dugu, baina oraingoan asinkronoa den zatian ondorengo klasea instantziatuko dugu:

```

1     final ProgressDialog progressDialog = new ProgressDialog(getContext());
2     progressDialog.setMessage("Kargatzen...");
3     progressDialog.setTitle("Erabiltzailea logeatzen, itxaron.");
4     progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
5     progressDialog.show();
6     progressDialog.setCancelable(false);
7

```

Ondoren, behin karga egin eta interfazean aldaketak egiten hasi, menu hori baztertu beharko dugu trabarik ez egiteko:

```

1     progressDialog.dismiss();
2

```

4.4 Probak

Produktu eta aplikazio guztiak garatutzat eman ondoren, atal oso garrantzitsua dator orain: ondo funtzionatzen dutela bermatzea. Horretarako funtzio gehienak aktibatzen dituzten erabilpen-kasu multzo garrantzitsuenak ikusiko ditugu. Nola egingo ditut probak?

1. Datu-base guttiz utzik laga. Horrela hemendik aurrera egiten diren gauza guztiak aurrekoetatik independenteak izango dira. Mantenduko dudan gauza bakarra *root* erabiltzailea eta bere komunitatea izango da.

| Tabla | Acción | Filas | Tipo | Cotejamiento | Tamaño | Residuo a depurar |
|--------------------------------------|---|-------|--------|-------------------|--------|-------------------|
| alineazioa | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 96 KB | - |
| erabiltzailea | Examinar Estructura Buscar Insertar Vaciar Eliminar | 1 | InnoDB | latin1_swedish_ci | 80 KB | - |
| erabiltzailearen_jokalaria | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| eskaintza | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 48 KB | - |
| jardunaldia | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| jokalaria | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 64 KB | - |
| komunitatea | Examinar Estructura Buscar Insertar Vaciar Eliminar | 1 | InnoDB | latin1_swedish_ci | 32 KB | - |
| merkatuko_jokalaria | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 80 KB | - |
| mezua | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 48 KB | - |
| puntuazioa_erabiltzailea_jardunaldia | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| puntuazioa_jokalaria_jardunaldia | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| taldea | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | - | - |
| transakzioa | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| 13 tabla(s) | Número de filas | 2 | InnoDB | latin1_swedish_ci | 608 KB | 0 B |

4.14 Irudia: Datu-base hutsik.

2. Bi bezero desberdin ditudanez, bakoitzean komunitate bat simulatuko dut. Modu horretan erabilpen-kasu bakoitza gutxienez bi aldiz probatuko dut.
3. Proba bakoitza baloratzeko, datu-basean eduki duen eragina erakutsiko dugu, dena ondo joan dela demostratzeko.
4. Bukatzeko, erabiliko dudan ordena, "orden logikoa" izango da, hau da, denboraldia-ren hasiera dela simulatuko dut.

4.4.1 Denboraldia hasi aurretik datuak eguneratzea

Oraindik ez da erabiltzailerik erregistratu eta izango dugun rola *administratzaile orokorrena* da. Hauek dira exekutatu beharreko funtzioak eta ordena:

1. Taldeak eguneratu
2. Jokalariak eguneratu
3. Soldatak eguneratu
4. Jardunaldiak hasieratu (hasierako data 2018ko urriaren 16a)

| Tabla | Acción | Filas | Tipo | Cotejamiento | Tamaño | Residuo a depurar |
|--------------------------------------|---|-------|--------|-------------------|--------|-------------------|
| alineazioa | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 96 KB | - |
| erabiltzailea | Examinar Estructura Buscar Insertar Vaciar Eliminar | 1 | InnoDB | latin1_swedish_ci | 88 KB | - |
| erabiltzailearen_jokalaria | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| eskaintza | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 48 KB | - |
| jardunaldia | Examinar Estructura Buscar Insertar Vaciar Eliminar | 39 | InnoDB | latin1_swedish_ci | 32 KB | - |
| jokalaria | Examinar Estructura Buscar Insertar Vaciar Eliminar | 497 | InnoDB | latin1_swedish_ci | 128 KB | - |
| komunitatea | Examinar Estructura Buscar Insertar Vaciar Eliminar | 1 | InnoDB | latin1_swedish_ci | 32 KB | - |
| merkatuko_jokalaria | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 88 KB | - |
| mezua | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 48 KB | - |
| puntuazioa_erabiltzailea_jardunaldia | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| puntuazioa_jokalaria_jardunaldia | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| taldea | Examinar Estructura Buscar Insertar Vaciar Eliminar | 30 | InnoDB | latin1_swedish_ci | 16 KB | - |
| transakzioa | Examinar Estructura Buscar Insertar Vaciar Eliminar | 0 | InnoDB | latin1_swedish_ci | 32 KB | - |
| 13 tablas | Número de filas | 568 | InnoDB | latin1_swedish_ci | 688 KB | 0 B |

4.15 Irudia: Datu-basea hasieratua.

4.4.2 Komunitateak eta erabiltzaile kontuak sortu

Lehen esan dugun bezala, bezero bakoitzak proba propioak egingo ditu. Ondorengo taulan ikusi dezakegu sortuko ditugun kontuen *nick*-ak (gainontzeko datu guztiak jarri ezkerro taula oso handia izango zen eta ez orain bezalako argia).

| Bezeroa | Komunitatea | Erabiltzailea |
|--------------------|--------------|----------------|
| Android aplikazioa | komunitatea1 | admin1 |
| | | erabiltzailea1 |
| | | erabiltzailea2 |
| Web aplikazioa | komunitatea2 | admin2 |
| | | erabiltzailea3 |
| | | erabiltzailea4 |

4.1 Taula: Komunitate eta kontuen datu basikoak.

| ↑ Opciones | | | | | | | | | | | |
|--------------------------|-----------------|----------------|--------------------------|-----------|----------------------------|----------|-------------------|-----------|----------|------------------|---------------|
| | idKomunitatea | nick | izen_osoa | pasahitza | ongietorri_mezua | saria1 | saria2 | saria3 | | | |
| <input type="checkbox"/> | 1 | root | | | | 0 | 0 | 0 | | | |
| <input type="checkbox"/> | 3 | komunitatea1 | Lehen komunitatea | 123456 | Ongietorri komunitatera! | 5000000 | 2500000 | 1500000 | | | |
| <input type="checkbox"/> | 4 | komunitatea2 | Bigarren Komunitatea | 123456 | Kaixo komunitate onenera!! | 8000000 | 4000000 | 1500000 | | | |
| ↑ Opciones | | | | | | | | | | | |
| | idErabiltzailea | nick | email | pasahitza | izen_osoa | dirua | administratzailea | hizkuntza | koloreak | ordu_diferentzia | idKomunitatea |
| <input type="checkbox"/> | 1 | root | | | | 0 | 0 | 0 | 0 | 0 | 1 |
| <input type="checkbox"/> | 3 | admin1 | adm1@gmail.com | 123456 | Administratzailea Bat | 40000000 | 1 | 0 | 0 | 0 | 3 |
| <input type="checkbox"/> | 4 | erabiltzailea1 | erabiltzailea1@gmail.com | 123456 | Lehen Erabiltzailea | 40000000 | 0 | 0 | 0 | 0 | 3 |
| <input type="checkbox"/> | 5 | erabiltzailea2 | erabiltzailea2@gmail.com | 123456 | Erabiltzaile Izena Bi | 40000000 | 0 | 0 | 0 | 0 | 3 |
| <input type="checkbox"/> | 6 | admin2 | admin2@gmail.com | 123456 | Bigarren Adminstratzailea | 40000000 | 1 | 0 | 0 | 0 | 4 |
| <input type="checkbox"/> | 7 | erabiltzailea3 | erabiltzailea3@gmail.com | 123456 | Hirugarren Erabiltzailea | 40000000 | 0 | 0 | 0 | 0 | 4 |
| <input type="checkbox"/> | 8 | erabiltzailea4 | erabiltzailea4@gmail.com | 123456 | Laugarren Erabiltzailea | 40000000 | 0 | 0 | 0 | 0 | 4 |

