

▪ Gradu Amaierako Proiektua ▪

Software Ingeniaritza

Web-aplikazioak garatzeko gertaeretan oinarritutako teknologien azterketa eta aplikazio praktikoa: GAUTENA erakundearen familientzako web-aplikazioa.

Mikel Agirre

2017 - iraila

Laburpena

Proiektu honek bi helburu nagusi ditu: batetik, web-aplikazioak garatzeko gertaeretan oinarritutako teknologia aztertzea eta, bestetik, paradigma hau jarraitzen duten garapen-inguruneetatik bat aukeratzea web-aplikazio erreal bat garatzeko.

Gertaeretan oinarritutako zerbitzarien kontzeptua web-aplikazioen ikuspegitik aztertuko da lehenik, berori erabiltzen duten teknologien analisia egiteko ondoren. Azterketa honetan, teknologia honek beste batzuekin konparatuz web-aplikazioen garapenerako dakartzan hobekuntza eta berezitasunetan sakonduko da, eta jarraian, teknologia hori erabiltzen duten hainbat *frameworken* arteko konparaketa egingo da, interesgarriena edota abantaila gehien eskaintzen dituenaz zein den aztertuz.

Amaitzeko, eta azterketa biribiltzeko, aukeratutako garapen-ingurunearen aplikazio praktikoa egingo da, aplikazio erreal bat garatuz: GAUTENA Gipuzkoako Autismo Erakundearentzako "Gune Pribatua". Web-aplikazio honen inplementazioaren bitartez, gertaeretan oinarritutako zerbitzariak erabiltzen dituzten garapen-inguruneetako batek praktikan nola funtzionatzen duen aztertu ahal izango da, egindako aukeraketaren egokitasuna frogatzeaz batera.

Laburbilduz, teknologia zehatz bat erabiltzen duen garapen-ingurune "onenaren" bilaketa egingo da, ondoren web-aplikazio erreal baten garapenaren bidez aukeraketaren egokitasuna frogatzeko.

Gaien aurkibidea

Laburpena	iii
Gaien aurkibidea	v
Irudi eta taulen zerrenda.....	vii
Esker onak	ix
1. SARRERA.....	1
1.1 AURREKARIAK.....	2
1.2 PROIEKTUAREN TESTUINGURUA ETA MOTIBAZIOA	2
1.3 PROPOSATUTAKO SOLUZIOA	3
2. KUDEAKETA-PLANA	5
2.1 IRISMENA	6
2.2 PLANGINTZA.....	6
2.2.1 Lan-paketeen deskonposaketa	6
2.2.2 Lan-kargaren estimazioa.....	8
2.2.3 Gantt diagrama.....	9
2.3 ARRISKUEN IDENTIFIKAZIOA.....	10
2.4 INTERESATUAK	10
3. WEB-APLIKAZIOAK GARATZEKO GERTAERETAN OINARRITUTAKO ZERBITZARIAK ERABILTZEN DITUZTEN TEKNOLOGIA ETA INGURUNEEN AZTERKETA.....	13
3.1 GERTAERETAN OINARRITUTAKO PROGRAMAZIOA	14
3.2 WEB-APLIKAZIOEN GARAPENERAKO GERTAERETAN OINARRITUTAKO ZERBITZARIAK ERABILTZEN DITUZTEN TEKNOLOGIAK.....	14
3.3 GARAPEN-INGURUNEAK.....	16
3.3.1 Konparaketa-irizpideak.....	17
3.3.2 Framework-konparaketa	18
3.3.3 Konparaketaren laburpena eta ondorioak	22
4. METEOR	25
5. WEB-APLIKAZIOAREN ANALISI ETA DISEINUA	29
5.1 ANALISIA.....	30
5.1.1 Eskakizunen bilketa	30
5.2 DISEINUA.....	31
5.2.1 Erabilpen-kasuen diagrama	31
5.2.2 Klase-diagrama	38

5.2.3	Sekuentzia-diagramak	42
6.	INPLEMENTAZIOA	47
6.1	APLIKAZIOAREN EGITURA ETA INPLEMENTAZIOAREN NONDIK NORAKOAK	48
6.1.1	Egitura orokorra	48
6.1.2	Erabilitako paketeak	49
6.2	ARAZOAK ETA SOLUZIOAK.....	49
6.3	KODE ZATI ESANGURATSUAK	50
6.4	INTERFAZEAK.....	52
7.	JARRAIPEN ETA KONTROLA	59
7.1	LAN-KARGA ETA DESBIDERAKETAK	60
7.2	ALDAKETAK.....	61
8.	ONDORIOAK	63
8.1	HELBURU OROKORREN KALITATE MAILA	64
8.1.1	Teknologia-azterketa	64
8.1.2	Web-aplikazioa	64
8.1.3	Ikasitako lezioak	65
8.2	WEB-APLIKAZIOAREN GARAPENETIK ATERATAKO ONDORIOAK	66
8.3	ETORKIZUNEKO AUKERAK	67
	Erreferentzia nagusiak	69
	A Eranskina: Proiektuaren jarraipen-dokumentua	71
	B Eranskina: Bilera-aktak.....	81
	C Eranskina: Sekuentzia-diagramen bilduma osoa.....	89

Irudi eta taulen zerrenda

IRUDIEN ZERRENDA

1. irudia: GRALaren LDE diagrama.	7
2. irudia: Proiektuaren Gantt diagrama.	9
3. irudia: MEANen osagaiak.	19
4. irudia: Meteor ingurunearen ikuspegi orokorra.	20
5. irudia: Mojito ingurunearen barne-egitura.	21
6. irudia: Socket.io-ren egitura.	22
7. irudia: Erabilpen-kasuen diagrama.	32
8. irudia: "Foroak" atalaren erabilpen-kasuen diagrama.	33
9. irudia: "Berriak" atalaren erabilpen-kasuen diagrama.	34
10. irudia: "Ikastaroak eta formakuntza" atalaren erabilpen-kasuen diagrama.	35
11. irudia: "Jarduerak" atalaren erabilpen-kasuen diagrama.	36
12. irudia: "Profila" atalaren erabilpen-kasuen diagrama.	36
13. irudia: "Erabiltzaile-kudeaketa" atalaren erabilpen-kasuen diagrama.	37
14. irudia: "Biltegi digitala" atalaren erabilpen-kasuen diagrama.	37
15. irudia: Web-aplikazioaren klase-diagrama.	38
16. irudia: "Login egin" erabilpen-kasuaren sekuentzia-diagrama.	42
17. irudia: "Pasahitza aldatu" erabilpen-kasuaren sekuentzia-diagrama.	43
18. irudia: "Eztabaidan sartu" erabilpen-kasuaren sekuentzia-diagrama.	43
19. irudia: "Eztabaida sortu" erabilpen-kasuaren sekuentzia-diagrama.	45
20. irudia: Login orria.	53
21. irudia: Administratzailearen menu nagusia.	53
22. irudia: Biltegi digitala (administratzaile-rolan).	54
23. irudia: Dokumentu bilduma (administratzaile-rolan).	55
24. irudia: Foroa (erabiltzaile-rolan).	55
25. irudia: Eztabaida (erabiltzaile-rolan).	56
26. irudia: Foro berria sortzeko menua (administratzaile-rolan).	56
27. irudia: Jarduerak (erabiltzaile rolean).	57
28. irudia: Erabiltzaile-kudeaketa (administratzaile rolean).	57
29. irudia: "Foroa ezabatu" erabilpen-kasuaren sekuentzia-diagrama.	89
30. irudia: "Jarduera ikusi" erabilpen-kasuaren sekuentzia-diagrama.	90
31. irudia: "Jardueran izena eman" erabilpen-kasuaren sekuentzia-diagrama.	90
32. irudia: "Irten" erabilpen-kasuaren sekuentzia-diagrama.	91
33. irudia: "Dokumentua igo" erabilpen-kasuaren sekuentzia-diagrama.	92
34. irudia: "Argazkia ikusi" erabilpen-kasuaren sekuentzia-diagrama.	92

TAULEN ZERRENDA

1. taula: Lan-paketeen denbora-estimazioak.	9
2. taula: Garapen-inguruneen osagaien laburpena.	22
3. taula: Garapen-inguruneen funtzionalitate-ezaugarrien laburpena.	23
4. taula: Erabilitako aurredefinitutako paketeak.	49
5. taula: Lan-paketeen denbora estimatuaren eta dedikatuaren arteko alderaketa.	60

Esker onak

Lehenik eta behin, eskerrik asko Gradu Amaierako Proiektu honetan nire tutoreak izan diren Xabier Artola eta David Buenestadori, hilabete hauetan izandako dedikazioagatik eta irakatsi eta lagundu didazuen guztiagatik. Eskerrik asko bie.

Mila esker ere graduan zehar izandako gainontzeko irakasleei eta esker bereziak tutoretzetan nirekin pazientzia izan duzuenoi (ez zarete gutxi, eta). Noski, mila esker ere graduan zehar zuen babesa eman eta animatu nauzuen lagunei.

Eta, nola ez, eskerrik asko fakultatean egindako lagun guztiei. Plazer hutsa izan da zuekin urte hauetan ikasketa, barreak, parrandak eta beste hainbeste gauza konpartitzea. Mila esker lau urte hauengatik, bikainak izan dira.

1. SARRERA

Kapitulu honetan, proiektua testuinguruan jarriko da eta sarrera labur bat egingo da, proiektuaren ideia nola garatu den azalduz.

1.1 AURREKARIAK

Gaur egun web-aplikazioak aurki ditzakegu nonahi. Eta gero eta gehiago dira hauek sortzeko garatzaileen eskura dauden plataforma, software, garapen-ingurune eta abarrak. Baina, ingurune hauek egitura komunik jarraitzen al dute? Badute antzeko ezaugarririk?

Galdera horiei lotuta garatzaile batek beste zalantza bat izan dezake web-aplikazio bat eraikitzeke plataforma egokia bilatzerako garaian: zein aukeratu? Azken galdera honen harira eraikitzen da proiektu hau, zeinak web-aplikazioak garatzeko egungo programazio-paradigma ezagunena aztertuko duen: gertaeretan oinarritutako programazioa. Paradigma hau gero eta gehiago erabiltzen da programazioan, eta bere aplikazio zehatza du web-aplikazioetarako, gertaeretan oinarritutako zerbitzariak erabiltzen dituzten web-aplikazioetarako, hain zuzen.

Hainbat teknologiak ahalbidetzen dute gertaeretan oinarritutako zerbitzarien erabilera, eta teknologia bakoitza hamaika garapen-ingurune ezberdinek baliatzen dute, beste osagai batzuekin batera, web-aplikazioak garatu ahal izateko. Hain zuzen, azterketa honetatik ateratako ondorioak web-aplikazio bat sortzeko proposamenean erabiliko dira: Gautena erakundearen gunee pribatua.

1.2 PROIEKTUAREN TESTUINGURUA ETA MOTIBAZIOA

Gautena, Gipuzkoako Autismo Elkarte, irabazi-asmorik gabeko erakundea da eta autismoaren inguruan gizartea sentsibilizatzea eta autismoa duten pertsonen arreta ematea ditu helburutzat.

1978 urtean ekimen sozialetik sortutako elkarte honek, egun 750 familia ingururi ematen die arreta, eskaintzen dituen zerbitzuen bitartez: diagnostia eta tratamendua, hezkuntza, eguneko arreta, etxebizitzak, familiei aholkularitza... Horrez gain, urtero hainbat sentsibilizazio-kanpaina eta -ekimen antolatzen ditu, autismodun pertsonen gizartean duten ikusgarritasuna hobetze aldera, hala nola iragarkiak, urtero egiten den "Urdinduz" ibilaldi solidarioa edota 2017ko apirilean aurkeztu berri duten "deskonektapp" autismodun pertsonen sorturiko mugikorrerako aplikazioa.

Irabazi-asmorik gabeko erakundea izanik, administrazio publikotik jasotzen du bere aurrekontuaren gehiengoa (% 92) eta gainerakoa euren zerbitzuak erabiltzen dituzten familien eta bestelako bazkideen ekarpenekin osatzen da. 9.000.000 € inguruko aurrekontua du elkarteak azken urteotan.

Gautena, familiei ematen dien zerbitzu eta arreta hobetze aldera, gaur egun hain garrantzitsuak diren plataforma digitalak (webguneak, sare sozialak...) hobetzeko ahalegina egiten ari da. Horrelako euskarrien garapenak familiekiko gertutasuna handiagoa izatea ahalbidetzen du, eta hauek zenbait informazio edo laguntza errazago eskuratu ahal izatea ere. Testuinguru horretan sortzen da webgune moderno bat edukitzeko beharra eta proiektu honen baitan sortuko den gunee pribatua diseinatu eta eraikitzeke proposamena.

Aurreko atalean aipatu bezala, familiak Gautenako bazkide egin daitezke, erakundeari hilero laguntza ekonomiko bat emanez, baina hori ez da ezinbesteko baldintza Gautenaren zerbitzuak jaso ahal izateko, Osakidetza bideratuta, doakoak baitira diagnosi-

eta tratamendu-zerbitzuak edozein haurrentzat. Beraz, badira Gautenaren zerbitzuak jaso arren, bazkideak ez diren familiak.

Gune pribatuaren helburuetako bat bazkide diren familiei zerbitzu gehigarri bat ematea da, normalean jasotzen dutenaren osagarria eta erosoena. Beste helburu garrantzitsu bat, adituengandik laguntza jasotzeaz gain, gune pribatua familien arteko komunikazioa errazteko tresna izatea da, beren artean zalantzak eta esperientzia partekatuz errazagoa izan daitekeelako autismodun pertsona baten zaintza kudeatzea. Azken helburua familiek formatu digitalean lagungarria suertatu ahal zaien materiala eskuratu ahal izatea da, Gautenak egiten dituen jardueren argazki eta bideoez gain.

1.3 PROPOSATUTAKO SOLUZIOA

Web-aplikazio bat sortzeko proposamena mahai gainean izanda, gradu-amaierako proiektu bezala teknologia-azterketa bat egitea erabakitzen da, interes nagusia aplikazioan jarri ordez (normalean egiten den moduan), teknologian jartzeko. Web-aplikazioa sortzeko proposamena onartuko da, baina teknologia-azterketan lortutako informazioa erabili eta ateratako ondorioak praktikan frogatzeko baliatuko da garatze-prozesu hori. Laburbilduz, helburu nagusia teknologia-azterketa bat egitea izango da, gertaeretan oinarritutako programazioaren paradigmatik abiatuz eta web-aplikazioen sortze-prozesuarekin duen erlazioan sakonduz, ondoren, azterketa web-aplikazioaren garatzearekin biribilduz.

Azterketaren xehetasun gehiago eta emaitzak 3. atalean azalduko dira.

2. KUDEAKETA-PLANA

Atal honetan, proiektua gauzatu ahal izateko diseinatu den kudeaketa-plana azalduko da, egindako estimazioekin batera.

2.1 IRISMENA

Lehenik eta behin, proiektuaren irismena definituko da, hau da, zein atal egon beharko diren proiektuaren entrega egiterako garaian osatuak eta zeintzuk ez, irismenetik kanpo geratu direlako. Proiektu honek, aurrez aipatu bezala, atal teoriko bat (teknologia-azterketa) eta haren osagarria den atal praktiko bat (web-aplikazioaren garapena) dituenez, atal bakoitzarentzat irismen zehatza finkatu da.

Teknologia-azterketaren irismena:

- Gertaeretan oinarritutako programazioaren paradigma aztertu eta web-aplikazioak garatzeko testuinguruaren barnean azaltzea.
- Gertaeretan oinarritutako zerbitzariak baliatzen dituzten hainbat teknologia aztertzea eta horietako bat aukeratzea, teknologia horri lotutako hainbat garapen-ingurune aztertzea.
- Garapen-inguruneen arteko konparaketa bat egitea, funtzionalitate ezberdinak aztertuz eta onena dena aukeratzea.

Web-aplikazioaren irismena:

- Bezeroaren eskakizunak aintzat hartuz, eskatutako web-aplikazioaren prototipo bat garatzea, aplikazio erreal baten adibide modukoa izan daitekeena. Irismenetik kanpo geratzen da web-aplikazioa funtzionamenduan jartzea, erakundearen zerbitzaria baliatuz.
- Web-aplikazioaren garapenaren bitartez, aukeratutako teknologia eta garapen-ingurunearen egokitasuna frogatzea, aldeko eta kontrako ondorio zehatzak lortuz.