4.16 Irudia: Komunitateak eta erabiltzaileak datu-basean.

4.4.3 Logeaturiko erabiltzaileen ekintzak

Behin komunitateak eta kontuak ondo sortu direla, goazen logeaturiko erabiltzaile batek egin ditzakeen zenbait probekin.

Jokalariak ikusi eta boskotea aukeratu

Erabiltzaile guztien kontuetan sartu eta bakoitzari boskote bat jarriko diot. Gainera, *erabiltzailea1*, *erabiltzailea3* eta *erabiltzailea4*-ren boskoteak publikoki publikatuko ditut.

| Opciones | | IDErabiltzailea | 1 | idJokalaria |
|--------------------------|-----------------------|-----------------|---------|-------------|
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 200746 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 200794 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 202325 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 203078 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 203497 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 1626170 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 1626209 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 1627746 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 1628373 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 1628404 | |
| <input type="checkbox"/> | Editar Copiar Borrarr | 4 | 101108 | |

4.17 Irudia: Erabiltzailearen jokalariaik datu-basean.

| Opciones | | IDErabiltzailea | mota | egunda | idJokalaria1 | idJokalaria2 | idJokalaria3 | idJokalaria4 | idJokalaria5 |
|--------------------------|-----------------------|-----------------|------|------------|--------------|--------------|--------------|--------------|--------------|
| <input type="checkbox"/> | Editar Copiar Borrarr | 3 | 0 | 2019-06-04 | 1628373 | 203078 | 200746 | 200794 | 203497 |
| <input type="checkbox"/> | Editar Copiar Borrarr | 4 | 0 | 2019-06-04 | 101108 | 1628401 | 203507 | 203953 | 202700 |
| <input type="checkbox"/> | Editar Copiar Borrarr | 4 | 2 | 2019-06-04 | 101108 | 1628401 | 203507 | 203953 | 202700 |
| <input type="checkbox"/> | Editar Copiar Borrarr | 5 | 0 | 2019-06-04 | 202689 | 202704 | 203926 | 203954 | 201599 |
| <input type="checkbox"/> | Editar Copiar Borrarr | 6 | 0 | 2019-06-04 | 203078 | 1628404 | 200794 | 200746 | 203497 |
| <input type="checkbox"/> | Editar Copiar Borrarr | 7 | 0 | 2019-06-04 | 1628401 | 203079 | 203937 | 1628470 | 202700 |
| <input type="checkbox"/> | Editar Copiar Borrarr | 7 | 2 | 2019-06-04 | 1628401 | 203079 | 203937 | 1628470 | 202700 |
| <input type="checkbox"/> | Editar Copiar Borrarr | 8 | 0 | 2019-06-04 | 203487 | 1629126 | 203926 | 203954 | 203086 |
| <input type="checkbox"/> | Editar Copiar Borrarr | 8 | 2 | 2019-06-04 | 203487 | 1629126 | 203926 | 203954 | 203086 |

4.18 Irudia: Alineazioak datu-basean.

| Komunitatea | | | | | |
|---------------------|--|--|--|--|--|
| erabiltzaile2 | | | | | |
| | | | | | |
| erabiltzaile4 | | | | | |
| | | | | | |
| Alineazio Publikoak | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

4.19 Irudia: Erabiltzailea4-k ikusten dituen alineazio publikoak.

Merkatuak

Merkatuan dauden jokalariei hainbat eskaintza egingo dizkiet, eta ondoren hauek datu-basean gorde direla konprobatu. Aldi berean, beste hainbat jokalaria merkatuan jarriko ditut eskaintza automatikoa iristen dela konprobatzeko.

| Erabiltzailea | Jokalaria | Merkatuko Jokalaria | Eskaintza |
|------------------------------|----------------|---------------------|-----------|
| erabiltzailea4 (id=8) | Kenneth Faried | 9 | 15000000 |
| | Anthony Davis | 15 | 28000000 |
| root (id=1) | Courtney Lee | 17 | 11641091 |

4.2 Taula: Egin beharko diren eskaintzak.



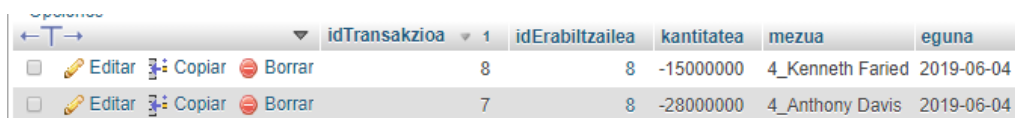
| | idEskaintza | idMerkatuko_Jokalaria | idErabiltzailea | eskaintza |
|---|-------------|-----------------------|-----------------|-----------|
| <input type="checkbox"/> Editar Copiar Borrar | 1 | 9 | 8 | 15000000 |
| <input type="checkbox"/> Editar Copiar Borrar | 2 | 15 | 8 | 28000000 |
| <input type="checkbox"/> Editar Copiar Borrar | 3 | 17 | 1 | 11641091 |

4.20 Irudia: Eskaintzak datu-basean.

Orain **administratzaile orokorraren** rola jarraituz, merkatuak eguneratuko ditugu. Ondorioz, komunitateen merkatuak eguneratzeaz gain, onarturiko eskaintzak tramitatu beharko ditu.

Lehenengo biak automatikoki tramitatuko ditu sistema berak jarritako jokalariai direlako eta jasotako eskaintzarik handienak direlako. Bigarrena ez dugu onartuko eta ondorioz ezingo du tramitatu.