Helburu gehigarriak:

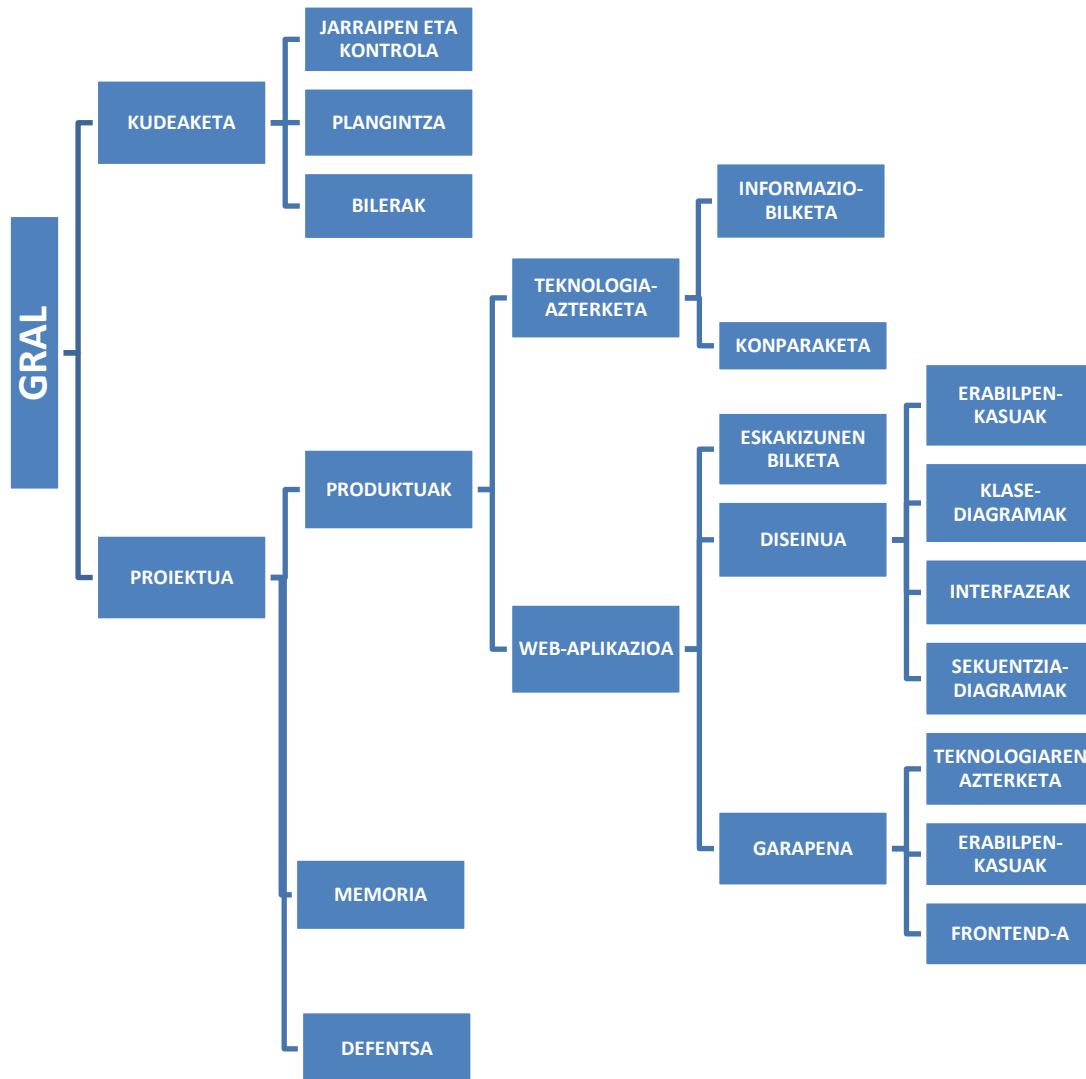
- Teknologia berri batekin trebatzea eta ikasitakoa garapenean aplikatzea.

2.2 PLANGINTZA

Proiektuaren irismena, hau da, lortu beharko diren emaitzak jakinda, proiektua aurrera eramateko planifikazio bat burutu da, lan-ataza ezberdinak definituz eta denbora kudeatzeko plan bat eginez.

2.2.1 Lan-paketeen deskonposaketa

Proiektua hainbat lan-paketetan banatu da, zeregin desberdinak eta euren lan-kargak bereizteko. Jarrion erakusten den LDE diagraman ikus daiteke banaketa.



1. irudia: GRALaren LDE diagrama.

Jarraian, lan-paketeak deskribatuko dira atazaka:

- **Kudeaketa**
 - **Jarraipen eta kontrola:** proiektuaren egoera kontrolatzeko, egindako lanaren eta izandako arazoaren eguneroko moduko bat osatzea.
 - **Plangintza:** proiektua aurrera eramateko ataza ezberdinen identifikazioa eta denbora-estimazioak egitea.
 - **Bilerak:** proiektuaren egoeraren berri emateko eta zuzenketak egiteko tutorearekin egindako bilerak.

- **Produktuak, teknologia-azterketa:**
 - **Informazio-bilketa:** aztertuko den gaiari buruz ikertzea eta ahalik eta iturri gehienetatik informazioa eruztea, analisi zabala burutu ahal izateko.
 - **Konparaketa:** aztertuko den gaiari lotutako teknologia eta garapen-ingurune ezberdinak alderatzea, aurrez lortutako informazioa baliatuz.
- **Produktuak, web-aplikazioa:**
 - **Eskakizunen bilketa:** web-aplikaziorako bezeroak eskatutako ezaugarriak jaso eta aztertzea.
 - **Diseinua:** web-aplikazioaren diseinua egitea, alderdi ezberdinetan: interfazeen diseinua, barne egituraren diseinua...
 - **Garapena:** aplikazioa garatzea garapen-teknologia aztertuz eta erabiliz.
- **Memoria:** proiektuaren fase guztiak eta egindako lana jasotzen dituen txostena idaztea.
- **Defentsa:** Informatika Fakultateko epaimahaiaren aurrean proiektua aurkeztu eta egindako lana justifikatzeko beharrezko materiala prestatu eta hizkera lantzea.

2.2.2 Lan-kargaren estimazioa

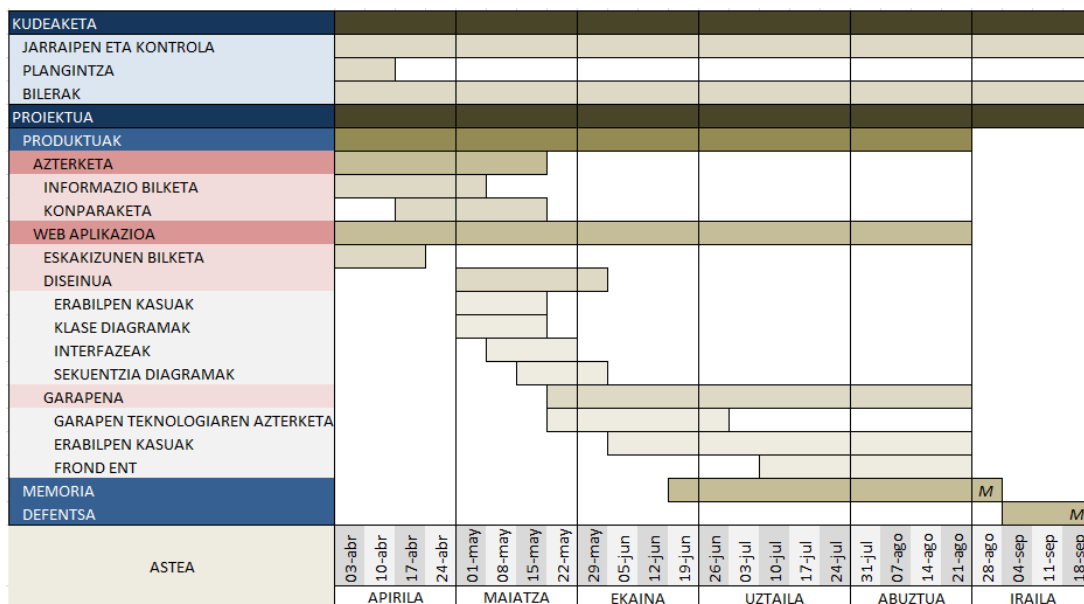
Proiektua lan-paketetan deskonposatu ondoren, ataza bakoitzaren lan-karga esleitu da estimazioak eginez. Estimazio hauek proiektua amaitzean berrikusiko dira, desbideraketa posibleak eta hauen arrazoiak aztertzeko.

LAN-PAKETEAK	ESTIMAZIOA (ORDUTAN)
KUDEAKETA	50
JARRAIPEN ETA KONTROLA	35
PLANGINTZA	5
BILERAK	10
PROIEKTUA	250
PRODUKTUAK	210
AZTERKETA	42
INFORMAZIO-BILKETA	12
KONPARAKETA	30
WEB-APLIKAZIOA	168
ESKAKIZUNEN BILKETA	10
DISEINUA	38
ERABILPEN-KASUAK	12
KLASE-DIAGRAMAK	13
INTERFAZEAK	5
SEKUENTZIA-DIAGRAMAK	8
GARAPENA	120
TEKNOLOGIAREN AZTERKETA	15
ERABILPEN-KASUAK	83
FRONTEND-A	22
MEMORIA	32
DEFENTSA	8

1. taula: Lan-paketeen denbora-estimazioak.

2.2.3 Gantt diagrama

Lan-pakete bakoitzak denboran zehar izango duen iraupena eta baita hasiera- eta amaiera-datak finkatzeko, Gantt diagrama osatu da, ataza bakoitza zein epetan egingo den adierazten duena. Berriz ere, diagrama honek hasierako estimazioak erakusten ditu, eta proiektua amaitzean jarraipen eta kontroleko oharrak aztertuz konprobatuko da ea epeak bete diren edota desbideraketak egon diren.



2. irudia: Proiektuaren Gantt diagrama.

2.2.3.1 Mugarriak

Proiektu honetan bi mugarri identifikatzen dira, eta Gantt diagraman “M” hizkiak adierazten dira, dagokien astean. Zehazki, hauek dira mugarriak:

- **2017ko irailak 3a, proiektuaren azken entrega-data.** Egun horretarako proiektuak amaituta egon beharko du eta memoriaren txostena ADDI plataformara igo beharko da, tutoreak oniritzia eman ondoren.
- **2017ko irailaren 18-21a, proiektuaren defentsa-data.** Tarte horretako egun batean (oraindik zehaztu gabe dagoena), epaimahaiaren aurrean aurkeztu beharko da proiektua, egindako lana justifikatuz eta web-aplikazioaren demo moduko bat aurkeztuz. Aste horren hasierarako, beraz, aurkezpenerako beharrezko materialak prest egon beharko du eta ahozko mintzaira ere landu beharko da.

2.3 ARRISKUEN IDENTIFIKAZIOA

Proiektuan zehar hainbat ezusteko gerta daitezke. Batzuk egilearen kontrolpean egongo dira baina beste batzuk kanpo-faktoreetatik etor daitezke. Proiektu honetan honako arriskuak identifikatu dira, ezustekoen aurrean prebentzio-neurriak hartu ahal izateko:

- **Informazio-galera:** Informazioa galtzea proiektu honen arrisku nagusia da, proiektuan atzerapen handiak eragin baititzake. Proiektuan zehar osatzen joango den materiala proiektua bera aurrera eramateko beharrezkoa izateaz gain, GRALaren Memorian sartu behar da, egindako lana justifikatzeko. Arazoa ekiditeko, dokumentu eta fitxategi guztien hainbat kopia egingo dira mementoro: bata proiekturako erabiliko den konputagailuan eta bestea Google Drive sareko plataforman. Horrez gain, hodeian gordetzen den guztiaren segurtasun-kopia bat egingo da periodikoki, kanpo-memoria batean gordetzeko. **Arrisku maila: handia.**
- **Konputagailua hondatzea:** Honek arazo handia suposa dezake, ordenagailurik gabe proiektua ezin baita aurrera eraman eta gutxiago proiektuaren inplementazioa (kodetzea). Hala ere, konputagailu bat hondatuz gero etxeko edo Informatika Fakultateko beste bat erabiltzea posible izango litzateke, honek atzerapen txiki bat eragingo balu ere. **Arrisku maila: ertaina.**
- **Web-aplikazioaren garapen-arazoak:** Garapenean gertatzen den ohiko arazo bat inplementatzerako garaian aurreikusten ez ziren zailtasunak izatea edota estimatutakoa baino denbora gehiago behar izatea da. Oso garrantzitsua izango da ahalik eta estimazio zehatzena egitea eta garapen-teknologia ondo aztertzea tartea eduki ahal izateko. Gehiegizko atzerapenak proiektua aurkeztuko den deialdia atzeratzea eragin dezake. **Arrisku maila: handia.**
- **Bezeroaren eskakizunak aldatzea:** Posible izango litzateke bezeroak proiektu erdian bere eskakizunak aldatzea eta honek proiektuaren garapena atzeratzea edota denbora-galera handia sortzea. Kasu honetan, ordea, bezeroa oso eskertua agertu da proiektuaren proposamenarekin eta garatzaileak egin dezakeen edozerekin konforme dagoela adierazi du; ez du eskakizun zorrotzik egin. **Arrisku maila: baxua.**

2.4 INTERESATUAK

Proiektu honetan hainbat interesatu agertzen dira. Lehena, proiektuaren egilea bera, proiektua kudeatzeko eta modu arrakastatsuan berau amaitzeko ardura eta erantzukizuna dituen.

Beste interesatu bat proiektuaren zuzendaria da, Informatika Fakultateko Xabier Artola irakaslea. Haren ardura proiektuaren egilea gidatzea, aholkatzea eta zuzentzea izango da, proiektua modu egokian burutu dezan.

Azkenik bezeroa dugu, Gautenako teknikaria den Jimmy Brossa, zeinak proiektuaren barnean kokatzen den web-aplikazioaren prototipoa garatzea nahi duen, erakundearen zerbitzuak jasotzen dituzten familiei zerbitzu hobea eta osatuago bat eman ahal izateko. Berea izan da web-aplikazioa garatzeko proposamena eta berak du proiektuaren egileari beharrezkoak dituen eskakizun eta xehetasun teknikoak helarazteko ardura, proiektuak erakundearen hainbat aspektu hobetzen lagun dezan. Bezeroarekin batera Gautenako

familiak ere proiektuko interesatuak dira, web-aplikazioak zenbait informazio eta material adituengandik modu errazagoan lortzea ahalbidetu diezaiekeelako.

3. WEB-APLIKAZIOAK GARATZEKO GERTAERETAN OINARRITUTAKO ZERBITZARIAK ERABILTZEN DITUZTEN TEKNOLOGIA ETA INGURUNEEN AZTERKETA

Atal honetan teknologia-azterketa egingo da, gertaeretan oinarritutako zerbitzarien kontzeptua aztertuz eta garapen-ingurune egokiena bilatzeko asmoz.

Egungo garapen-teknologia eta -inguruneen aukera zabala aztertu eta horietatik hainbatetan sakondu nahi izatetik dator azterketa hau egiteko ideia. Egungo *framework* berritzaileen artean, modu ahalik eta objektiboenean, “hoberena” edo interesgarriena aurkitzea baitu helburutzat. Aukera onenak eskaintzen dituen ingurunea, ondoren, gure web-aplikazioa eraikitzeke erabiliko da. Horretarako, web-aplikazioetan oso erabilia den paradigma bat aztertuko da lehenik, jarraian berori erabiltzen duten hiru teknologia aztertu eta, amaitzeko, horietako batekin lan egiten duten garapen-inguruneen artean egokiena aukeratzeko.

3.1 GERTAERETAN OINARRITUTAKO PROGRAMAZIOA

Programazioan, hainbat paradigma definitu dira gaur egun arte, kasu zehatz bakoitzarentzat egokiena izan daitekeen soluzioa erabiltzeko asmoz. Paradigma bakoitzak kontzeptu baten inguruan oinarritzen du bere estiloa edo egitura, eta horietako batzuk hauek dira: datuetan oinarritutako programazioa, objektuei orientatutakoa, deklaratihoa, semantikoa, lengoian oinarritutakoa, automatetan oinarritutakoa... Horien artean, gertaeretan oinarritutako programazioaren paradigma da atal honetan aztertuko dena.

Gertaeretan oinarritutako programazioan¹ (*event-driven*, ingelesez), izenak adierazten duen bezala, programaren fluxua gertaerek definitzen dute, hala nola, erabiltzaileak sakatutako botoiek edo saguarekin egindako *inputek*. Beraz, programaren fluxua ez du programaren egileak definitzen, baizik eta ondoren programa erabiltzen duen erabiltzaileak (edo programa martxan jartzen duen dena delakoak). Horretarako, programaren logika egituratzerakoan, ezinbestekoa da programaren exekuzioan zehar gerta daitezkeen gertaerak definitzea, hauek aktibatzean jazoko diren ekintzekin batera. Hori lortzeko, *kudeatzaileak* edo *event-handler* delakoak erabiltzen dira. Kudeatzaile bakoitzak kode zehatza izango du, dagokion gertaerari erantzuteko balio duena.

3.2 WEB-APLIKAZIOEN GARAPENERAKO GERTAERETAN OINARRITUTAKO ZERBITZARIAK ERABILTZEN DITUZTEN TEKNOLOGIAK

Kontzeptu hau web-zerbitzarietara pasatzen badugu (proiektu honetan web-aplikazio bat sortuko den heinean), paradigmak aldi berean eta modu konkurrentean hainbat konexio kudeatzeko beharrari soluzioa bilatzeko balio digu. Web-zerbitzari batek hori egiteko hainbat modu ditu: konexio bakoitzeko prozesu bat abiatzea, bakoitzak hari (*thread*) bana izatea... Baina gertaeretan oinarritutako programazioaren xedea erabiliz, prozesu bakar baten bidez egiten da dena, honek gertaera bakoitza erabiltzailearen *inputei* erantzunez kudeatuko baitu. Hortaz, web-zerbitzari tradizionalekin alderatuz, non erabiltzaile bakoitzeko prozesu bat abiatu eta prozesu hau ekintzak amaitu arte martxan mantendu behar den, gertaeretan oinarritutako zerbitzariak prozesu bakarrekin

¹ <http://eventdrivenpgm.sourceforge.net/> (kontsulta: 2017-06-10).

erabiltzaile guztien ekintzak kudea ditzakete, gertaerak sortzen direnean bakarrik erantzun behar baitute. Programa abiatzean beharrezko hasieraketak egingo dira, eta ondoren programa blokeatuta geratuko da, gertaeraren bat jasotzeko zain.

Paradigma hau interfaze grafikodun aplikazioetan erabiltzen da, hala nola, VB, .Net edota Javascript bidez garatutakoetan, adibide esanguratsu batzuk aipatzeagatik. Javascripten erabilerak, hain zuzen, izugarri bultzatu du azken urteotan web-aplikazioak garatzeko gertaeretan oinarritutako teknologien erabilera, lengoiaia bai bezeroan bai zerbitzarian erabili ahal delako.

Gertaeretan oinarritzen diren zerbitzariak erabiltzen dituzten hainbat teknologia oso ezagun bilakatzen ari dira egungo programatzaileen artean. Hurrengo azpiatalean, horietako hiru aztertuko ditugu, web-aplikazio bat garatzeko teknologia aproposena bilatzeko asmoz: *NodeJS*, *Twisted* eta *Vert.x*.

NodeJS²

NodeJS Javascript bezeroan eta zerbitzarian erabiltzen duen software libreko ingurunea da. C, C++ eta JavaScript lengoaietan dago idatzia.

Bi aldeetan Javascript lengoiaia bakarra erabiltzen duenez, NodeJS erabiltzen duten garapen-inguruneak *isomorfikoak* direla esaten da. Zerbitzarian, orrialdeak erabiltzailearen web-nabigatzaileara bidali aurretik, eduki dinamikoa sortzea ahalbidetzen dio Javascripten erabilerak.

Aurreko atalean aipatu bezala, NodeJS-k ere exekuzio-hari bakarra erabiltzen du, sarrera eta irteera asinkronoak erabiliz. Horrela oso konkurrentzia altua izatea ahalbidetzen da. Exekuzio-ingurunean, konexio bakoitzari memoria dinamikoko zati bat esleitzen zaio, hari propio bat sortzeko beharrik gabe.

Beste ezaugarri nabarmen bat programazio asinkronoaren erabilera da. Prozesu bat martxan dagoen bitartean, exekuzio guztia esperoan mantendu ordez (ohikoa den bezala), prozesuak erabiltzen dituen baliabideak aska daitezke amaitu bitartean beste prozesu batzuk erabili ahal izateko. Hala ere, programazio asinkronoa erabiltzen duten *callback* anitzek kodea ulertzeko konplexuagoa izatea eragiten dute, funtzio batzuk besteen barruan sartzen baitira.

Twisted³

Hau ere software libreko ingurunea da, baina, aurreko teknologia ez bezala, Twisted Python lengoiaz idatzia dago eta bere diseinuaren oinarria protokolo logiko eta geruza fisikoen bidezko garraioaren arteko bereizketan dago. Plataforma honek HTTP, FTP, IMAP, POP3 eta beste hainbat protokolo onartzen ditu

Bere ezaugarri bereizgarrienetako bat *deferred* kontzeptua erabiltzea da, hau da, etorkizunean erabiliko den ezaugarri bat definitzeko aukera ematea, baina bere balioa ezagutu gabe, oraindik ez delako posible hori egitea. *Deferred* ezaugarriak balioa hartzen

² <https://nodejs.org/en/> (kontsulta: 2017-07-16).

³ <https://twistedmatrix.com/trac/> (kontsulta: 2017-08-08).

duenean (normalean haria amaitzerakoan) “atzeraka” pasatzen die balioa aurretik jaso dituen dei guztiei.

Python lengoaiak eskaintzen dituen errore-kudeaketako sistemei esker, Twisted oso teknologia egonkorra eta segurua da. Internet zerbitzuak sortzeko hamaika API eskaintzen ditu, erabiltzeko prest dauden hainbat protokolo eta ezaugarri esker.

Twisted sarearen euskarri-geruza bezala erabil daiteke, kanpoko-gertaeren begiztekin batzeko aukera ematen du eta. Horrela, posible da adibidez, web-zerbitzari baten prozesu oso bat interfaze aplikazio batekin konbinatzea.

Vert.x⁴

Software libreko teknologia hau Java makina birtualean exekutatzen da eta, askoren hitzetan, NodeJS-ren Javako baliokidea da, lengoia horretan idatzia baitago. Ingurune simple eta arina da (memoriako 650kB hartzen ditu soilik) baina aplikazio osatuak eraikitzeko aukera ematen du. Garapen-ingurune azkarrenetakoa da gaur egun⁵, TechEmpower webguneak egindako testen emaitzei jarraiki. Bosch eta Ticketmaster bezalako enpresek Vert.x bidez garatutako softwarea erabiltzen dute gaur egun.

Hala ere, ez da beharrezkoa Java ezagutzea teknologia honetan programatzeko, Vert.x-en ezaugarri esanguratsuena lengoia anitzetan idatz daitekeela baita (*polyglot*): Java, JavaScript, Groovy, Ruby, Ceylon, Python... Hain zuzen, bere izenaren parte den “x”-ren bitartez adierazi nahi izan zuten bere garatzaileek lengoia-aniztasun hori.

Aurrekoak bezalaxe prozesu bakarra erabiltzen du konexio guztietarako, eta, bere gertaera-busari esker, zeina bezero, zerbitzari eta nabigatzaileko JavaScript geruzan hedatzen den, denbora errealeko aplikazioak sortzeko aukera ematen du.

3.3 GARAPEN-INGURUNEAK

Azterketan sakontzen jarraitzeko, hiru teknologia hauen artean bat aukeratzea erabaki da, teknologia horretan oinarrituta dauden edo hura erabiltzen duten hainbat garapen-inguruneraren arteko konparaketa osatzeko. Kasu honetan, NodeJS hautatu da, hainbat arrazoi direla medio. Alde batetik, ikusi da NodeJS-z baliatzen diren hainbat eta hainbat framework⁶ daudela merkatuan gaur egun baina beste bi teknologietan aukera ez dela hain anitza, adibide batzuk badiren arren (Vert.x-en Axon⁷, esaterako). Vert.x eta Twisted zuzenean garapen-ingurune bezala erabili ohi dira, NodeJS beste teknologia mota batzuekin konbinatzeko baliabide bezala erabiltzen den bitartean. Bestalde, NodeJS proiektu honen egileak erabili izan du beste proiektu batzuetan, eta, Javascript lengoia ezaguna izanik, logikoagoa dirudi hori hautatzeak, teknologia-azterketaren jarraian garatuko den aplikazioa eraikitzeko.

⁴ <https://picodotdev.github.io/blog-bitix/2016/01/introduccion-y-ejemplo-basico-sobre-vertx/> (kontsulta: 2017-08-14).

⁵ <https://www.techempower.com/benchmarks/#section=data-r8&hw=i7&test=plaintext> (kontsulta: 2017-08-08).

⁶ <https://www.devsaran.com/blog/10-best-nodejs-frameworks-developers> (kontsulta: 2017-08-08)

⁷ <http://blog.trifork.com/2012/11/27/basic-axon-framework-sample-using-vert-x-and-angular-js/> (kontsulta: 2017-08-08).

NodeJS erabiltzen duten hainbat garapen-ingurune daude erabilgarri gaur egun, bakoitza bere berezitasunekin. Aurrez aipatu bezala, azterketa honen azken puntuan horietatik “objektiboki” (hau da, dagokigun aplikazioaren kasuari aplikatu gabe) ezaugarri interesgarrien edo onenak dituen aurkitzen saiatuko gara. Aurki daitezkeen framework anitzen artean, ondorengo bostak aukeratu dira elkarren artean konparatzeko: Mean, Meteor, Mojito, Socket.IO eta Total.

3.3.1 Konparaketa-irizpideak

Konparaketa egiteko erabiliko diren irizpideak bi multzotan sailkatzea erabaki da: alde batetik, osagaiekin lotutako irizpideak eta, bestetik, funtzionalitateei lotutakoak. Lehenak, garapen-inguruneak osatzen dituzten aplikazio, ingurune edo “osagaiak” hartzen ditu barne, hala nola, erabiltzen den datu-base mota edo bezeroko teknologiak; bigarrenak, aldiz, garapen-ingurune bakoitzak garapenerako ematen dituen baliabide tekniko edo funtzionalak hartzen dira kontuan, MVC patroia jarraitzen duen edo aurredefinitutako txantiloirik erabiltzeko aukera ematen duen, esaterako. Jarraian deskribatuko dira irizpideok, bakoitza bere testuinguruaren barnean. Konparaketaren amaieran, framework egokiena aukeratzeko funtzionalitateei lotutako irizpideak hartuko dira kontuan, eta osagaien azterketa erabiltzen dituen baliabideak ezagutze aldera burutuko da soilik.

3.3.1.1 Osagaiei lotutako irizpideak

- **AngularJS**⁸. Googlek garatutako ingurune honek aplikazioen *frontend-a* garatzeko aukera zabalak ematen ditu. Bootstrap txantiloiez eta berezkoak dituen HTML etiketen bitartez, edozein aplikaziorantzako interfazeen garapena errazten du, zuzenean mugikorrerako egokituia bihurtuz (*responsive*). Aurredefinitutako txantilo bat egokitu daiteke minutu gutxitan aplikazioaren interfazea osatuta edukitzeko eta, horrez gain, inguruneak aplikazioen programazioan hain erabilia den MVC (Model-View-Controller) patroia jarraitzen du. Normalean, hala ere, Angular-ekin patroia honen moldaketa bat erabili ohi da: MVVM (Model-View-ViewModel).
- **Express**⁹. Web-aplikazioak eta APIak sortzeko zerbitzari-ingurunea da. Erabiltzeko erraza eta sinplea izateagatik da ezaguna eta, horrez gain, saio-kudeatzailea eta routinga bezalako aukerak eskaintzen ditu.
- **Backbone**¹⁰. JSON interfazeko JavaScript ingurunea, orri bakarreko web-aplikazioak sortzeko diseinatua. Beraz, aplikazioen aurkezpena garatzeko balio du. AngularJS-tik ezberdintzen duena programazio inperatiboa erabiltzea da, deklaratiiboaren ordez. MVC patroia jarraitzen du.
- **MongoDB**. Datu-base ez-erlazionala (NoSQL), hau da, ohiko egitura jarraitzen ez duena. Klaseak eta horien arteko erlazioak erabili ordez, *Collection*-ak erabiltzen ditu. Bilduma hauek lerro bakar batekin sor daitezke eta ez da eremurik definitu behar hauek definitzeko, ondoren nahi ahala gehitu daitezke eremuak. Gakorik ez du erabiltzen, aldiz, identifikatzaile bat txertatzen dio objektu bakoitzari modu

⁸ <https://angularjs.org/> (kontsulta: 2017-07-16).

⁹ <http://www.nodehispano.com/2012/01/express-el-framework-web-para-nodejs/> (kontsulta: 2017-07-16)

¹⁰ <http://backbonejs.org/> (kontsulta: 2017-07-16)

automatikoan, gero identifikatu ahal izateko. Datu-base hau oso erabilia da ingurune isomorfikoetan, haren sinpletasunak asko errazten baititu gauzak, baina kontuan izan behar dira egitura zehatzik ez jarraitzeak sor ditzaken arazoak: bildumen elementuen arteko eremu ezberdintasunak sortzea, lotura definitzeko aukera eza...

- **MySQL, HSQL.** Datu-base erlazionalak izateagatik batera aztertuko dira. Datu-base erlazional bezala, datuak klasetan antolatzen dira eta klase bakoitzak hainbat eremu edo atributu ditu, eta horietako bat gakoa da, hau da, klase bateko objektu bakar bat identifikatzeko balio duena. Klaseen arteko erlazioak definitzen dira datu-base erlazionalaetan, klase-egitura bat osatuz.
- **Grunt¹¹.** Web-aplikazioen sortze-prozesuan jarduera errepikakorrak automatizatzea ahalbidetzen duen tresna bat da, hala nola, testing-a, konpilazioa... Hainbat plugin ditu ohikoak diren prozesuak automatizatzeke eta garatzaileak berriak kodetu eta defini ditzake, bere beharren arabera.

3.3.1.2 Funtzionalitateei lotutako irizpideak

- **MVC patroia jarraitzea.** Model-View-Controller egitura (edo aldaera bat) erabili ahal izatea garrantzitsutzat jotzen da, garatzaileak izandako esperientziagatik egitura logiko eta erosoena delako aplikazio bat garatzerako garaian. Egitura honek aplikazioa hiru zatitan bereizten du: aplikazioaren interfazea edo aurkezpena, aplikazioaren egitura edo informazioaren irudikapena, eta bi alderdion arteko lotura egiten duen kontrolagailua.
- **Denbora-errealeko aplikazioak sortzea.** Denbora-errealeko aplikazio deitzen zaio eguneraketak modu automatikoan eta aplikazioa edo web-orria birkargatzeko beharrik gabe erakusten dituen aplikazioari. Erabiltzaile kopuru handi batek erabiltzen dituen aplikazioen egungo hazkunde ikaragarria kontutan izanda, garrantzitsutzat hartuko da erabiltzaile guztiek une berean datu eguneratuak jasotzeko aukera izatea.
- **Testing-erako aukerak ematea.** Aplikazioaren inplementazioa zuzena dela konprobatzeko, garapen-inguruneak testak egiteko aukera edo tresna integraturik eskaintzen duen baloratuko da.
- **Aurredefinitutako paketeak erabiltzeko aukera ematea.** Zenbait garapen-ingurunek aurredefinitutako funtzioak edo funtzio bildumak eskaintzen dituzte, funtzionalitate zehatz bat aldaketa gutxi batzuk eginda funtzionamenduan izateko.
- **Aurredefinitutako txantiloak erabiltzeko aukera ematea.** Interfazeen aldetik, aurkezpena garatzeko txantiloak erabili ahal izatea baloratuko da.

3.3.2 Framework-konparaketa

Behin jarraituko diren irizpideak argituta, garapen-inguruneen arteko alderaketa egingo da hurrengo atalean. Aztertuko diren inguruneak, aurrez esan bezala, hauek dira: Mean, Meteor, Mojito, Socket.IO eta Total.

¹¹ <https://gruntjs.com/> (kontsulta: 2017-07-16).

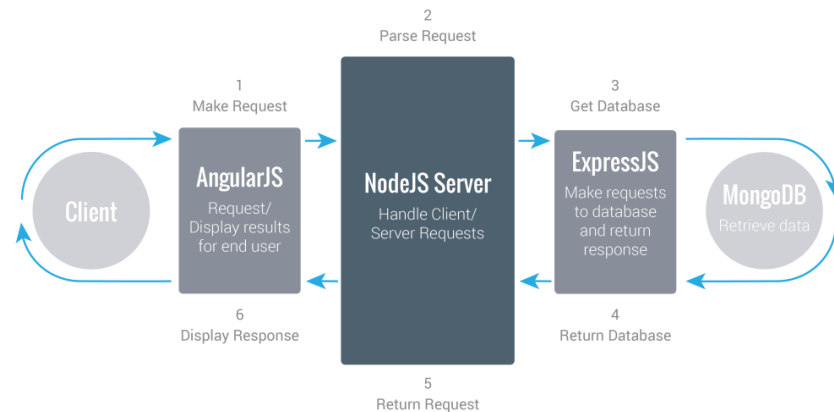
MEAN

MEAN¹² hainbat software-osagaiz osatutako garapen-ingurune ezaguna da, erabiltzaile gero eta gehiagok horiek batera erabiltzen zituztela ikusita sortu zena. Osagai horien lehen hizkiek frameworkaren izena osatzen dute: MongoDB datu-basea, Express zerbitzarian, Angular bezeroan eta NodeJS zerbitzarian ere.

Ingurune honek garatzailearen lana erraztea eta kodea ongi antolatua mantentzea du helburutzat, egitura sinplea eta ulertzeko erraza erabiltzeaz batera.

Angular erabiltzen duenez, Bootstrap CSS txantiloiak erabil daitezke frontend-a inplementatzeko, eta horrez gain, MVC patroia jarraitzen du. Hala ere Angular-ek ez du ohiko moduan erabiltzen egitura hau, aurrez aipatu bezala.