Dena ondo joan dela jakiteko, transakzioak kontsultatuko ditugu datu basean.



| | idTransakzioa | idErabiltzailea | kantitatea | mezua | eguna |
|---|---------------|-----------------|------------|------------------|------------|
| <input type="checkbox"/> Editar Copiar Borrar | 8 | 8 | -15000000 | 4_Kenneth Faried | 2019-06-04 |
| <input type="checkbox"/> Editar Copiar Borrar | 7 | 8 | -28000000 | 4_Anthony Davis | 2019-06-04 |

4.21 Irudia: Transakzioak datu-basean.

Puntuak eguneratu

Erabiltzaileen puntuak lortzeko, lehendabizi alineazioak finkatu beharko ditugu. Horretarako *administratzaile orokorraren* interfazean sartu eta finkatu beharko ditugu. Horrek datu-basean lerro berriak sortuko ditu:

| | idErabiltzailea | mota | egunda | idJokalaria1 | idJokalaria2 | idJokalaria3 | idJokalaria4 | idJokalaria5 |
|---|-----------------|------|------------|--------------|--------------|--------------|--------------|--------------|
| <input type="checkbox"/> Editar Copiar Borrar | 3 | 0 | 2019-06-04 | 1628373 | 203078 | 200746 | 200794 | 203497 |
| <input type="checkbox"/> Editar Copiar Borrar | 3 | 1 | 2019-06-04 | 1628373 | 203078 | 200746 | 200794 | 203497 |
| <input type="checkbox"/> Editar Copiar Borrar | 4 | 0 | 2019-06-04 | 101108 | 1628401 | 203507 | 203953 | 202700 |
| <input type="checkbox"/> Editar Copiar Borrar | 4 | 1 | 2019-06-04 | 101108 | 1628401 | 203507 | 203953 | 202700 |
| <input type="checkbox"/> Editar Copiar Borrar | 4 | 2 | 2019-06-04 | 101108 | 1628401 | 203507 | 203953 | 202700 |
| <input type="checkbox"/> Editar Copiar Borrar | 5 | 0 | 2019-06-04 | 202689 | 202704 | 203926 | 203954 | 201599 |
| <input type="checkbox"/> Editar Copiar Borrar | 5 | 1 | 2019-06-04 | 202689 | 202704 | 203926 | 203954 | 201599 |
| <input type="checkbox"/> Editar Copiar Borrar | 6 | 0 | 2019-06-04 | 203078 | 1628404 | 200794 | 200746 | 203497 |

4.22 Irudia: Alineazioak datu-basean.

Behin hori eginda, jardunaldi bat aukeratu dugu eta puntuak eguneratu. Proba gehiago egiteko, lehenengo 5 jardunaldien puntuazioak eguneratuko ditut.

| | idErabiltzailea | idJardunaldia | puntuak |
|---|-----------------|---------------|---------|
| <input type="checkbox"/> Editar Copiar Borrar | 3 | 1 | 18 |
| <input type="checkbox"/> Editar Copiar Borrar | 3 | 2 | 54 |
| <input type="checkbox"/> Editar Copiar Borrar | 3 | 4 | 45 |
| <input type="checkbox"/> Editar Copiar Borrar | 3 | 5 | 52 |
| <input type="checkbox"/> Editar Copiar Borrar | 4 | 1 | 47 |
| <input type="checkbox"/> Editar Copiar Borrar | 4 | 2 | 88 |
| <input type="checkbox"/> Editar Copiar Borrar | 4 | 4 | 109 |

4.23 Irudia: Erabiltzaileen puntuak datu-basean

| | idJokalaria | idJardunaldia | puntuak |
|---|-------------|---------------|---------|
| <input type="checkbox"/> Editar Copiar Borrar | 1629244 | 2 | 0 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629244 | 1 | 7 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629234 | 5 | 0 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629234 | 4 | 2 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629234 | 2 | 0 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629234 | 1 | 0 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629185 | 2 | 5 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629185 | 1 | 0 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629164 | 2 | 0 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629164 | 1 | 2 |
| <input type="checkbox"/> Editar Copiar Borrar | 1629151 | 5 | 0 |

4.24 Irudia: Jokalarien puntuak datu-basean

Ondorioz, jardunaldi horretan komunitate bakoitzean lehen sailkaturiko erabiltzaileek sari berezia jaso beharko dute. Hori datu-basean transakzioetan gordeta gelditzen da.

| | idTransakzioa | idErabiltzailea | kantitatea | mezua | eguna |
|---|---------------|-----------------|------------|-------|------------|
| <input type="checkbox"/> Editar Copiar Borrar | 32 | 5 | 1500000 | 3_ | 2019-06-04 |
| <input type="checkbox"/> Editar Copiar Borrar | 31 | 3 | 2500000 | 2_ | 2019-06-04 |
| <input type="checkbox"/> Editar Copiar Borrar | 30 | 4 | 5000000 | 1_ | 2019-06-04 |
| <input type="checkbox"/> Editar Copiar Borrar | 29 | 8 | 1500000 | 3_ | 2019-06-04 |
| <input type="checkbox"/> Editar Copiar Borrar | 28 | 7 | 4000000 | 2_ | 2019-06-04 |
| <input type="checkbox"/> Editar Copiar Borrar | 27 | 6 | 8000000 | 1_ | 2019-06-04 |
| <input type="checkbox"/> Editar Copiar Borrar | 26 | 3 | 1500000 | 3_ | 2019-06-04 |
| <input type="checkbox"/> Editar Copiar Borrar | 25 | 5 | 2500000 | 2_ | 2019-06-04 |

4.25 Irudia: Transakzioak datu-basean.

5. KAPITULUA

Jarraipena eta Kontrola

Atal honetan proiektua garatzeko erabili dudan denbora erreala eta egindako plangintza-
ren arteko ezberdintasunak aztertuko dira, desbideraketan zergatia azalduz.

Ondorioz, Gantt diagrama eta mugarriak berriz definitzeko beharra izan dugu.

Gainera, hasieran finkaturiko irismen-maila eta kalitate-plana bete den aztertu beharko
dugu, baita proiektuan zehar edukitako arriskuen aurrean hartutako erabakiak.

5.1 Desbideraketak

Proiektu handi guztietan gertatzen den bezala, hasierako plangintzaren eta benetan gerta-
tzen denaren artean aldaketak egotea oso ohikoa den gauza bat da. Hori argiago ikusteko,
hasierako orduen estimazioen taulari denbora errealak gehitu dizkiot, ondoren ikusi dai-
tekeen taulan.