Testing automatikoa ahalbidetzen du Grunt tresnaren bitartez. *Testing-a* oso garrantzitsua da proiektuak abiatzera doazenean, inplementatutako guztia behar bezala doala ziurtatzeko.



3. irudia: MEANen osagaiak.

METEOR¹³

Gero eta jarraitzaile gehiago dituen kode-irekiko frameworka da hau. MongoDB erabiltzen du datu-base bezala, Angular erabiltzeko aukera ematen du bezeroan eta NodeJS zerbitzarirako. Denbora errealeko aplikazioak sor daitezke Meteor-ekin.

Ez du MVC patroia jarraitzen, karpeta-egitura ia bakoitzak bere gustuen arabera molda dezake, baina horrek MVC modura erabiltzea ere ahalbidetzen du. Berezko karpetak *Client*, *Server*, *Lib* eta *Public* ditu.

Meteor-ek berez hainbat pakete ekartzen ditu instalatzearekin bakarrik. Pakete horiek instalatzeko oso errazak dira (komando bat), eta funtzionalitate oso bat eskaintzen dute, zuzenean aplikazioan erabili daitekeena. Pakete horien adibide esanguratsuenak "accounts" paketea da, zeinak erabiltzaile eta pasahitzak erabiltzen dituen login-sistema bat

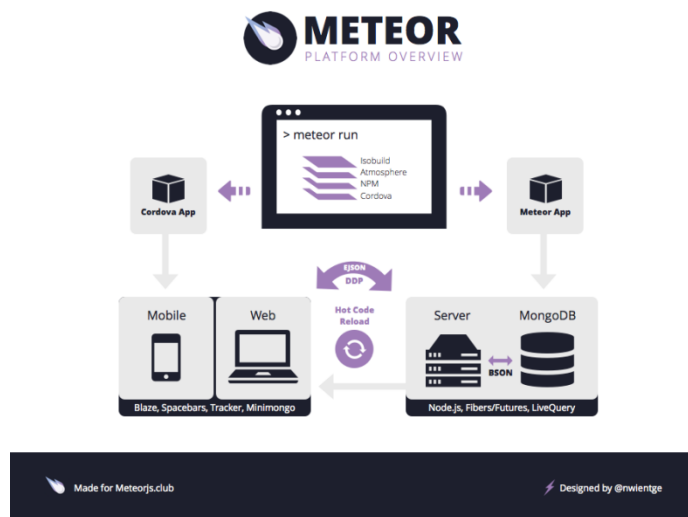
¹² <http://meanjs.org/> (kontsulta: 2017-07-16).

¹³ <https://www.meteor.com/> (kontsulta:2017-08-08).

eskaintzen duen. Hainbat paketeren konbinaketarekin funtzionalitate aurreratuagoak lor daitezke.

Aurredefinitutako txantiloiak kargatzeko aukera ematen du Meteor-ek HTML etiketak edo CSS fitxategiak erabiltzeko beharrik gabe (aukera honek gainerako frameworketatik bereizgarria egiten du Meteor). Aukera honetaz gain, Angular-ekin bateragarria denez, Bootstrap txantiloiez ere balia gaitzke.

Ezaugarri gehigarri moduan, esanguratsua da komunitate oso aktiboa duela. Framework ezaguna da bere komunitatearengatik, laguntza handia aurki baitaiteke haren foro, bideo eta abarretan.



4. irudia: Meteor ingurunearen ikuspegi orokorra.

MOJITO

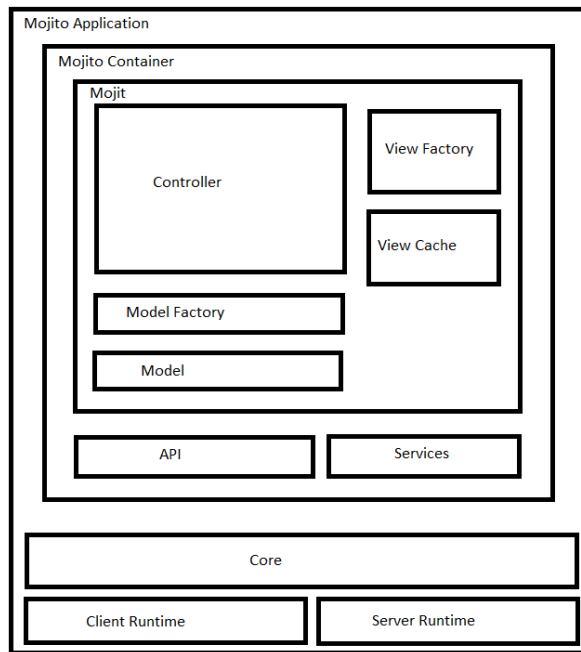
Mugikorrerako aplikazioak garatzeko *Yahoo! Cocktails*¹⁴ plataforman oinarritua dagoenez gero, edozein web-aplikazio zuzenean *responsive* izatea eskaintzen du, Bootstrap txantiloia bat erabiltzeko beharrik gabe. MVC patroia jarraitzen du eta test unitarioak egiteko tresna integraturik du, *Arrow* izeneko hain zuzen .

Internazionalizazioa eta lokalizazioa sinplifikatzeko liburutegia du, zeinak lerro gutxi batzuk kodetuz aplikazioak hainbat hizkuntzatan erabili ahal izateko aukera ematen duen. Horrez gain, sintaxi- eta bestelako kode-zuzentzaileak ere barneratzen ditu.

Bezero alderako, ez du Angular edo bestelako teknologiarik erabiltzeko aukerarik ematen, HTML eta CSS soilaz baliatzen da.

Besterik ezean, datu-basearen konfigurazioa *in-memory* HSQL datu-base baten bidez egiten da, baina MySQL erabil daiteke konfigurazio-aldaketa sinple batzuen bitartez.

¹⁴ <https://developer.yahoo.com/cocktails/mojito/> (kontsulta: 2017-08-08)



5. irudia: Mojito ingurunearen barne-egitura.

SOCKET.IO

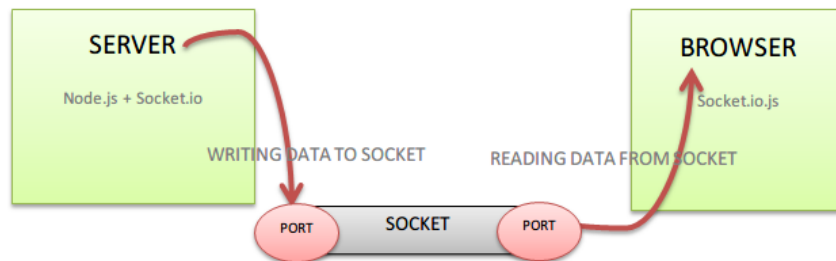
Ingurune honek denbora errealean eguneratzen diren aplikazioak sortzen ditu, WebSocket-en erabilerari esker, eta hau du bere ezaugarri indartsuena. Aurrez aipatu bezala, egungo merkatuan ezinbesteko garrantzia duen ezaugarria da hau. Bestalde, aplikazioak azkar sortzeko pentsatua dago, bere garatzaileen esanetan¹⁵.

Mezularitza eta txata sortzeko aurredefinitutako paketea dakar. Lerro gutxi batzuekin txat moduko bat eraikitzeko aukera ematen du, baina barneratuta duen pakete bakarra da hau.

Taldean lan egiteko baliabideak eskaintzen ditu, hainbat garatzailek modu konkurrentean aldi berean lan egiteko, dokumentu edo fitxategi berean beste garatzaileek egiten dituzten aldaketak denbora errealean ikustea ahalbidetzen baitu.

Ez da datu-base bakarra erabiltzera mugatzen, garatzaileak berak nahiago duena aukera baitezake. Normalean framework honekin erabili ohi diren datu-baseen artean MongoDB eta MySQL daude, baina konfigurazio-aldaketa gutxi batzuekin beste edozeinekin ere bateragarria da.

¹⁵ <https://socket.io/> (kontsulta: 2017-08-08)



6. irudia: Socket.io-ren egitura.

TOTAL

Aztertu diren frameworken artean berriena da hau¹⁶. Hiru urte eskas ditu baina, jada, bere bigarren bertsioa kaleratu da eta hazkunde handia izaten ari da programatzaileen artean. Egitura aldetik, MVC patroia jarraitzen du.

Bezero aldean Angular.js, Polymer, Backbone.js eta gehiago erabil daitezke. Beraz, edozein frameworkek baino aukera zabalagoa ematen du txantiloien aldetik. Horrez gain, kodetzen hasteko adibide-eskaintza oso zabala du, bere webgune ofizialean ikus daitekeen bezala. NoSQL datu-base ez-erlazionala barneratzen du (*built-in*) eta hedagarria eta asinkronoa da.

Aurredefinitutako hainbat modulu eskaintzen ditu garapena erraztu eta kodetzea aurrezteko, baina denak ez dira doakoak. Doakoen artean saio-kudeatzaileak, errore-kudeatzaileak, bisita-zenbatzaileak eta beste hainbat aurki daitezke; beraz, nahiko aukera zabala eskaintzen du ikuspuntu honetatik. Gehigarri modura, WebSocket eta bideo-streaming-a ere onartzen ditu.

3.3.3 Konparaketaren laburpena eta ondorioak

Aipatutako ezaugarri guztiak laburbiltzeko, honako taulak osatu dira, irizpide multzo bakoitzeko bana ("H" bat duten ezaugarriak hautazkoak dira):

OSAGAIEI LOTUTAKO IRIZPIDEAK

	Bezeroan	Zerbitzarian	DBa
MEAN	Angular	Express NodeJS	MongoDB
METEOR	Angular (H)	NodeJS	MongoDB
MOJITO	-	NodeJS	HSQL MySQL
SOCKET	-	NodeJS	Aukerakoa
TOTAL	Angular, Polimer, Backbone... (H)	NodeJS	NoSQL barneratua

2. taula: Garapen-inguruneen osagaien laburpena.

¹⁶ <https://www.totaljs.com/> (kontsulta: 2017-08-08).

FUNTZIONALITATEEI LOTUTAKO IRIZPIDEAK

	MVC	Denbora errealeko aplikazioak	Testing-a	Aurredefinitutako paketeak	Aurredefinitutako txantiloak	Besterik
MEAN	MVVM	Ez	Bai	Ez	Bai	
METEOR	Bai (H)	Bai	Ez	Bai	Bai	
MOJITO	Bai	Ez	Bai	Ez	Ez	Internazionalizazioa
SOCKET	-	Bai	Ez	Mezularitza eta txata	Ez	Talde-lanerako baliabideak
TOTAL	Bai	Ez	Ez	Bai	Bai	Bideo-streaming-a

3. taula: Garapen-inguruneen funtzionalitate-ezaugarrien laburpena.

Egindako azterketa laburbilduz eta laburpen-taulari erreparatuz ikus daitekeen bezala, bada aztertutakoen artean garapen-ingurune bat besteek baino irizpide gehiago betetzen dituena: Meteor framework-a hain zuzen. Definitu diren bost irizpideetatik lau betetzen ditu, testing-a izanik horietatik barneratzen ez duen funtzionalitate bakarra.

Konparaketa honetan, Meteor beste inguruneetatik bereizi duena aurredefinitutako txantiloak eta pakete aukera zabala izatea izan da, Totalek bakarrik eskaintzen baitu horrelakoak erabiltzeko aukera. Aldiz, Totalek ez ditu denbora errealean eguneratzen diren aplikazioak sortzen.

Bukatzeko, aipatu beharra dago badirela zenbait ingurunek barneratzen dituzten bestelako funtzionalitate gehigarriak, oso interesgarriak hauek ere. Hala ere, azterketa honetan aurrez zehaztutako bost irizpideak soilik hartu dira kontutan, garrantzitsuenak direla erabaki delako. Funtzionalitate gehigarriak laburpen-taulan adierazi dira, ikuspegi orokor bat edukitzeko informazio interesgarria eskaintzen dutela erabaki baita dokumentu honen egilearen partetik.

Beraz, hurrengo orrialdeetan azalduko den web-aplikazioa garatzeko, Meteor frameworka erabiliko da. Inguruneari buruzko xehetasun gehiago azalduko dira hurrengo atalean.

4. METEOR

Atal honetan, aukeratutako garapen-inguruneari buruzko azalpen eta argibide orokor batzuk emango dira.

Meteor kode irekiko garapen-ingurune isomorfikoa da, bereziki denbora errealeko aplikazioak sortzeko pentsatua eta sistema eragile anitzekin bateragarria: kode berak Android, iOS, web edo mahagaineko aplikazioak sortzeko balio dezake. Aurrez aipatu den bezala, hainbat teknologia eta tresna erabiltzeko aukera ematen du, hala nola Angular, Express eta NodeJS. Ingurune honen lehen bertsioa 2012ko urtarrilean kaleratu zen¹⁷.

Garapen-ingurune hau Javascript, C eta C++ lengoaiak erabiliz idatzi du *Meteor Development Group* garatzaile-taldeak, baina bere funtzionalitate nagusia diren aurredefinitutako paketeak framework-aren komunitate gero eta handiagoak (garatzaile eta erabiltzaile independenteez) garatzen ditu, oro har. Pakete hauek komando soil batekin instala daitezke, eta garatzaileek sarean argitaratu ohi dituzte (GitHub plataforman normalean) hauek erabiltzeko argibideak eta azalpenak.

Paketeen gaian zerbait gehiago sakonduz, aipatu behar da Meteor-en barne-egitura bera berezko hainbat paketetan banatuta dagoela, garatzaileak horietaz arduratu behar ez badu ere. Horietaz gain, beste lau pakete mota¹⁸ bereizten dira ingurunean. Badira funtzionalitate anitz eskaintzen dituzten hainbat pakete; horietako batzuk “First-party” izenekoak dira eta Meteor-en garatzaileek kudeatzen dituzte, eta besteak komunitatetik sortutakoak dira (aurreko paragrafoan aipatutakoak). Badira ere pakete lokalak, erabiltzaile bakoitzak sortu eta dagokion “*packages*” karpetan koka ditzakeenak, sortzen ari den aplikazioan erabiltzeko. Azkenik, aurreko paketeetatik batzuk NodeJS-ren NPM paketeekin konbina daitezke.

Meteor-en egiturarekin jarraituz, ingurunearen zerbitzariko aldeak bakarrik du datu-basea atzitzeko aukera, aplikazioaren eta datuen segurtasuna bermatu ahal izateko. Hala ere, inguruneak datu-basearen kopia bana du bai zerbitzariaren eta bai bezeroaren aldean, eta horrela lortzen da bi aldeak beti eguneratuta egotea eta denbora errealeko aplikazioak sortzea. Bezeroaren aldeko informazioa zerbitzariaren aldekoarekin eguneratua mantentzea DDP protokoloaren erabilerekin lortzen da.

Datu-baseko eguneraketok, berdin aplikatzen dira HTML, Javascript edo CSSan egiten den edozein aldaketako. Gainera, edozein aldaketa dagoenean, Meteor-ek mementoan erakusten ditu, jada gordeta egongo balira bezala. Gero, konexioa eten edo beste edozein arazo gertatzen bada, erabiltzaileak jakinarazpen-mezu bat jasotzen du eta aldaketak deuseztatzen dira.

Datu-basearekin jarraituz, framework honek bezeroak datuak zerbitzaritik atzitzeko metodo berezi bat proposatzen du: argitaratze-harpidetza sistema¹⁹. Modu honen bitartez, zerbitzariak datu-baseko bildumak (*collections*) argitaratzen ditu (*publish*) bezero txantiloiak behar dituztenetara harpidetu daitezen (*subscribe*). Zerbitzari aldean, gainera, bezerotik datuak moldatu ahal izateko, baimen bereziak eman behar zaizkio, `allow` klausula

¹⁷ <https://github.com/meteor/meteor/commit/4e4358e2c068ffd0c0bc565bb2a9438c68bab7fb> (kontsulta: 2017-08-07).

¹⁸ <http://es.discovermeteor.com/chapters/getting-started/> (kontsulta: 2017-08-07).

¹⁹ <https://www.meteor.com/tutorials/blaze/publish-and-subscribe> (kontsulta: 2017-06-20).

erabiliz. Allow klausularen bitartez, idazteko, irakurtzeko edo ezabatzeko baimena ematen zaio bezeroari, bilduma zehatz baten gainean.

```
Meteor.publish('collection', function() {
  return Collection.find();
});

Template.random.onCreated(function() {
  Meteor.subscribe('collection');
});
```

Harpidetutako datuak HTML orrian bistartzeko, Meteor-ek Spacebar-ak erabiltzen ditu. Spacebar-ak²⁰ HTMLk zerbitzaritik datozen datuak bistartzeko normalean erabiltzen dituen Handlebar-en aldaera berezi bat dira. Meteor-en barne-egituran egin zaizkien aldaketa batzuei esker, Spacebar-ek denbora errealean egiten dituzte eguneraketak, beste garapen-ingurune batzuetatik Meteor bereiziz. Hurrengo kode zatian erakutsiko den Spacebar-en erabilerarekin, esaterako, txantilo baten barruan beste bat txerta daiteke, eta honek zerbitzaritik hartzen dituen datuak erakutsi:

```
<template name="random">
  <div>
    {{#each randoms}}
      <div class="zerrenda">
        {{> random2}}
      </div>
    {{/each}}
  </div>
</template>

<template name="random2">
  <span class="text"><strong>{{izenburua}}</strong>
  <br/>
  {{testua}}
</span>
</template>
```

Meteor-ek erabiltzen duen Javascript kodearen barruan, 3 elementu nagusi bereiz daitezke: gertaerak, laguntzaileak eta hasieraketak (events²¹, helpers²², onCreated²³). Helpers-en bidez, hainbat funtzio edo aldagai laguntzaile moduko sor daitezke, ondoren HTML orrian datuak atzitzeko eta bistartzeko erabil daitezkeenak. Laguntzaile hauek boolearrak izan daitezke edota bilduma bateko elementu guztiak gordetzen dituztenak, esaterako. Events atalean hainbat gertaera defini daitezke, erabiltzaileak sortutako inputei erantzuteko (botoiak zapaltzea, formularioa bidaltzea...).

²⁰ <http://blazejs.org/guide/spacebars.html> (kontsulta: 2017-05-22).

²¹ <https://www.meteor.com/tutorials/blaze/forms-and-events> (kontsulta: 2017-05-16).

²² <https://www.meteor.com/tutorials/blaze/templates> (kontsulta: 2017-05-16).

²³ <https://docs.meteor.com/v1.3.5/api/templates.html> (kontsulta: 2017-05-16).

Azkenik, `onCreated` atalean, beharrezko hasieraketak egiten dira txantiloientzat, adibidez, zerbitzariko argitalpenetara harpidetzea.

```
Template.txantiloizena.onCreated(function() {
  //hasieraketak
});

Template.txantiloizena.helpers({
  helper1() {
    //kodea
  },
});

Template.txantiloizena.events({
  'click .random'() {
    //kodea
  }
});
```

Aurrez aipatu den bezala, Meteor-ek karpeta-egitura finkatzerako garaian malgutasun handia ematen du, izan ere, hainbat berezko karpeta dituen arren (3. atalean aipatutakoak), `lib` izeneko karpeta da derrigorrez jarri behar den bakarra, beharrezko hasieraketak bertatik irakurtzen baititu. Argazkiak eta bestelako elementu estatikoak erabili nahi badira, hauek `public` izeneko karpetan gordetzea beharrezkoa, modu egokian kargatu daitezten.

Bukatzeko, datu-basearen funtzionamendu orokorra azalduko da. Aipatu bezala, Meteor-ek ez-erlazionala den MongoDB erabiltzen du. Honek esan nahi du ez dagoela ez datu-klaserik ez erlaziorik. Datu-basean `Collections`²⁴ edo bildumak sor daitezke, kode-lerro sinple batekin eta inolako atributurik sortu gabe (bilduma bakoitzari beranduago txertatu ahal zaizkio nahi diren ezaugarriak). Datu-baseko datuak atzitzeko, nahiko da, lehen esan den bezala, zerbitzariak bilduma zehatz bat argitaratzea eta bezeroa bilduma horretara harpidetzea.

```
import { Mongo } from 'meteor-mongo';
Bilduma = new Mongo.Collection('bilduma');
```

Ondoren bilduma horri “atributuak” gehitzeko, nahikoa da elementu berri bat sortzen duen metodoaren barruan (zerbitzariaren aldean) ezaugarri horiek txertatzea.

```
sortuElementua: function(ezaugarria){
  Bilduma.insert({
    ezaugarria: ezaugarria
  });
}
```

²⁴ <https://docs.mongodb.com/v3.2/core/databases-and-collections/> (kontsulta: 2017-05-03).

5. WEB-APLIKAZIOAREN ANALISI ETA DISEINUA

Atal honetan, web-aplikazioa gauzatzeko egindako analisiaren eta diseinuaren gorabeherak azalduko dira.

5.1 ANALISIA

5.1.1 Eskakizunen bilketa

Bezeroak, web-aplikazioa eraikitzeke proposamena helarazteko garaian, hainbat ezaugarri adierazi zituen prototipoan garatu beharko liratekeenak bezala, berarekin egindako bilera batean. Aurrez 2. atalean aipatu den bezala, eskakizun hauek ez ziren zorrotzak izan eta bezeroak funtzionalitateen aldetik malgua izateko aukera jakinarazi zion garatzaileari, hau da, funtzionalitateen bat garatzeko bideragarria kontsideratzen ez bazen, hau moldatu, berregituratu edo alde batera uzteko aukera jarri zen mahai gainean.

Jarraian, bezeroaren hasierako eskakizunak deskribatuko dira berak helarazitako modu berean, eta hurrengo atal batean egindako aldaketak aipatuko dira:

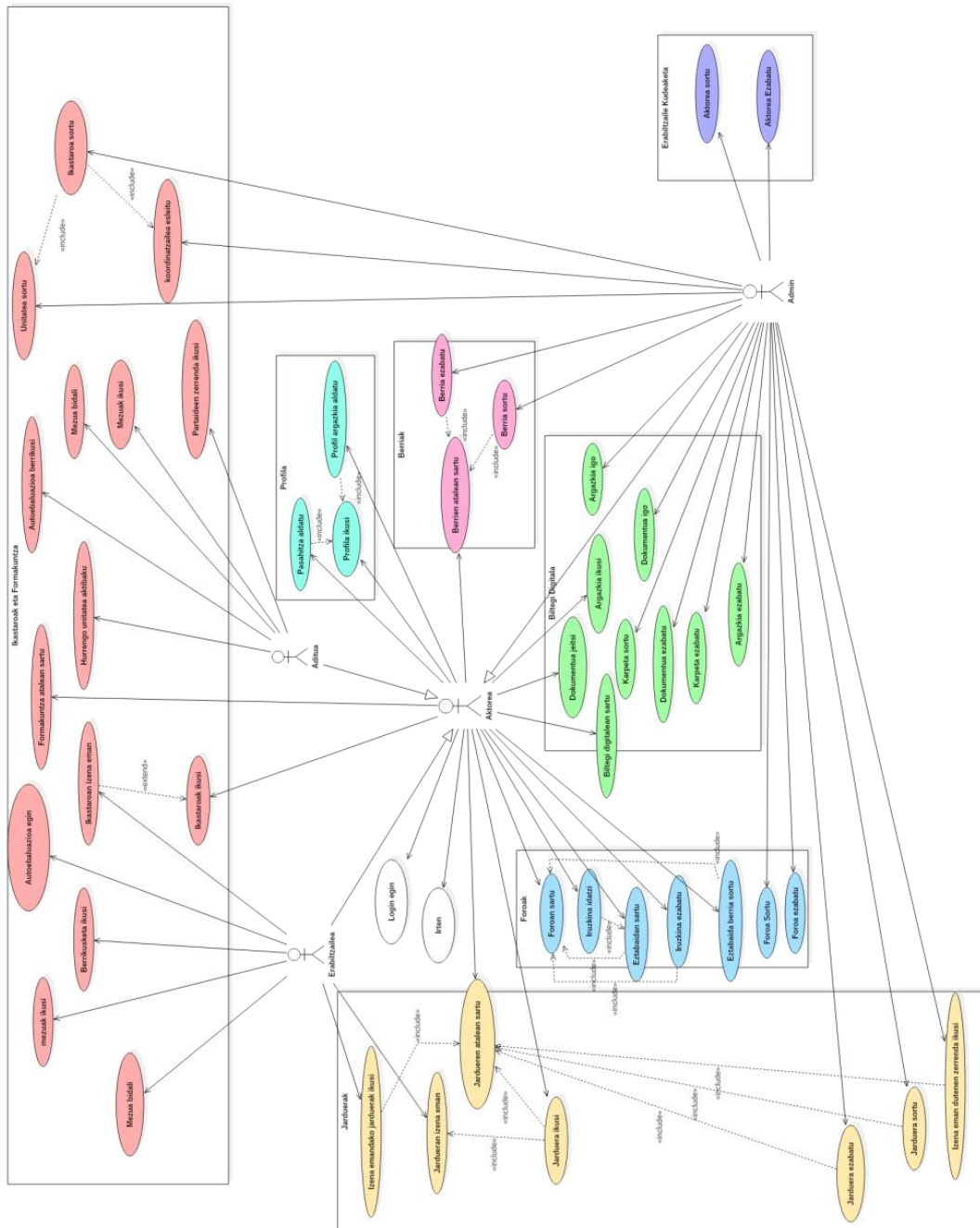
- Dagoeneko eraikitze-prozesuan dagoen Gautenaren web-orri berriari txertatu ahal izango den web-aplikazio bat sortzea.
- Ez du zertan mugikorrerako egokitua izan, familiek ordenagailutik atzitu baitute normalean. Denbora aldetik posible bada, hala ere, ez legoke gaizki mugikorrerako egokitzea.
- Euskaraz eta gaztelaniaz erabiltzeko aukera izatea.
- Web-aplikazioak honako atal eta funtzionalitateak izatea:
 - Rol-bereizketa egotea, 3 erabiltzaile mota bereiziz: erabiltzaile arrunta, administratzailea eta aditua.
 - Foro-atal bat izatea. Foro bat baino gehiago sortzeko aukera izatea eta 3 roletako erabiltzaileek bertan idatzi ahal izatea. Administratzaileak bakarrik edukiko du foro eta iruzkinen sortzea eta ezabatzea kudeatzeko aukera.
 - Jarduera-atal bat izatea. Bertan erabiltzaile arruntek (soilik) Gautenaren jardueretan izena emateko aukera izan beharko lukete. Jardueren kudeaketak administratzailearen esku egon beharko du.
 - Biltegi digital bat edukitzea, hainbat argazki eta dokumentu biltegitzeko. Erabiltzaile mota guztiek atzitu ahal izango dituzte fitxategiak baina administratzaileak bakarrik kudea ditzake.
 - On-line ikastaroak egiteko atal baten sorrera. Bertan aditu eta erabiltzaile arruntek hartuko dute parte, eta administratzaileak sortu eta esleituko dizkie adituak ikastaroei. Atal honetan, hainbat zeregin proposatuko dira asteko, eta erabiltzaile arruntek hauek osatzeko eta bidaltzeko aukera izango dute, dokumentuak jaitsi eta igoz. Adituak, ondoren, zereginak berrikusi eta hurrengo fasera pasatzeko aukera emango die erabiltzaileei.
 - Administratzaileak erabiltzaileak kudeatu ahal izateko aukera izatea (sortu, ezabatu, baimenak eman...), web-aplikazioko atal zehatz batetik.
 - Familientzako gunean sortzen diren hobekuntzak eta bestelako oharren argitaratzeko berriari atal bat egotea, non administratzaileak bakarrik idatzi ahal izango duen eta rol guztiek irakurri.

Azkenik, bezeroaren eskakizunetan ez zegoen arren, erabiltzailearen profila ikusteko aukera ere gehitzea erabaki da, estilo honetako web-aplikazio guztiek ematen duten aukera baita, pasahitza-aldaketa sinple bat burutzeko.

5.2 DISEINUA

5.2.1 Erabilpen-kasuen diagrama

Behin bezeroaren eskakizunak jasota, funtzionalitate eta ezaugarri horietarako beharrezkoak diren erabilpen-kasuak zehaztu dira, 3 rol motak aintzat hartuz. Jarraian ikus daiteke erabilpen-kasuen diagrama:



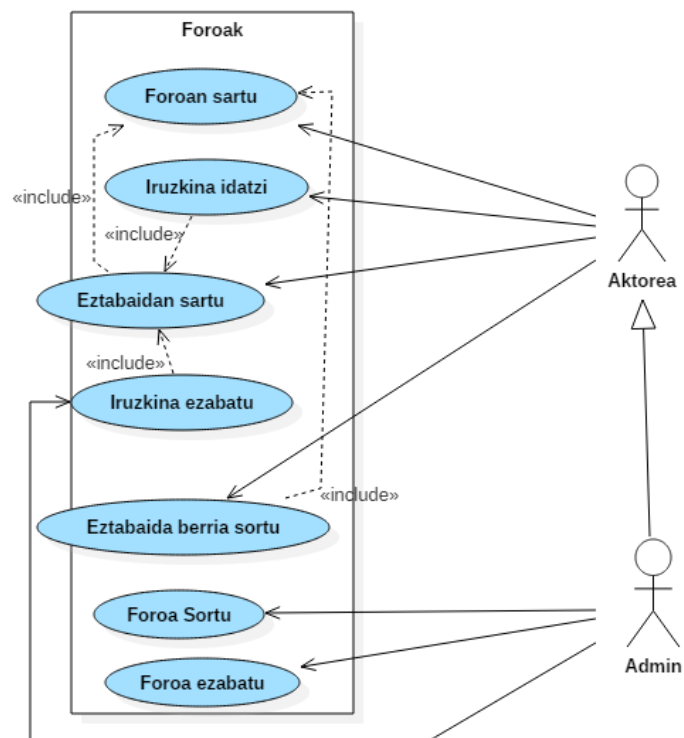
7. irudia: Erabilpen-kasuen diagrama.

Ikusten den moduan, erabilpen-kasurik web-aplikazioaren atalen arabera multzokatuta (sistemen bidez) eta kolore ezberdinez margotu dira, hauen identifikazioa errazagoa izateko. Diagrama nahiko handia denez, jarraian atal bakoitza bere aldetik aurkeztuko da, xehetasun guztiak modu argian bistartzeko.

Ikusten den bezala, hiru aktore mota bereiz daitezke diagraman, bakoitza rol bati dagokiona; "Aktorea" aktorearen espezializazioak dira hirurak. Hala ere, "Aktoreak" aktore mota mantendu da diagraman, 3 rolerik egin ditzaketen erabilpen-kasuetan erlazioak osatzeko. Horrela, gezi asko aurreztu eta diagrama txukunago mantentzen da.

Aipatu beharra dago ere, goiko diagraman adierazi ez den arren, bertako erabilpen-kasu guztiak “Login egin” erabilpen-kasuaren menpekoak direla, baina, erabilpen-kasu horrek besteekiko duen “<<include>>” lotura eliditu dela diagraman, honen ulergarritasuna bermatze aldera, gainjartzen diren gezi mordoak ekidinez. Berdina gertatzen da “Ikastaroak eta formakuntza” eta “Biltegi digitala” sistemekin, non “Formakuntza-atalean sartu” eta “Biltegi digitalean sartu” erabilpen-kasuak egitea derrigorrezkoa den, gainontzekoak egin ahal izateko. Kasu hauetan ere ez da adierazi erlazioen bitartez diagrama ulergarri mantendadin.

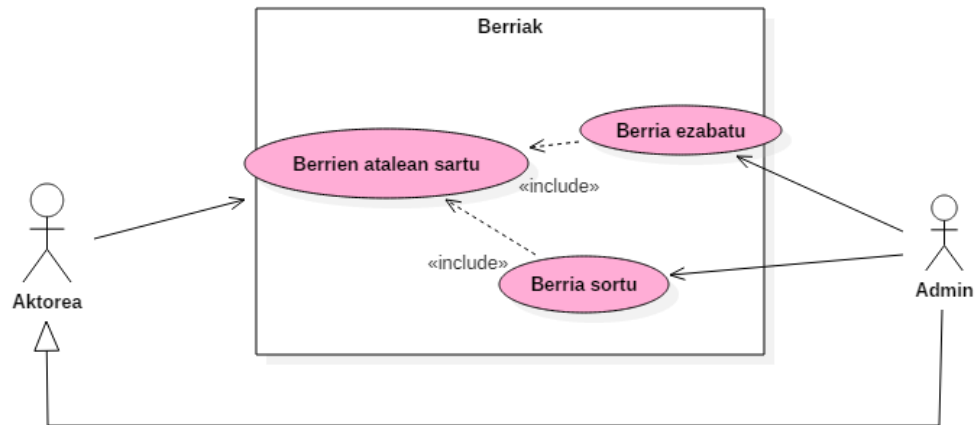
FOROAK ATALA



8. irudia: “Foroak” atalaren erabilpen-kasuen diagrama.

Diagraman ikus daitekeen moduan, 3 aktore motek parte hartzen dute atal honetako erabilpen-kasuetan. Modu berean, parte hartzen dute mezu-trukaketan oinarritzen diren erabilpen-kasuetan eta foroen sorrera eta ezabaketa administratzailearen eginkizuna dira eskusiboki.

BERRIAK ATALA



9. irudia: "Berriak" atalaren erabilpen-kasuen diagrama.

Aktore guztiek dute berriak ikusteko aukera baina administratzaileak soilik hauek kudeatzekoa.