Ohikoa izaten den bezala, hasieran estimatuak ziren ordu kopurua motz gelditu da. Hala
ere, plangintza nahiko txukuna eta erreala zenez, desbideraketak ez dira oso handiak izan
eta proiektua modu egokian aurrera eramateko aukera izan dut.

| Kodea | Ataza | Estimazioa | Errealak | Desbideraketa | |
|-----------------------------|---------------------------------|-----------------|-----------------|-----------------|--------------|
| Kudeaketa | | | | | |
| 1.1 | Plangintza | 10 ordu | 15 ordu | +5 ordu | +%50 |
| 1.2 | Jarraipena | 10 ordu | 8 ordu | -2 ordu | -%20 |
| 1.3 | Kontrola | 5 ordu | 7 ordu | +2 ordu | +%40 |
| | | 25 ordu | 30 ordu | +5 ordu | +%20 |
| Ikerkuntza | | | | | |
| 2.1 | Testuingurua | 5 ordu | 1 ordu | -4 ordu | -%80 |
| 2.2 | Web-zerbitzuak | 5 ordu | 1 ordu | -4 ordu | -%80 |
| 2.3 | Zerbitzariaren arkitektura | 5 ordu | 5 ordu | 0 ordu | %0 |
| 2.4 | Boorstrap interfazeak | 5 ordu | 1 ordu | -4 ordu | -%80 |
| 2.5 | Android garapena | 5 ordu | 1 ordu | -4 ordu | -%80 |
| | | 25 ordu | 9 ordu | -16 ordu | -%64 |
| Analisi eta diseinua | | | | | |
| 3.1 | Erabilpen kasuak | 10 ordu | 15 ordu | +5 ordu | +%50 |
| 3.2 | Sekuentzia diagramak | 35 ordu | 50 ordu | +15 ordu | +%43 |
| 3.3 | Interfazeen diseinua | 5 ordu | 5 ordu | 0 ordu | %0 |
| | | 50 ordu | 70 ordu | +20 ordu | +%40 |
| Garapena | | | | | |
| 4.1 | Zerbitzaria | 50 ordu | 40 ordu | -10 ordu | -%20 |
| 4.2 | Web-aplikazioa | 35 ordu | 50 ordu | +15 ordu | +%43 |
| 4.3 | Android aplikazioa | 40 ordu | 30 ordu | -10 ordu | -%25 |
| | | 125 ordu | 120 ordu | -5 ordu | -%4 |
| Egiaztapena | | | | | |
| 5.1 | Produktuaren probak | 4 ordu | 5 ordu | +1 ordu | +%25 |
| 5.2 | Emaitzak aztertu | 1 ordu | 5 ordu | +4 ordu | +%400 |
| 5.3 | Errore identifikatu eta zuzendu | 5 ordu | 10 ordu | +5 ordu | +%100 |
| | | 10 ordu | 20 ordu | +10 ordu | +%100 |
| Dokumentazioa | | | | | |
| 6.1 | Memoria | 45 ordu | 60 ordu | +15 ordu | +%33 |
| 6.2 | Eranskinak | 2 ordu | 2 ordu | 0 ordu | %0 |
| 6.3 | Erabilpen kasuen gida | 3 ordu | 3 ordu | 0 ordu | %0 |
| 6.4 | Aurkezpena | 5 ordu | 5 ordu | 0 ordu | %0 |
| | | 55 ordu | 70 ordu | +15 ordu | +%27 |
| Guztira | | 290 ordu | 319 ordu | +29 ordu | +%10 |

5.1 Taula: Denbora estimazioaren eta errealaren arteko desbideraketa.

5.1.1 Kudeaketa

Hasieran estimaturik zeuden baino bost ordu gehiagoko lan-karga izan zen. Bereziki, desbiderapen handiena plangintzan zegoen. Espero nuena baino astunagoa egin zitzaidan hori modu egokian garatzea. Jarraipena eta kontrola bat bestearen bidez konpentsatuak gelditu dira.

5.1.2 Ikerkuntza

Denborarekin arazorik eduki ez izanaren erredun nagusia. Hasierako estimazioa egiterakoan, beharrezkoak ziren baino ordu gehiago jarri nituen, eta egia esan, banekien norma-lena sobran geratzea izango zela.

Testuingurua ezaguna zen, aurretik beste aplikazio bat garatua nuelako. Web-zerbitzuak *Software Ingeniaritza I* irakasgaiaren ikusitakoarekin nahikoa izan zen, *Boorstrap* eta *Android Studio*-ren inguruan nituen aurreko lezioekin gauza bera gertatu zen.

Zerbitzuaren arkitektura aztertze horretan pasa nuen denbora gehien, baina ez hasieran espero nuena baino gehiago.

Ondorioz, estimatutako 25 orduetatik 9 ordu errealetara pasa nintzen, 16 ordu aurreztuz, %64-eko desbideraketa suposatuz.

5.1.3 Analisi eta diseinua

Hemen ere atzerapen nahiko handi bat jasan nuen. Erabilpen kasuak definitzerako orduan, gertaera fluxu guztiak azaldu beharra ikusi nuen. Ondorio zuzen bezala, erabilpen-kasu bakoitzaren sekuentzia diagrama hasieran uste nuena baino luzeagoa izan zen.

Gainera, diseinaturiko hainbat sekuentzia diagrama berregin behar izan nituen, gaizki planteatu nituelako.

5.1.4 Garapena

Ordu gehien estimatu nituen atala, eta nahiko zuzen egon nintzen. Desbideraketa positibo txiki bat izan dut, eta espero nuena baino azkarrago garatu dut produktua.

Zergatia? *Android* garapena oso azkar egiteko aukera izan nuen aurretik esperientzia handia nuelako, eta zerbitzariaren lan-karga uste nuena baino txikiagoa zen.

Aldi berean, web-bezeroaren garapena nahiko astuna izan zen, etengabe arazo txikiak de-
tektatuz. Gainera, ondoren aipatuko dudan moduan, fitxategi bat galdu nuen eta berriz
idatzi beharra izan.

5.1.5 Egiaztapena

Atal honetan ere desbiderapen nahiko handiak izan ditut, baina posible da hasieran eza-
rritako ordu estimazioa erreala ez izatea. Normala da ohitura faltagatik horrelako gauzak
kontrolpean ez edukitzea, eta horregatik garrantzitsua da hurrengorako horrelako gauzak
baloratzea.

Desbiderapen handiena jasan duen atala izan da, %100. Hasieran estimatuak ziren 10
orduak, 20 ordutan bihurtu izan dira.

5.1.6 Dokumentazioa

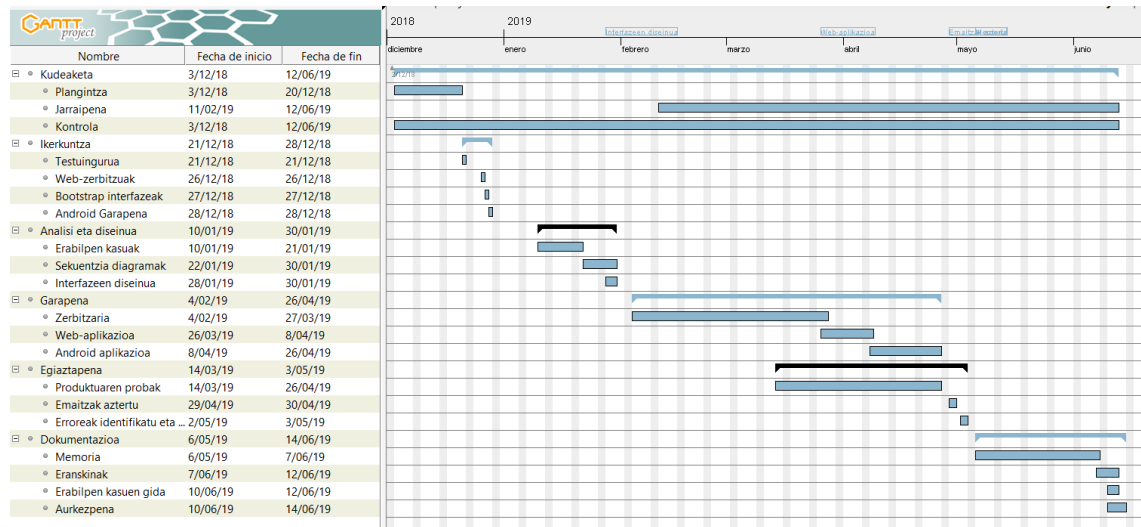
Azkenengo atal honetan izan dudan desbiderapen handiena memoriaren garapena izan da.
Gradu guztian zehar \LaTeX erabili izan dut txosten gehienak egiteko eta horregatik auke-
ratu nuen, baina egia da zenbait gauza egiteko motelagoa dela.

Espero nuen baino taula gehiago egin ditut, eta horrek denbora galera oso handia suposa-
tzen du. Azkenean memoria aurrera eramateko espero nuena baino 15 ordu gehiago sartu
behar izan ditut, %33-ko desbiderapena suposatuz.

5.2 Gantt diagrama

Atazak burutzeko desbideraketen ondorioz, hasieran planteaturiko Gantt diagramak ere
aldaketa txiki batzuk izan ditu.

Egia da, hasierako eta bukaerako Gantt diagramak oso antzekoak direla. Hori lortzeko,
espero ziren baino ordu gehiago eskieni dizkiot zenbait egunetan. Modu horretan epeak
ahal ziren moduan errespetatzeko aukera izan dut.



5.1 Irudia: Gantt diagramaren egoera erreala.

5.3 Mugarriak

Gantt diagraman ikusi den bezala, atzerapen batzuk egon dira zenbait atazekin. Horrekin, hurrengo atazetan ondorio zuzena izan du hauek atzeratuz.

Hau izan zen hasieran jarritako mugarriak eta bukaeran bete diren datak:

| | Hasierakoa | Erreala |
|----------------------------|------------|------------|
| Plangintza garatu | 2018-12-14 | 2018-12-20 |
| Ikerkuntza lana | 2018-12-28 | 2018-12-28 |
| Eskakizunen bilketa | 2019-01-16 | 2019-01-16 |
| Diseinua | 2019-01-30 | 2019-01-30 |
| Inplementazioa | 2019-04-19 | 2019-04-26 |
| Memoria | 2019-05-31 | 2019-06-07 |
| Aurkezpena | 2019-06-07 | 2019-06-14 |

5.2 Taula: Mugarrien eguneraketa.

5.4 Kalitate-plana

Proiektuaren plangintza egiterako orduan, proiektuak orokorrean guk nahi genuen kalitatea zuela ziurtatzeko, hainbat adierazle finkatu ziren.

- **Zerbitzariak funtzioak prozesatzeko abiadura.**

- *Deskribapena:* Zerbitzariak funtzioak arintasunez prozesatzea. Modu horretan bezero aplikazioak itxaron beharreko denbora txikiagoa izango da, erabiltzailearen esperientzia hobetuz.
- *Neurtzeko modua:* Funtzioak prozesatzeko beharrezkoa duen denbora.
- *Maximo onargarria:* 5 segundo
- *Helburua:* Segundo bat edo gutxiago
- **Lorturiko emaitza:** Funtzio batzuek beste batzuk baino gehiago behar dute, baina guztiak segundo batetik behera. Helburua lortua.

- **Bezeroen fidagarritasuna.**

- *Deskribapena:* Bezeroaren aldetik hainbat erabilpen kasu egingo ditu bere saioetan, funtzio askori deituz eta emaitzak pantailaratu. Helburua: denak ondo funtzionatzea.
- *Neurtzeko modua:* Gertatzen diren errore kopurua.
- *Minimo onargarria:* 100 deietatik 95ek ondo funtzionatzea.
- *Helburua:* Erabiltzaile guztiek saio guztietan arazorik ez gertatzea.
- **Lorturiko emaitza:** 5 lagun aldi berean probak egiten ibili ondoren, ez da inolako arazorik egon. Emaitza espero zena baino hobeagoa.

- **Kodearen ulergarritasuna.**

- *Deskribapena:* Proiektu bat garatzerakoan, kontuan eduki behar da kodearen inplementazioaren argitasuna. Bereziki informatika arloan oso garrantzitsua den atal bat da, beste pertsona batek kodea irakurri edo aldatu behar baldin badu etorkizunean.

- *Neurtzeko modua*: Aldagaien izenak, anotazioak, espazioak kontrolatzea...
- *Minimo onargarria*: Funtzio guztien funtzionamendua anotazioen bidez argitzea.
- *Helburua*: Minimoaz gain, aldagaien izenak argiak izatea, lerro saltoak errespetatzea, ezker marjina ondo eramatea... hau da, itxura elegantea izatea.
- **Lorturiko emaitza**: Zerbitzariaren funtzio guztiak komentatuak. Sarrerako eta irteerako aldagaiak zer diren eta funtzionamendua modu argian azalduta.

- **Interfazearen diseinua.**

- *Deskribapena*: Interfazea edozein gailutan funtzionala eta argia izatea, hau da, moldagarria. Erabilerraza egitea oso garrantzitsua da.
- *Neurtzeko modua*: Interfazeak moldaturik dauden gailu kopurua, bilaketak modu argian azaltzea eta klik kopurua gutxitzea.
- *Minimo onargarria*: Interfazeak gutxienez ordenagailuetarako pentsatuak egotea.
- *Helburua*: Ordenagailuez gain mugikorretan eroso izatea eta edozein erabilpen-kasu 3 klik-etara egotea.
- **Lorturiko emaitza**: Interfaze sinple baina oso indartsuak. Erabilpen kasu guztiak menu bakar batetik eskuragarriak. Interfazeek erabilpen-kasu bat baino gehiago irudikatu modu oso eroso batean.

- **Memoriaren kalitatea.**

- *Deskribapena*: Formatu egokia, eduki argiak eta modu osatuan izatea.
- *Neurtzeko modua*: Proiektu guztia dokumentatzea prozesu guztia azalduz, modu argi eta antolatu batean.
- *Minimo onargarria*: Memoriak aurretik ezarritako oinarritzko egitura izatea.
- *Helburua*: Edukiak ondo antolatuta egoteaz gain, prozesu guztia modu oso batean azaltzea, beharrezkoa den media erabiliz (argazkiak, grafikoak, taulak ...).

- **Lorturiko emaitza:** Ezarritako minimo onargarriak betetzen dituen memoria, eduki guztia oso ondo antolatua, taulak, argazkiak, grafikoak erabili... Emaitza borobila.