IKASTAROAK ETA FORMAKUNTZA ATALA

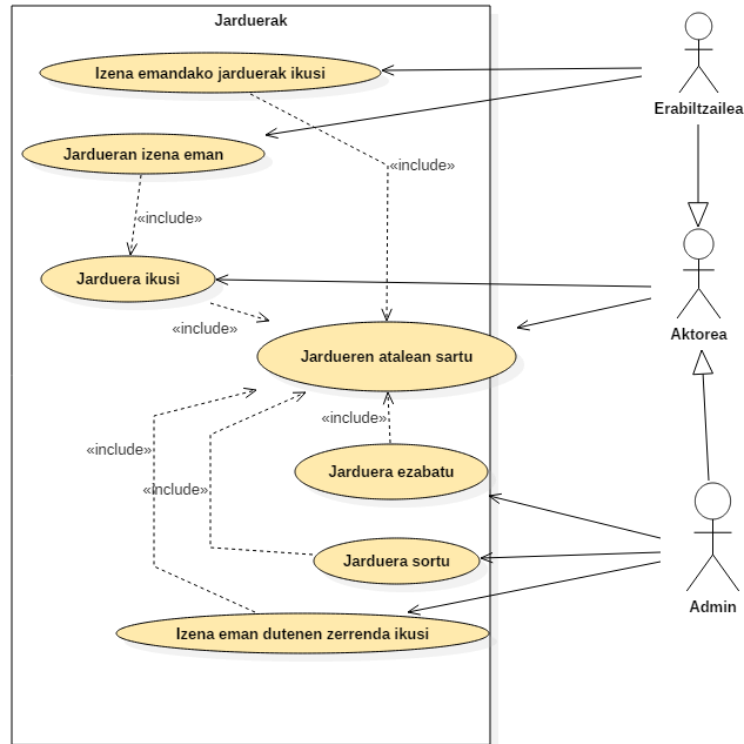


10. irudia: "Ikastaroak eta formakuntza" atalaren erabilpen-kasuen diagrama.

Atal hau da denetan konplexuena, aktore guztiek baitutuzte erabilpen-kasu propioak. Hasteko, administratzailea arduratzen da, normalean bezala, ikastaroak sortzeaz eta, kasu honetan, unitateak sortzeaz gain, koordinatzaile bat esleitzen du ikastaroko, aditu bat alegia. Adituek ikastaroetako partaideekin harreman zuzena dute eta mezuak truka ditzakete. Adituak berrikusten ditu partaideek osatutako autoebaluazioak eta hurrengo unitatea partaide jakinentzat erabilgarri jartzeko aukera du. Erabiltzaile arruntek, adituekin mezuak trukatzearaz gain, hauek esleitutako jarduerak egiten dituzte.

Oharra: "Mezua bidali" eta "Mezuak ikusi" erabilpen-kasuak errepikatuta agertzen dira goiko diagraman. Hori geziaren gainjartzea ekiditeko egin da. Erabiltzaile arruntek eta adituek dute mezuak bidali eta ikusteko aukera.

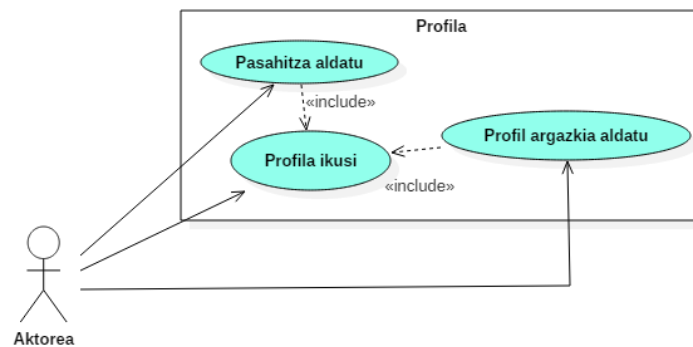
JARDUERAK ATALA



11. irudia: "Jarduerak" atalaren erabilpen-kasuen diagrama.

Atal honetako erabilpen-kasuetan erabiltzaile eta administratzailek parte hartzen dute nagusiki, adituek jarduerak ikusteko aukera baitute soilik. Erabiltzaile batek jarduera batean izena eman dezake eta bere jardueren zerrenda ikusi. Administratzaileak jardueren sortze eta ezabatzea kudeatzen du, eta jarduera bakoitzean izena eman duten erabiltzaileen zerrenda ikus dezake.

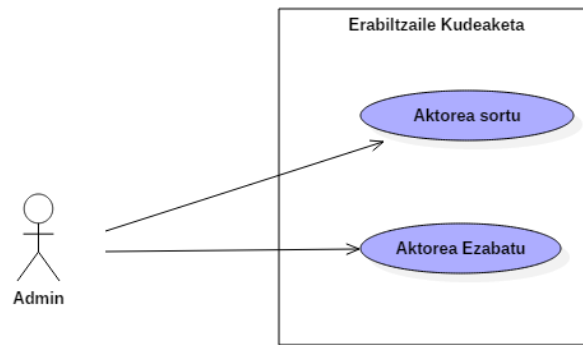
PROFILA ATALA



12. irudia: "Profila" atalaren erabilpen-kasuen diagrama.

Aktore guztiek dute atal honetako erabilpen-kasu guztiak egiteko baimena, alegia, profila ikusi eta pasahitza edo profileko argazkia aldatzea.

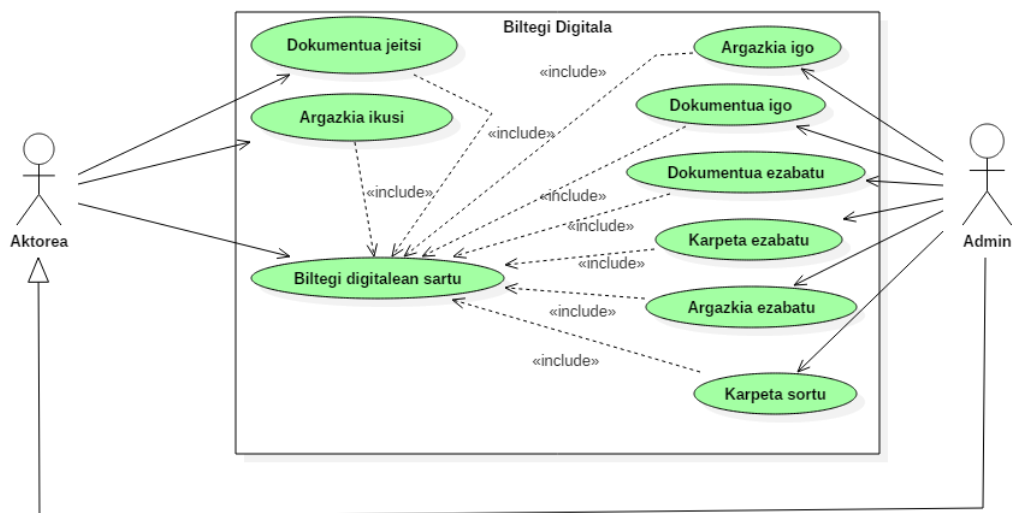
ERABILTZAILE-KUDEAKETA ATALA



13. irudia: "Erabiltzaile-kudeaketa" atalaren erabilpen-kasuen diagrama.

Aurrez aipatu bezala, administratzaile motako aktoreak edukiko dute aukera beste aktore batzuk sortu edo ezabatzeke.

BILTEGI DIGITALA ATALA



14. irudia: "Biltegi digitala" atalaren erabilpen-kasuen diagrama.

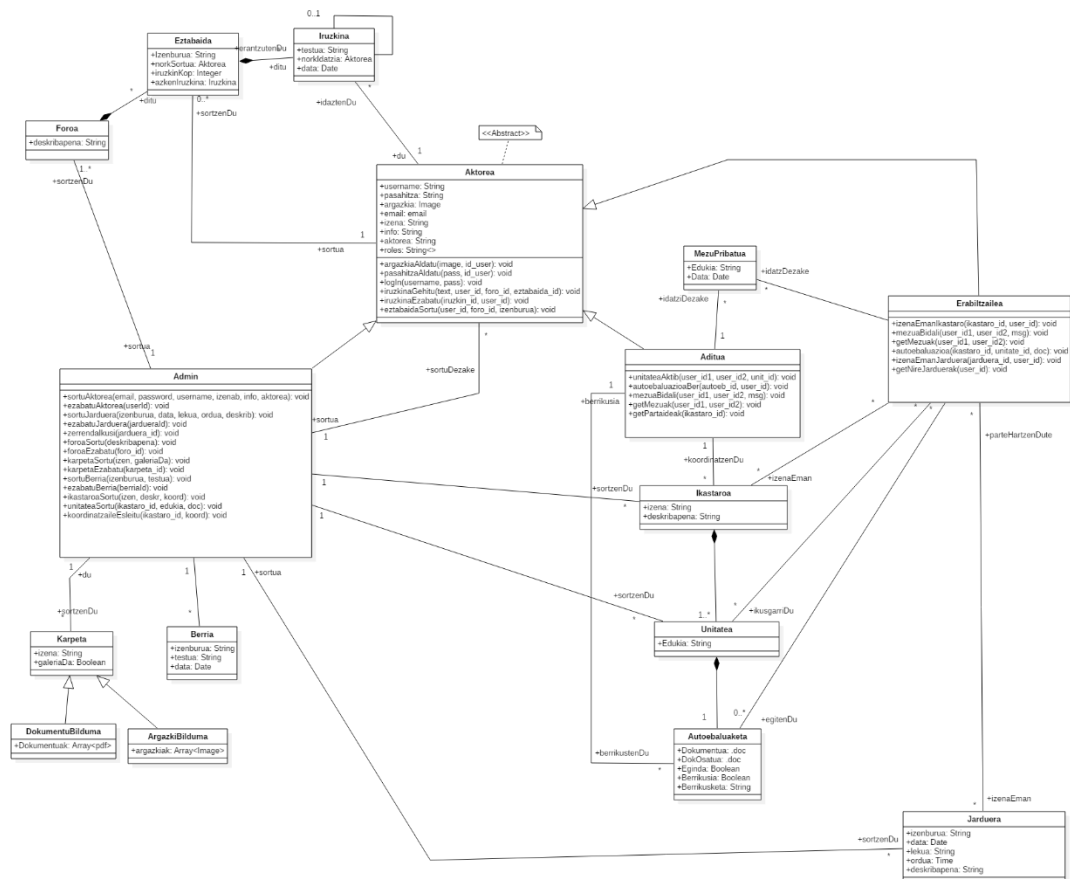
Atal honetan administratzailea da gehien parte hartzen duen aktore mota, dokumentuak sortu eta karpetak eta dokumentuak igo eta ezaba baititzake. Biltegi karpetak eta fitxategiak kontsultatzeko aukera berdina da aktore mota guztientzat.

5.2.2 Klase-diagrama

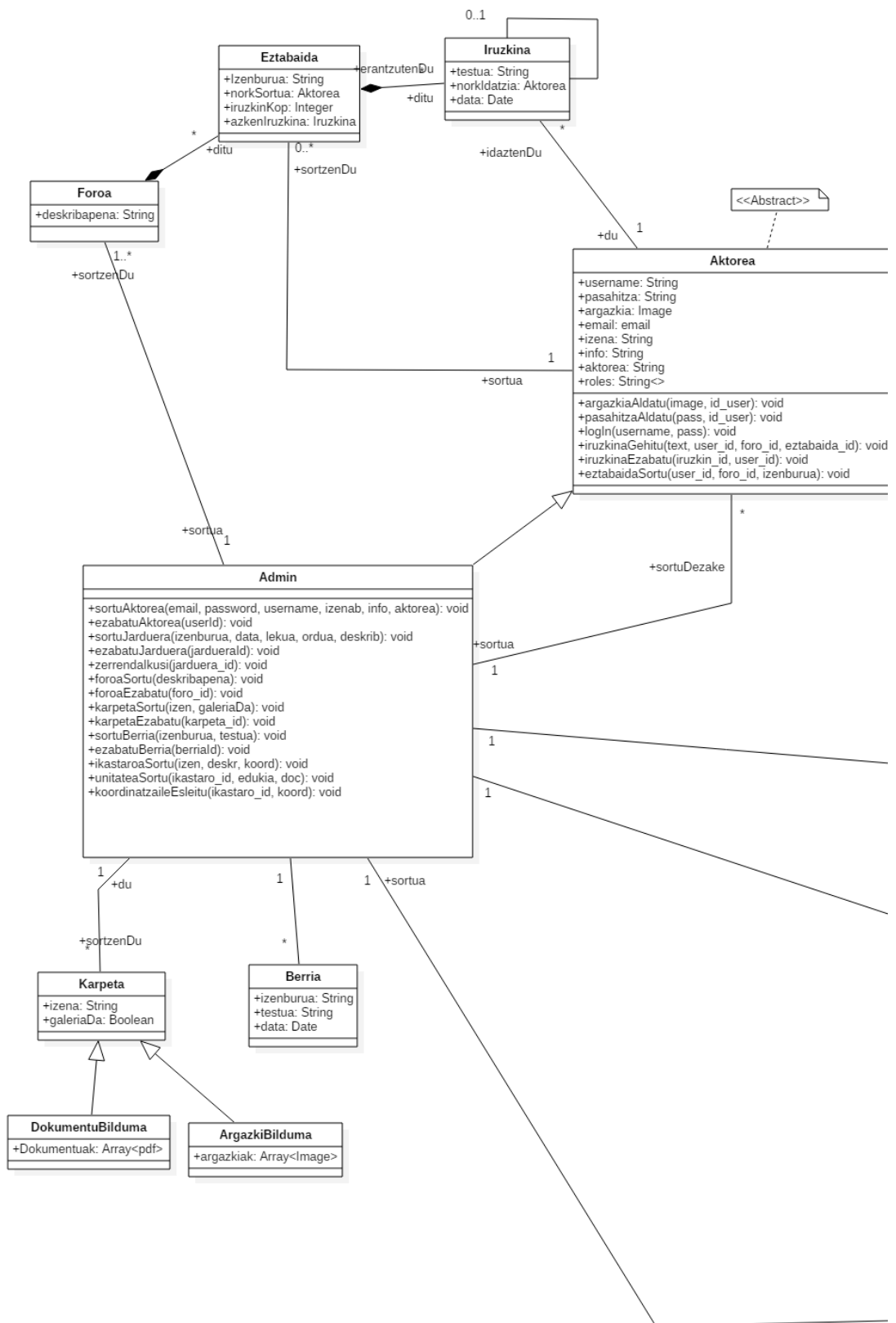
Jarriaian aplikazioaren klase-diagrama erakusten da. Aipatu behar da diagrama hau diseinu mailakoa dela soilik, hau da, aplikazioan klaseak ez dira eraiki diagraman ageri den bezala, datu-base ez erlazional batean ezin direlako horrela gauzatu. Diagramak aplikazioaren ikuspegi orokor bat egiteko balio du (eta garatzaileari balio izan dio) eta diagrama honetako klaseak nola implementatu diren 6. atalean azalduko da.

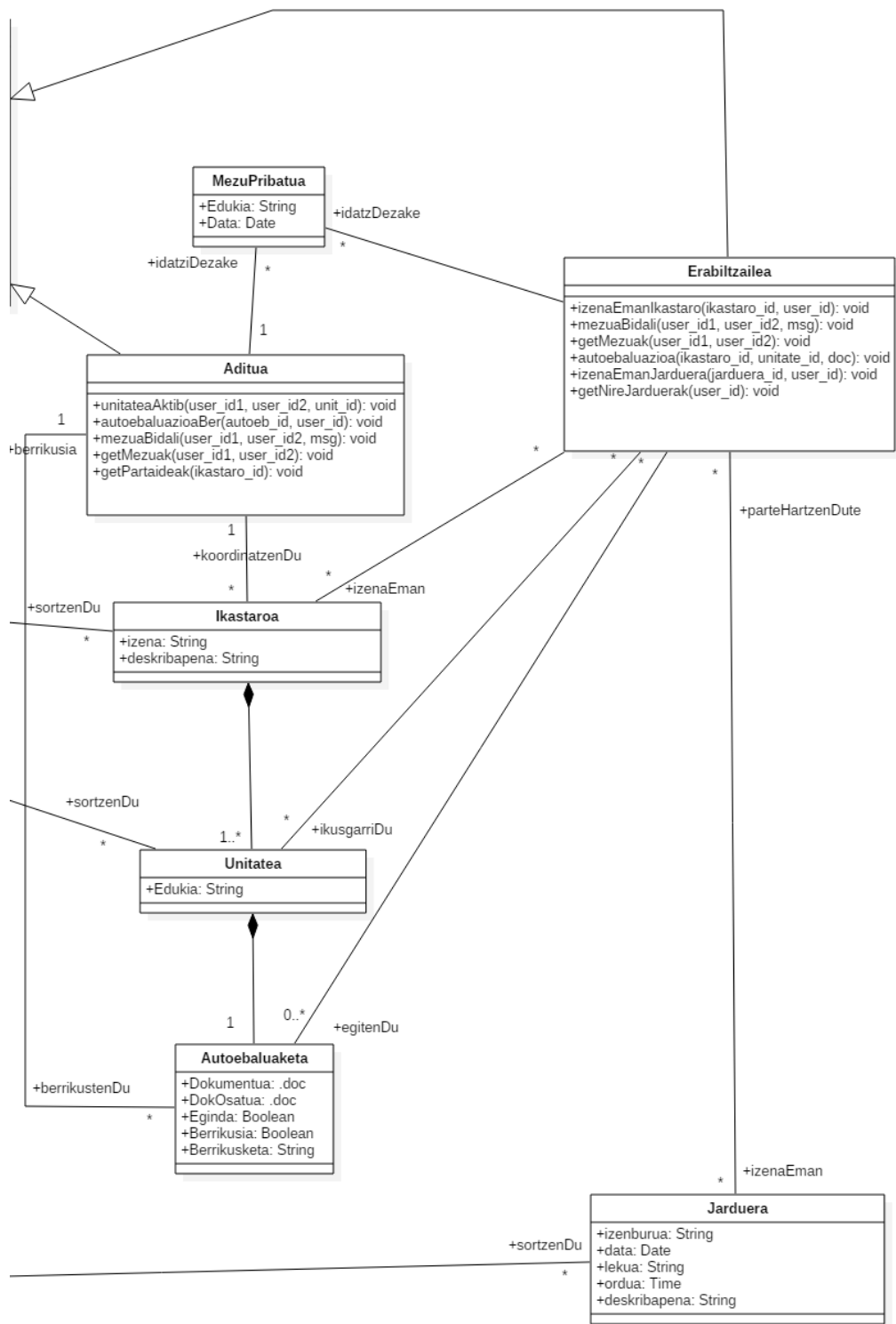
Klase nagusik 3 aktore motak dira, eta hauek dituzte esleituta erabilgarri izango dituzten metodoak. Inplementazioan, noski, beste toki batean joango dira metodo horiek, aurreko paragrafoan azalduko arrazoi beragatik.

Txosteneko 40 eta 41 orrialdeetan diagrama handituta aurkezten da, xehetasun guztiak modu argi batean irakurri ahal izateko.



15. irudia: Web-aplikazioaren klase-diagrama.



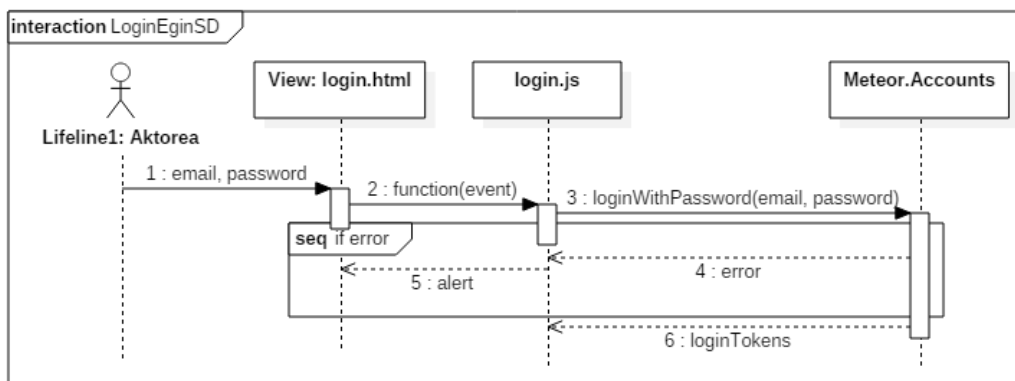


5.2.3 Sekuentzia-diagramak

Jarraian, erabilpen-kasu esanguratsuenen sekuentzia-diagramak aurkeztuko dira eta, hauekin batera, haien gertaera-fluxuak. Web-aplikazioaren erabilpen-kasuen kopurua nahiko handia denez, atal honetan aurkezteko adierazgarrienak direnak aukeratu dira, eta gainerakoak memoriaren C eranskinean txertatuko dira.

Bestalde, hainbat erabilpen-kasu antzekoak dira euren artean. Kasu horietan, antzekoak direnetatik bakarra aurkeztuko da eta baliokideak direnak aipatuko dira. Esaterako, "Foroa sortu" eta "Karpeta sortu" erabilpen-kasuek (eta beste hainbat) sekuentzia-diagrama eta gertaera-fluxu ia berak dituzte, erabiltzaileak sartzen duen informazioa eta aukeratutako metodoen izenak izanik desberdintasun bakarrak.

LOGIN EGIN



16. irudia: "Login egin" erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Aktore batek web-aplikazioan login egiten du.

Aktorea: Aktorea (erabiltzaile arrunta, administratzailea edo aditua).

Aurrebaldintzak: Erabiltzaileak aplikazioa martxan izatea.

Postbaldintzak: Aktoreak saioa hasiko du aplikazioan, datu zuzenak sartzen baditu.

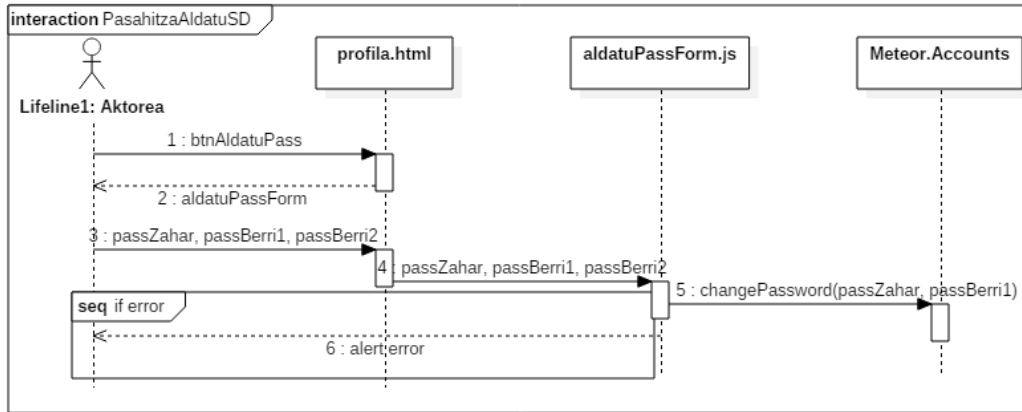
Ohiko fluxua:

1. Erabiltzaileak posta eta pasahitza sartu eta "Login" botoia sakatu.
2. Aplikazioak saioa hasiko du eta menu nagusira birbideratuko du erabiltzailea.

Ezohiko fluxua:

3. Datuak ez badira zuzenak, errore mezua bistaratuko da pantailan.

PASAHITZA ALDATU



17. irudia: "Pasahitza aldatu" erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Aktore batek bere profil-orritik pasahitza aldatzen du.

Aktorea: Aktorea (erabiltzaile arrunta, administratzailea edo aditua).

Aurrealdintzak: Erabiltzailea profil-atlean sartuta egotea.

Postbaldintzak: Erabiltzailearen pasahitza aldatuta.

Ohiko fluxua:

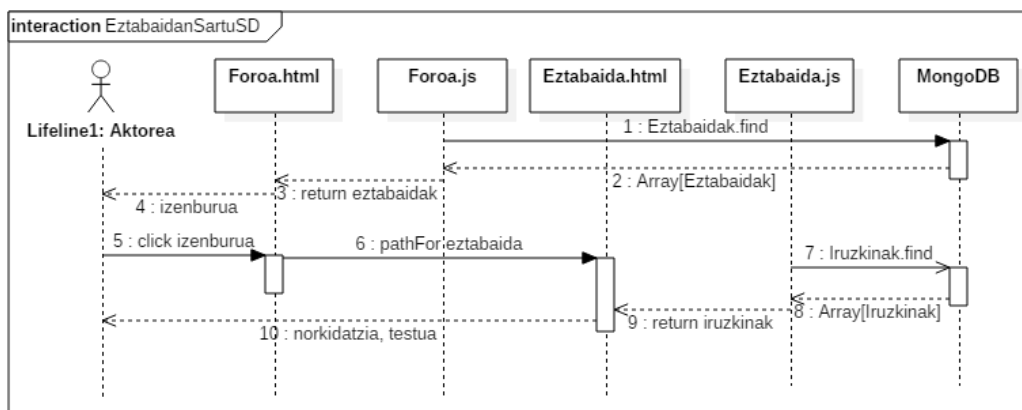
1. Pasahitza aldatu botoia sakatu.
2. Formularioa agertzean, pasahitza zaharra eta berria (bi aldiz) sartu eta baieztatze botoia sakatu.
3. Pasahitza aldatuko da datu-basean.

Ezohiko fluxua:

4. Datuak ez badira zuzenak (pasahitza zaharra okerra edo berriak ez datoz bat), errore mezua bustaratuko da pantailan.

EZTABAIDAN SARTU

Erabilpen-kasu honen sekuentzia-diagrama baliokidea da hurrengo erabilpen-kasuenekin: "Foroan sartu", "Biltegi digitalean sartu", "Berrien atlean sartu", "Profila ikusi" eta "Jardueren atlean sartu".



18. irudia: "Eztabaidan sartu" erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Aktore bat foro zehatz bateko eztabaida batean sartzen da.

Aktorea: Aktorea (erabiltzaile arrunta, administratzailea edo aditua).

Aurrealdintzak: Saioa hasita foro batean sartuta egotea.

Postaldintzak: Aukeratutako eztabaidako iruzkinak ikusiko dira.

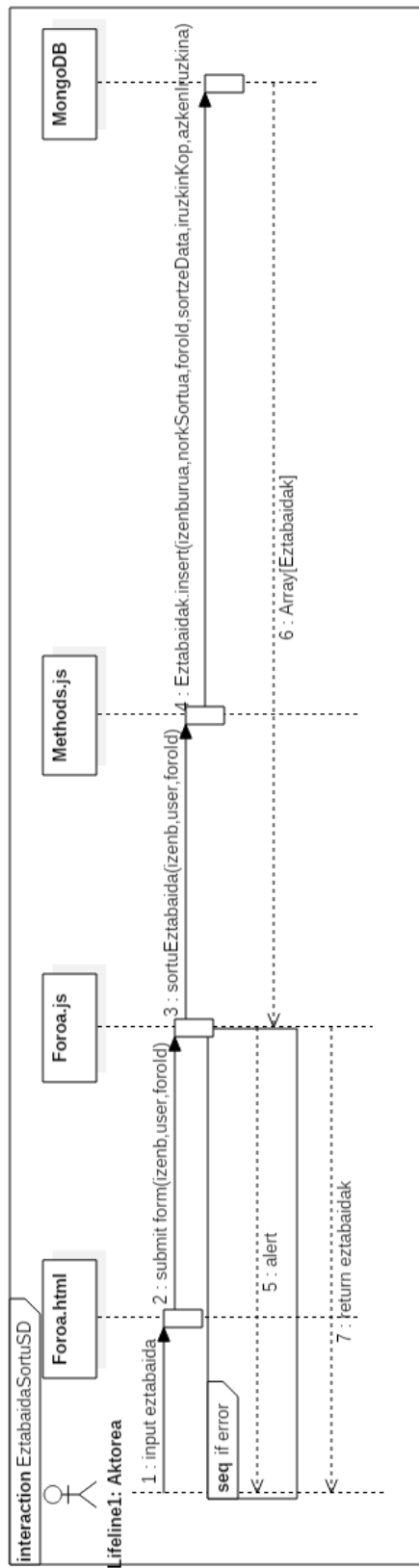
Ohiko fluxua:

1. Erabiltzaileak, foro batean sartuta, foro horretako eztabaidak ikusiko ditu pantailan.
2. Horietako bat aukeratu eta eztabaida horren orrira birbideratuko da.
3. Eztabaida horri dagozkion iruzkinak bistaratuko dira pantailan.

Ezohiko fluxua: -

EZTABAIDA SORTU

Erabilpen-kasu honen sekuentzia-diagrama baliokidea da hurrengo erabilpen-kasuenekin: "Foroa sortu" (admin), "Iruzkina idatzi", "Jarduera sortu" (admin), "Aktorea sortu" (admin), "Karpeta sortu" (admin) eta "Berria sortu" (admin).



19. irudia: "Eztabaida sortu" erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Aktore batek eztabaida berri bat sortzen du foro zehatz baten barruan.

Aktorea: Aktorea (erabiltzaile arrunta, administratzailea edo aditua).

Aurrealdintzak: Saioa hasita eta foro batean sartuta egotea.

Postaldintzak: Eztabaida berria bistaratuko da foroaren orrian.

Ohiko fluxua:

1. Erabiltzailea eztabaida berri baten izena sartu eta "Eztabaida gehitu" botoia sakatzen du.
2. Eztabaida berria gehituko da datu-basean eta aplikazioan bistaratuko da.

Ezohiko fluxua:

3. "Eztabaida gehitu" botoia testurik sartu gabe sakatzen bada, errore mezu bat agertuko da pantailan.

6. INPLEMENTAZIOA

Atal honetan, web-aplikazioaren implementazioaren nondik norakoak eta lortutako emaitzak azalduko dira.

6.1 APLIKAZIOAREN EGITURA ETA INPLEMENTAZIOAREN NONDIK NORAKOAK

6.1.1 Egitura orokorra²⁵

Web-aplikazioetan ohikoa den bezala, garatu den web-aplikazioan ere fitxategi ezberdinak karpeta sailkatzen dira. Kasu honetan, Meteor garapen-inguruneak hainbat karpeta “gako” ditu, barne-fitxategiak ordenatzeko garaian.

Hasteko, *client* eta *server* karpetak bereizten dira. Izenek adierazten duten bezala, *client* karpeta barruan dauden fitxategi guztiak bezeroan exekutatu dira, eta *server* karpeta barruan daudenak zerbitzarian. Gune pribatuaren aplikazioaren kasu zehatzean, *client* karpeta barruan *templates* izeneko azpikarpeta bat sortu da, eta bertan gordetzen dira HTML txantilo guztiak eta hauen logika osatzen duten Javascript fitxategiak. HTML eta JS fitxategiak karpeta berean gordetzea ez da beharrezkoa, baina asko errazten ditu gauzak, ez baitira deiak edo `import`-ak egin behar. *Server* karpeta, aldiz, zerbitzarian exekutatu behar diren fitxategiak daude, hala nola, *publish.js* (bildumak bezeroari eskuragai jartzeko), *account-creation.js* (aktore bat sortzerakoan konfigurazio gehigarriak ezartzeko) edo *methods.js* fitxategi nagusia. Fitxategi horretan, bezero fitxategiek deitzen dituzten metodo guztiak gordetzen dira, bildumak sortu, ezabatu eta beste hainbat gauza egiten dituztenak.

6.3 atalean, aurreko paragrafoan aipatu diren hainbat atalen kode zatiak atxikitu eta azalduko dira, hala nola: HTML eta JS orrien egitura, bildumen argitaratzea, elementuen txertaketa eta ezabaketa...

Aipatu behar da ere, *client* eta *server* karpeten arteko bereizketa egitea ez dela derrigorrezkoa, klausula zehatz batzuk erabiliz (*isServer* eta *isClient*) posible baita kode-lerro zehatz bat non exekutatu behar den adieraztea. Hala ere, proiektu honetan karpeta-bereizketa hori egitea egokiagoztat jo da, txukuntasunaren aldetik, batez ere.

Lib karpeta aplikazioak beharrezko dituen hasieraketak gordetzen dira, oro har. Gure kasuan, datu-baseko bilduma bakoitzeko fitxategi bat dago, bilduma sortu eta hasieratzen duena. Hasieraketak horrela egiten dira:

```
import {Mongo} from 'meteor-mongo';
Jarduerak = new Mongo.Collection('jarduerak');
```

Bestalde, *lib* karpeta *routing*-a (orrien arteko nabigazioa) egiteko fitxategia dago ere, *router.js* izenekoa. Bertan, helbide edo *ruta* bakoitzerako zein txantilo kargatu behar den esaten zaio aplikazioari. Routing sinpleena ondoko kode-lerroak erabiliz egiten da (6.3 atalean birbideratze konplexuagoak nola egin diren azalduko da ere):

```
Router.route('/jarduerak', {
  name: 'jarduerak'
```

²⁵ <https://guide.meteor.com/structure.html> (kontsulta: 2017-07-24).

});

Kasu honetan, “/jarduerak” izeneko bidea aurkitzen dugunean, automatikoki izen bera duen txantilo bat bilatuko du Meteor-ek eta bertara bideratuko gaitu. Bukatzeko, bide horri izen bat eman behar diogu, aplikazioak hala eskatzen duelako.

Bukatzeko, *public* karpetan aplikazioan erabiliko diren argazkiak eta abar biltegitratzen dira. Ez badira karpeta horretan gordetzen, aplikazioak ezingo ditu atzitu. Azkenik, “.meteor” karpetan Meteor-en aurredefinitutako paketeak instalatutakoan aplikazioan sartzten diren fitxategi guztiak gordetzen dira. Garatzaileak ez du karpeta horretan eskua sartzeko beharrik, konfigurazio guztiak eginak baitaude. Paketeren batean aldaketa bat egin nahi bada, hobe da zerbitzaritik egiteko aukerarik badagoen aztertzea.