5.5 Arriskuak

Proiektuaren garapenaren zehar ez dut arazo larririk eduki, baina bai pare-bat ezusteko.

Esan bezala, kodearen bertsio-kontrola edukitzeko eta aldi berean ez galtzeko, *GitHub* erabiltzea erabaki nuen. Baina arazo bat eduki ondoren, *php* fitxategi bat galdu nuen. Une horretatik aurrera, *Drive* erabiltzen hasi nintzen, interfaze aldetik erosoagoa zelako eta telefono mugikorretik modu erosoan kontrola eramateko aukera nuelako.

Bestalde, espero ez nuen arazo bat gertatu zitzaidan. Lagunekin hainbat proba saio egin ondoren, datu-basearekin arazo bat eduki eta erabiltzeen datuak galdu ziren. Zorionez, momentu horretan garrantzi handirik ez zuen arazo bat izan zen, proba garaia zelako.

Baina horren esker, lehen baloratzen ez nuen arrisku bat ikusi nuen, erabiltzeen datuak babestearena. Hori dela eta, egun horretarik aurrera, datu-basearen segurtasun-kopiak egunean eramatea erabaki nuen, etorkizunean arazorik gertatzen baldin bada, soluzio azkar bat emateko.

6. KAPITULUA

Ondorioak

Proiektuari itxiera bat emateko, gogora dezagun helburu nagusia: NBAren inguruan sare-sozial bat antolatzea, zerbitzari bakar bat eta bezero mota desberdinez osaturiko ekosistema batean.

Gradu Amaierako Proiektua hau Software Ingeniaritza espezialitatekoa denez, arreta berezia eskaini diogu Eskakizunen Bilketari eta Analisi-Diseinu atalei.

Bukatzeko, prozesu guzti honetan zehar lortu ditudan ondorioak azalduko ditut eta etorkizunerako hainbat ideia planteatu.

6.1 Ikasitako lezioak

Horrelako proiektu handi eta konplexu bat aurrera eramatea lortu ondoren, badira zenbait gauza oso garrantzitsu ikasten direla. Egia da, proiektu hau planteatu aurretik, nire kabuz beste hainbat aplikazio garatu nituela, baina inolako prozesurik jarraitu gabe; eta diferentzia oso handia da.

Proiektuaren bizi-ziklo guztian lezio asko ikasi ditudan arren, hauek dira esanguratsuenak:

- **Irismena finkatzea:** Produktua nondik noraino joango den. Ez gehiago, ez gutxiago. Garatzaileak bete nahi dituen helburu guztiak finkatu, horiek modu egoki eta antolatu batean garatu ahal izateko.

- **Mugarriak ezartzea:** Proiektu luzeetan, oso ohikoa da lanaren konstantzia oso aldakorra izatea. Hau da, egun batzuetan indar eta gogo askorekin hartzea, eta alderantziz. Hori saihesteko eta lanaren konstantzia minimo bat mantentze aldera, mugarriak finkatzea eta batez ere hauek errespetatzea ideia bikaina dela iruditzen zait. Hurrengo proiektuetan erabiliko dudak teknika bat izango da, dudarik gabe.
- **Domeinua ondo diseinatzea:** Eskakizunaren bilketa eta diseinua sartu daitezke zaku berean, baina benetan urrats garrantzitsuena domeinua ondo finkatzea da. Domeinu on bat finkaturik, ondoren ataza guztiak modu askoz errazago baten egiten dira. Gainera, edozeini zure proiektua kokatzeko oso baliagarria den materiala izango da.
- **Kodearen txukuntasuna:** Gradu guztian zehar etengabe adierazi digute kodea txukun mantentzeren garrantzia, eta horrelako proiektu handietan garbi ikusten da zergatia. Posible da hilabete ukitu ez duzun kode zatiak funtzionatzeari uztea, eta zergatia topatu behar izatea. Kodea txukun baldin badago, lana oso erraza izango da, eta alderantziz.

6.2 Etorkizunerako aukerak

Atal hau bi zatitan banatu dezakegu. Alde batetik produktua berak izango duen etorkizuna, eta bestalde garatzaileak (nik).

Lehendabizi produktuaren aukerak aztertuko ditugu. Orain, denboraldia bukatu dago, beraz aplikazioak ez du funtzionalitate gehiegirik izango.

Zein da helburua? Udan interfazeak hobetzen jarraitzea datorren urteko denboraldi hasieran erabiltzaile berriak erakartzeko. Produktu honek bizi-ziklo periodikoa izango du, hau da, urria erdialdetik ekaina hasierara, erabiltzaile-kopuru handiena beti hasierako hilabetean izanik.

Interesgarri izango zen hizkuntza eta plataforma gehiagotara moldatzea (bereziki *WindowsPhone* eta *iOS*).

Eta noski, aipatu ez dudak urrats garrantzitsuena, produktua eskuragarri jartzea. Horretarako zerbitzari bat alokatu beharko nuke eta *Android*eko aplikazioa *Google Play* merkatura igo.

Komentatu ez dudak baina oso garrantzitsua den atal bat baztertu dugu orain arte, erabiltzailearen segurtasuna bermatzea. Aplikazioa publikoa egin aurretik, datu-baseko pasahi-

tzak zifratuak egon beharko dira. Modu horretan datu-baseak kanpo eraso bat jasotzen baldin badu, erabiltzaileen pasahitzek ez dute arriskurik izango. Horrez gain, interesantea izango litzateke termino batzuk garatzea eta erabiltzaileak kontua sortzean hauek onartu behar izatea.

Aplikazioarekin bukatzeko, ondo egongo zen erabiltzaileek eta pasahitzentzako filtro txiki bat egotea. Funtzioak prestaturik daude, beraz inplementatzearekin nahikoa izango zen.

Bestalde, garatzaile moduan proiektu honek hainbat aukera eskaintzen dizkit.

Horien artean, kirola aldatu eta beste aplikazio bat publikatzea. Egia da hainbat gauza eraldatu beharko nituela, baina arkitektura guztia berrerabiltzeko aukera izan nuen: interfazeak, teknologia guztiak... API berri bat lortu ezker, prozesua oso erraza izango zen. Gainera, nahiz eta prozesua ia guztia automatizatua egon, aplikazioaren funtzionamendua egokia izateko administratzaile orokor bat beharko du. Ondorioz, hurrengo urtean egunero 5 minutu eskieni beharko dizkiot administratzaile orokorraren lana egiteko.

Eranskinak

A. ERANSKINA

Bilera-aktak

1. Bilera-akta

Deialdi data: 2019ko Urtarrilaren 22a

Deialdi ordua: 12:30

Bertaratuak:

- *Jon Guilló Rodríguez* (ikaslea)
- *Jon Iturrioz* (irakaslea)

Gai-zerrenda:

1. Proiektuaren deskribapena irakurri
2. Domeinua aztertu
3. Zehazturiko erabilpen-kasu guztiak ikusi

Hartutako erabakiak:

1. Domeinuak dituen diseinu akatsak zuzentzea.
2. Erabilpen kasuak ontzat eman

Hartutako eginkizunak:

1. Erabilpen kasuen gertaera fluxuak zehaztu
2. Aurrekoa lortzeko, erabiliko diren interfazeak nolakoak diren pentsatzen hasi

2. Bilera-akta

Deialdi data: 2019ko Urtarrilaren 30a

Deialdi ordua: 12:30

Bertaratuak:

- *Jon Guilló Rodriguez* (ikaslea)
- *Jon Iturrioz* (irakaslea)

Gai-zerrenda:

1. Gertaera fluxuak aztertu
2. Interfazeen diseinua nolakoa izango den oso gaietik ikusi

Hartutako erabakiak:

1. Lanak hiru mailako arkitektura erabiliko duela zehaztu.
2. Erabiliko diren teknologiak aztertu, eta *datu-base erlazioanala* erabiltzea erabaki. *MySQL*

Hartutako eginkizunak:

1. Merkatua aztertu (antzeko aplikazio eta web-orriek duten erabilpen kasuak ikusi eta konparaketa txiki bat egin). Ikerketa lana.
2. Gure *facade* API-ak izango dituen funtzioak pentsatzen hasi.
3. Erabiliko dudan datu basearen diseinua pentsatzen hasi (domeinuaren eredutik jarraituz).

3. Bilera-akta

Deialdi data: 2019ko Martxoaren 13a

Deialdi ordua: 12:30

Bertaratuak:

- *Jon Guilló Rodríguez* (ikaslea)
- *Jon Iturrioz* (irakaslea)

Gai-zerrenda:

1. API-ak izango dituen funtzioen aukeraketa aztertu.
2. Diseinaturiko datu-basea aztertu.

Hartutako erabakiak:

1. Datu-basearen teknologia: *Datu-base erlazionala MySQL*
2. Zerbitzariaren teknologia zehaztuz:
 - a. *Programazio-lengoaia: Java*
 - b. *Datu-basearekin konexioa ezartzeko: Hibernate*

Hartutako eginkizunak:

1. Sekuentzia diagramak diseinatzen hasi
2. Diagramak diseinatu ahala zerbitzari funtzioak inplementatzen hasi

4. Bilera-akta

Deialdi data: 2019ko Martxoaren 27a

Deialdi ordua: 12:30

Bertaratuak:

- *Jon Guilló Rodríguez* (ikaslea)
- *Jon Iturrioz* (irakaslea)

Gai-zerrenda:

1. Sekuentzia diagramak aztertu
2. Zerbitzari funtzioen implementazioa ikusi eta zuzendu

Hartutako erabakiak:

1. *Hibernate*-en konfigurazioan aldaketak egin sistema azkartzeko
2. Sekuentzia diagramak ontzat eman.

Hartutako eginkizunak:

1. *Web* eta *Android* aplikazioen garapenekin hasi.

5. Bilera-akta

Deialdi data: 2019ko Maiatzaren 16a

Deialdi ordua: 12:30

Bertaratuak:

- *Jon Guilló Rodriguez* (ikaslea)
- *Jon Iturrioz* (irakaslea)

Gai-zerrenda:

1. Produktu finala probatu:
 - a. Zerbitzariaren erantzun abiadura
 - b. Web-bezeroa
 - c. Android-bezeroa
2. Interfazeak baloratu
3. Memoriaren egitura

Hartutako erabakiak:

1. Produktua ontzat eman.
2. Interfazea interesgarria eta praktikoa.
3. Memoriaren egitura zehaztu.

Hartutako eginkizunak:

1. Probak dokumentatu.
2. Memoriarekin jarraitu.

B. ERANSKINA

Erabilitako materiala

Kapitulu honetan aplikazioaren garapenean erabili dudan kode guztia dago (erabilitako irudi, liburutegi..). Esteka honen bitartez *GitHub*etik repositorio guztia deskargatzeko aukera dago:

https://github.com/jongui14/NBA_SareSoziala

Kode guztia hiru azpi ataletan banatua dago:

- Android bezeroa
- Web bezeroa
- Zerbitzaria

Android bezeroaren karpeta, fitxategi kodeekin batera *.apk* exekutagarri bat dago lehenengo karpeta probak egitea nahi baldin bada.

Ondoren Web bezeroa dugu eta Bertan fitxategi guztiak azpi-karpeta gordeta daude. Modu erraz batean hasieratzeko, *index.php* bat kokatu dut, eta horren bidez dagokion web-orrira birbideratuko du erabiltzailea.

Bukatzeko zerbitzaria. Zerbitzariak datu-base bat behar duenez, *.sql* bat kokatu dut taulak sortzeko eta probak egiteko datu gutxi batzuekin. Horrez gain, zerbitzaria beraren kodea eta beharrezkoak diren liburutegiak.

C. ERANSKINA

Erabilera gida

Atal honekin hasi aurretik, interfazeak oso sinpleak eta intuitiboak direla argitu beharra dago. Gida hau erabili gabe ere edozein erabiltzaileek aplikazioa arazorik gabe erabiltzeko aukera izango du.

Erabilpen kasu guztiak ez egiteko, soilik konplexuenak azalduko ditugu.

C.1 Komunitate eta administratzailea sortu

Komunitatea sortu ahal izateko, erabiltzailea ezingo da logeatu egon, eta ondorioz beraren rolari dagokion menua azalduko zaio. Bertan, *Komunitate berria* aukeratuko du.



C.1 Irudia: Logeatu gabeko erabiltzailearen menua.

Komunitate berria sortzeko hutsune guztiak ondo bete beharko dira. Une horretan, *Komunitatea sortu* botoia aukeragarri jarriko da. Sartu beharreko daturen bat gaizki baldin badago, ezin izango da aurrera jarraitu

(a) Web

(b) Android

C.2 Irudia: Komunitate berria sortu.

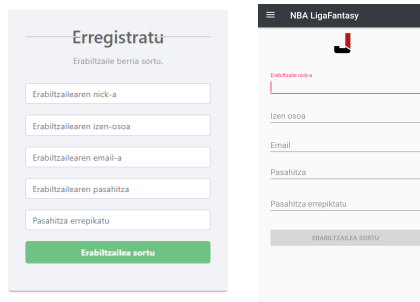
Datu guztiak ondo jarri ondoren, botoiaren kolorea aldatu egingo da, eta ondorioz aurrera jarraitzeko aukera eskainiko digu.

(a) Web

(b) Android

C.3 Irudia: Komunitate berria sortu.

Ondoren erabiltzaile (administratzailea) sortu beharko dugu, eta aurreko kasuan bezala, datuak modu egokian sartu arte ezingo dugu aurrera jarraitu.

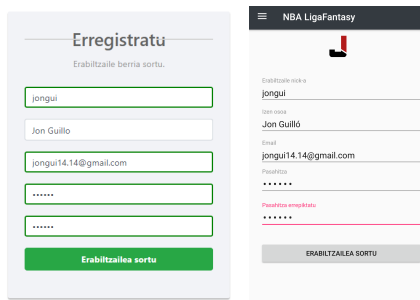


(a) Web

(b) Android

C.4 Irudia: Erabiltzaile berria sortu.

Datuak ondo sartu ondoren erabiltzailea sortuko dugu.



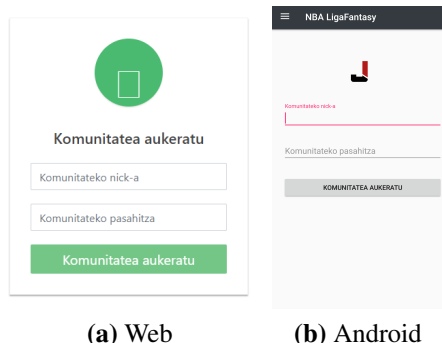
(a) Web

(b) Android

C.5 Irudia: Erabiltzaile berria sortu.

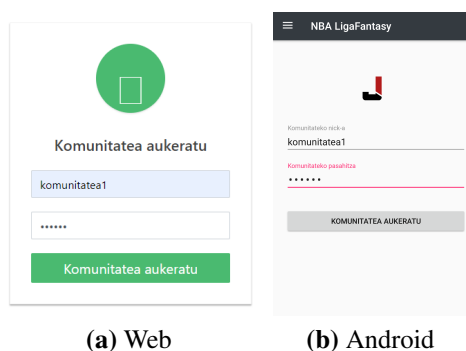
C.2 Erabiltzailea sortu

Kasu honetan, komunitatea aurretik existitu beharko litzateke. Orduan, gure lagunak komunitatearen *nick* eta *pasahitza* zein den esango digu. Orduan, C.1 menuan *Erabiltzaile berria* aukeratuko dugu. Kasu honetan desberdintasun txiki bat dago mugikorretako eta ordenagailuko bertsiorean artean. Batean, botoia blokeaturik egongo da datuak ondo sartu arte eta bestean aldiz botoia beti egongo da klikatzeko moduan baina soilik datuak ondo sartuta dauden kasuan jarraituko du aurrera.



C.6 Irudia: Erabiltzaile berria sortu.

Datuak ondo sartu ondoren, ikusi dezakegu ordenagailuko bertsiotan botoiak aldatuta bat jasaten duela.



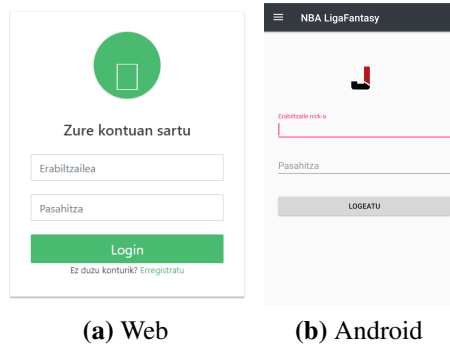
C.7 Irudia: Erabiltzaile berria sortu.

Komunitatea aukeratu ondoren, ikusiko genukeen interfazea C.4-en berdina izango da. Datuak ondo sartu ondoren erabiltzailea sorturik edukiko genukeen.

C.3 Kontura sartu

Erabiltzaile bat sortu ondoren, kontura sartzeko aukera izango dugu. Horretarako [C.1](#) menuan *Kontura sartu* aukeratuko dugu.

Ondoren, datuak ondo sartu beharko ditugu ondorengo interfazean.



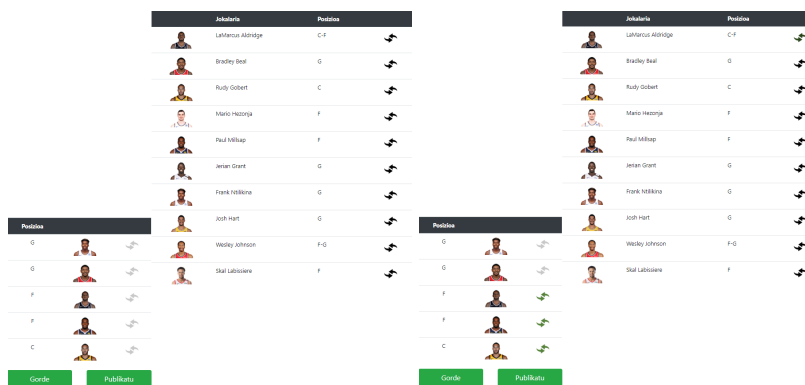
C.8 Irudia: Erabiltzaile kontura sartu.

Behin datuak ondo sartu eta *Login* egin ondoren, ondorengo menua azalduko zaigu:

C.4 Alineazioa aldatu

Logeatutako erabiltzaile guztiek beraien boskotea aldatzeko aukera dute. Horretarako ?? menutik *Jokalariak* -> *Alineazioa* aukeratuko dute. Egia da, web eta mugikorreko bertsioen artean aldaketa nahiko handia dagoela interfaze eta erabiltzeko erarekin.

- Web-bezeroan lehendabizi jokalaria aukeratu behar dugu eta sistemak joka ditzakeen posizioak erakutsiko dizkio.
- Mugikorrean aldiz, posizio bat aukeratzen da eta bertan joka dezaketen jokalariak soilik azaltzen dira.



(a) Web: Alineazioa

(b) Web: Jokalaria hautatu ondoren



(c) Android: Alineazioa

(d) Android: Posizioa hautatu ondoren

Ondoren bi bezeroek eskaintzen digute *Gorde* edo *Publikatzeko* aukera.

Bibliografia

- [1] JBoss Tools (Hibernate) <http://tools.jboss.org/documentation/>.
- [2] Wikipedia: Datu-baseak <https://eu.wikipedia.org/wiki/Datu-base> 2019 maiatzaren 29a.
- [3] Bootstrap: Documentation <https://getbootstrap.com/docs/4.3/getting-started/introduction/> 2019.
- [4] Hibernate: Query <https://howtodoinjava.com/hibernate/hibernate-criteria-queries-tutorial/> 2015.
- [5] José Dimas Luján Castillo *Desarrollo de aplicaciones Android con Android Studio: Conoce Android Studio* 2019ko martxoaren 6a.
- [6] Fabián García: CONSUMIENDO WEB SERVICES CON KSOAP <https://notasdeandroid.wordpress.com/2014/06/30/consumiendo-web-services-con-ksoap/> 2014ko ekainaren 30a.
- [7] Octavia Andreea Anghel: Consuming Web Services with PHP Using NuSOAP <https://www.developer.com/lang/php/consuming-web-services-with-php-using-nusoap.html> 2017ko martxoaren 15a.
- [8] Oracle: Java Endpoint <https://docs.oracle.com/javase/7/docs/api/javax/xml/ws/Endpoint.html>.
- [9] Android Studio: Prozesu eta azpi-prozesuak <https://developer.android.com/guide/components/processes-and-threads.html?hl=es>.
- [10] JSON & Java <https://www.journaldev.com/2315/java-json-example>.