6.1.2 Erabilitako paketeak

Web-aplikazioaren gauzatzerako, Meteor inguruneak eskaintzen dituen aurredefinitutako paketeetatik hainbat erabili dira. Hurrengo taulan laburbilduko dira erabilitakoak:

PAKETEAK	FUNTZIONALITATEA
<i>Accounts-password</i> ²⁶	Erabiltzaileen erregistroa, logina eta saiok kudeatzea.
<i>Alanning:roles</i> ²⁷	Aktore ezberdinentzako rola definitu eta kudeatzea.
<i>Iron:router</i> ²⁸	Orri edo txantiloien arteko nabigazioa definitu eta URLak kudeatzea.
<i>Cfs:standard-packages</i> ²⁹	Argazki eta dokumentuak bezalako fitxategien biltegitratze eta igoera kudeatzea.

4. taula: Erabilitako aurredefinitutako paketeak.

6.2 ARAZOAK ETA SOLUZIOAK

Web-aplikazioaren garapenean zehar hainbat arazo sortu dira, planifikatutako inplementazioan aurreikusi ez zirenak eta modu batera edo bestera aldatetarik sortu dituztenak. Jarraian, horietako hainbat deskribatuko dira, hauek konpontzeko erabilitako soluzioarekin batera (konpontzeko modua aurkitu bada).

- **Erabiltzaileen sorrera.** Lehen arazo esanguratsua garapena hastearekin batera agertu zen, erabiltzaileak sortzeko garaian, hain zuzen. Hasteko, Meteor-en *Accounts-password* paketeak bere erregistrorako funtzionalitate propioa erabiltzera behartzen du garatzailea, eta ez dago modurik beste nolabait erabiltzaile bat sortzeko. Beharrezkoa zen aplikazioan erabiltzaile bat erregistratua izatea behintzat, erabiltzaile-kudeaketa atala inplementatzeko; eta, hortaz, erregistro-orri bat

²⁶ <https://guide.meteor.com/accounts.html> (kontsulta: 2017-07-24).

²⁷ <https://github.com/alanning/meteor-roles> (kontsulta: 2017-07-24).

²⁸ <https://github.com/iron-meteor/iron-router> (kontsulta: 2017-07-24).

²⁹ <https://github.com/CollectionFS/Meteor-CollectionFS> (kontsulta: 2017-07-24).

implementatu behar izan zen, lehen erabiltzaile bat gordetzeko datu-basean (erabiltzaile honekin gehiago sortzeko ondoren). Behin erabiltzailea sortuta, erregistro-orria kendu egin zen.

- **Erabiltzaileen atributuen sorrera.** *Accounts-password* paketeak erabiltzaileak bi atributu izatera mugatzen ditu: emaila eta pasahitza. Erabiltzaileek atributu gehiago izatea nahi zenez (izena, erabiltzaile-izena, informazioa...), eskuz kodetu behar izan zen hau. Sarean bilatzen ordu dezente pasatu ondoren, zerbitzariaren aldean *Accounts* paketean moldaketak egiteko *onCreateUser()* izeneko funtzio bat erabil zitekeela ikusi zen. Funtzio hori erabiltzaile (user) berri bat sortzen den bakoitzean exekutatzeko da eta bertatik zehatz daitezke atributu gehigarriak, *profile* izeneko atributuan azpiatributuak sortuz.
- **Routing dinamikoa.** Badira web-aplikazioan modu dinamiko batean esleitu behar diren ibilbide edo estekak, foro edota karpeta zehatzenak esaterako. Hauetan, egitura orokor bat jarraitzen duen txantiloia bat kargatu behar da, baina erakusten den informazioak elementu ezberdin bakoitzari dagokiona izan behar du: foro edo karpeta zehatz bakoitzarena, alegia. Hori lortzeko arazo dezente topatu ziren, txantiloiak ez baitzituen atributuak antzematen zuzenean, eta ez baitzen *iron:router* paketearen dokumentazioan honi buruzko informaziorik aurkitzen. Azkenean, foro batean topatu zen posible zela *data* izeneko ezaugarri berezi bat erabiltzea *route()* funtzioaren barruan, honen bidez elementu bakoitzaren informazioa erabili ahal izateko.
- **Angular-en txantiloiak.** Interfazeak garatzeko Angular inguruneak eskaintzen dituen aurredefinitutako txantiloia eta aukerak erabiltzeko saiakera egin zen, baina ikusi da Angular-en erabilera ez dela bateragarria Meteor-en zenbait paketerekin. Angular instalatu eta bere berezko etiketak erabiltzen hastean, aplikazioak errore-mezu bat adierazi zuen, Angular-ek *iron:router* paketearekin talka egiten zuela adieraziz. Horren ordez, Angular-en routing-erako pakete berezia erabili behar zela adierazten zuen. Arazo horri soluzioa bilatzen saiatu ondoren, aukera bakarra aplikazio osoa Angular paketeekin berreraikitzea zela ikusi zen eta, aukera hori bideraezina izanik aplikazioa nahiko aurreratua zegoelako, Angular ingurunea erabiltzea baztertu zen.

6.3 KODE ZATI ESANGURATSUAK

TXANTILOI BATEN EGITURA

```
<template name="foroa">
  <h2>{{izenburua}} foroa</h2>
  <div>
    <h4><a class="grixa" href="{{pathFor 'foroak'}}">Atzera (Foroak)</a></
h4>
  </div>
  <form>
    Gehitu eztabaida berri bat:
    <input type="text" placeholder="Eztabaidaren izenburua"
name="eztabaida">
    <input type="submit" value="Eztabaida Gehitu">
  </form>
  <br/>
  <p><strong>Foro honetako eztabaidak:</strong></p>
  <div>
    {{#each eztabaidak}}
      <div class="zerrenda3">
        {{#if isInRole 'admin'}}
          <button alt="Ezabatu" class="deleteE">&times;</
button>
        {{/if}}
        <strong><a href="{{ pathFor
'eztabaida'}}">{{izenburua}}</a></strong>
```

```

        </div>
      {{/each}}
    </div>
  </template>

```

Ikusten den moduan, txantilo bat definitzeko `<template>` etiketa erabiltzen da eta bere barruan kokatzen da egitura guztia `body` etiketa erabiltzen denean bezalaxe. Ikus dezakegu *spacebar* delakoen erabilera, zerbitzaritik jasotako datuan bistartzeko, eta `#each` klausularen bitartez nola osa daitekeen `for` moduko bat, jasotako elementu bakoitzarekin begizta moduan jokatzeko. *Spacebar*-ek *Handlebar* moduan jokatzeko dute eta giltza bikoitzekin adierazten dira: `{{ }}`.

Spacebar-etan erabiltzen den beste klausula ohiko bat `#if` da. Kodean ikus daitekeen moduan, erabiltzailearen rola identifikatzeko erabiltzen da, txantilo honetan. `isInRole` hitz-gakoaren erabilera posible da *Roles* paketeak eskaintzen duen funtzionalitate bat delako.

TXANTILOI BATEN LOGIKAREN EGITURA (Javascript)

```

import {Meteor} from 'meteor-meteor';
import {Template} from 'meteor-templating';
import {Accounts} from 'meteor-accounts-base';
import {Session} from 'meteor-session';
import {ReactiveDict} from 'meteor-reactive-dict';

Template.foroa.onCreated(function() {
  this.state = new ReactiveDict();
  Meteor.subscribe('eztabaidak');
});

Template.foroa.helpers({
  eztabaidak() {
    return Eztabaidak.find({foroId: this._id}, {sort: {sortzeData:-1}});
  },
});

Template.foroa.events({
  'submit form': function(event) {
    event.preventDefault();
    var izenb = event.target.eztabaida.value;
    var user = Meteor.user();
    var foroId = this._id;
    if (izenb!=""){
      Meteor.call('sortuEztabaida',izenb, user, foroId,
function(err, fileObj){
      if (!err){
        window.alert("Eztabaida berria gehitu da.");
        event.target.eztabaida.value = '';
      }else{
        console.log("something goes wrong with the
following error message " +err.reason )
      }
    })
  }else{
    window.alert("Izenburu bat sartu behar duzu.");
  }
},
'click .deleteE'() {
  if( confirm('Ziur eztabaida hau ezabatu nahi duzula? Erabiltzaile guztiak
sortutako iruzkinak ezabatuko dira.')){
    Meteor.call('ezabatuEztabaida', this._id);
  }
});

```

Hemen Javascript fitxategi baten ohiko egitura azter daiteke. Lehenik eta behin, Javascript orrietan beharrezko `import`-ak txertatu behar dira, batez ere aurredefinitutako pakete bateko funtzioak erabili nahi badira. Jarraian, ikus daiteke nola dei egiten zaion nahi den txantiloari, osatu nahi den bere logikaren atalaren arabera.

`OnCreated` atalean, hasierako konfigurazioak egiten dira. Kasu honetan, garrantzia duen lerroa `Meteor.subscribe()` funtzioa erabiltzen duena da, zeinak zerbitzaritik helarazten diren argitalpenak (datu-baseko informazioa) jasotzeko balio duen.

`Helpers` atalean, HTML orrian `spacebar`-en bitartez atzitu ahal izango diren aldagai laguntzaile moduko batzuk sortzen dira. Laguntzaile hauek aldagai boolearrak, elementuak edo elementu-array-ak gordetzen dituzte normalean. Kasu honetan, eztabaidak izeneko laguntzailea sortu da, foro horretako eztabaida guztien zerrenda biltegitzeko. Helper hau, aurreko atalean aipatu den `#each` klausularekin erabiltzen da, begizta osatzeko.

Bukatzeko, atal garrantzitsuena aztertuko dugu: `events` atala. Hemen aplikatzen da gertaeretan oinarritutako programazioaren funtsa, erabiltzaileak egindako `input`-ei erantzuteko logika inplementatzen baita bertan. Ikus dezakegunez, bi `event` daude definituta: `submit form` eta `click .delete`. Lehen, formulario bateko `Submit` botoia sakatzean jartzen da martxan eta beharrezko deiak egiten ditu, eztabaida berri bat sortu eta datu-basean biltegitzeko. Bigarrenak, erabiltzaileari baieztapena eskatu ondoren, eztabaida bat ezabatzeko metodoari deitzen dio.

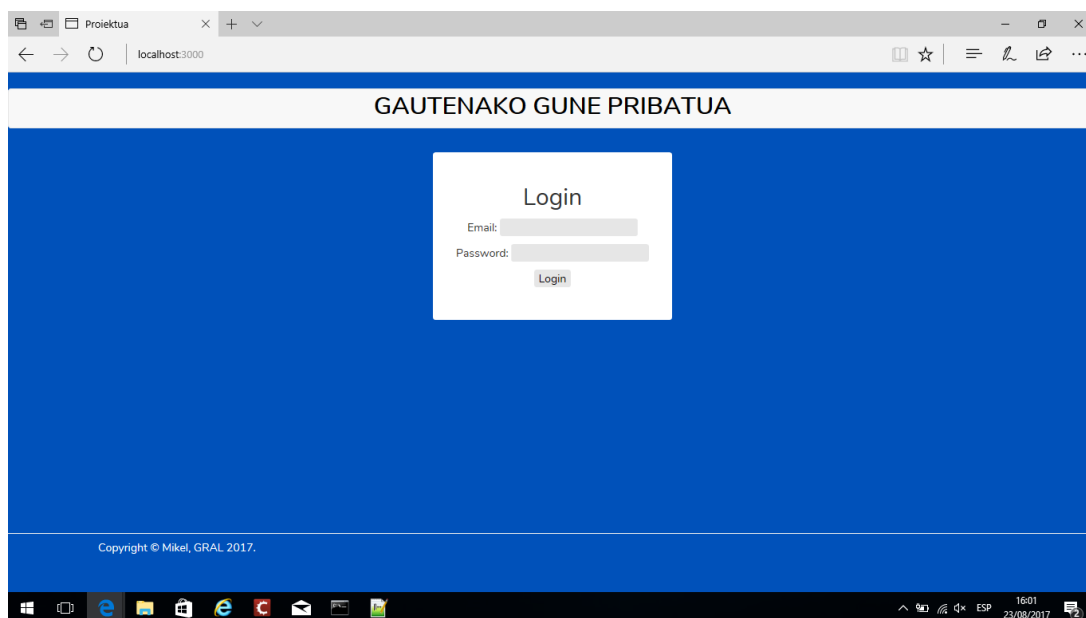
6.4 INTERFAZEAK

Gune pribatuko interfazeak garatzeko garaian, lehenik eta behin kontuan izan da web-aplikazio hau ez dela web isolatu bat, web-orri nagusi baten osagarri bat baizik. Hortaz, web-aplikazio honek izan beharreko interfazeak web nagusiaren parekoak (edo antzekoak, behintzat) izan behar du, erabiltzaileek batetik bestera pasatzerakoan aldaketa arrotzik nabari ez dezaten.

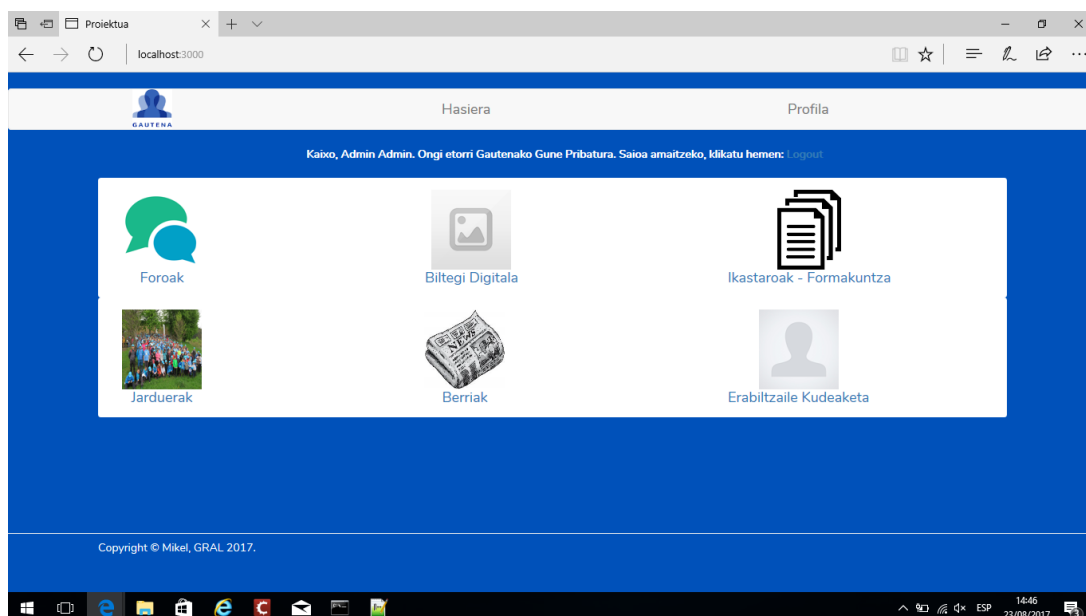
Asmo horri jarraiki, Gautena erakundeak bere web-orri berrituan (www.gautena.org) erabiltzen duen hizki-mota eta kolore korporatibo berak erabili dira aplikazioko orri guztietan; *Nunito* letra-mota testuentzako eta zuria, beltza eta urdina (`#0051ba`), batez ere, kolore bezala.

Interfazeak garatzeko, HTML eta CSS soilak erabili dira. Alde honetatik, Bootstrap, Angular edo bestelako aurredefinitutako txantiloiren bat erabiltzea baztertu da, alde batetik, aurreko atal batean aipatu den Angular eta Meteor paketeen arteko bateraezintasunagatik eta, bestetik, aplikazio nahiko estatiko eta sinplea denez, CSSa eskuz kodetzea errazagotzat jo delako Bootstrap txantilo egokia aurkitu eta aplikazioan txertatzea baino. Bootstrap Meteor-en erabil daitekeen arren, garapen-ingurunearen *template* bidezko egiturara egokitu behar dira bertako txantiloiak. Bootstrap paketeak ematen dituen aukera sinple batzuk (zutabeak, esaterako) bai baliatu direla aurkezpena txukuntzeko, txantilo bat zuzenean erabili ez den arren.

HTML orriak horrela egituratzen dira: *mainLayout.html* izeneko orria da oinarritzko txantilo edo template-a eta bere barnean txertatzen dira gainerako guztiak. Txantilo honetan dago kokatua nabigazio-barra, eta erabiltzaile bat saioa hasita badago bertako edukia aldatzen da. Txantilo nagusia denez gero, bere edukia uneoro egongo da ikusgarri web-aplikazioan, eta jarraian datorrena izango da aldatzen joango dena. *mainLayout* txantiloaren barnean *start.html* orria kokatzen da, eta honek kudeatzen du saioa hasia dagoen edo ez, horren arabera login orria edo menu nagusia (*dashboard.html*) bistaratzeko. Hurrengo bi irudietan ikus daiteke paragrafo honetan azaldutakoa.



20. irudia: Login orria.



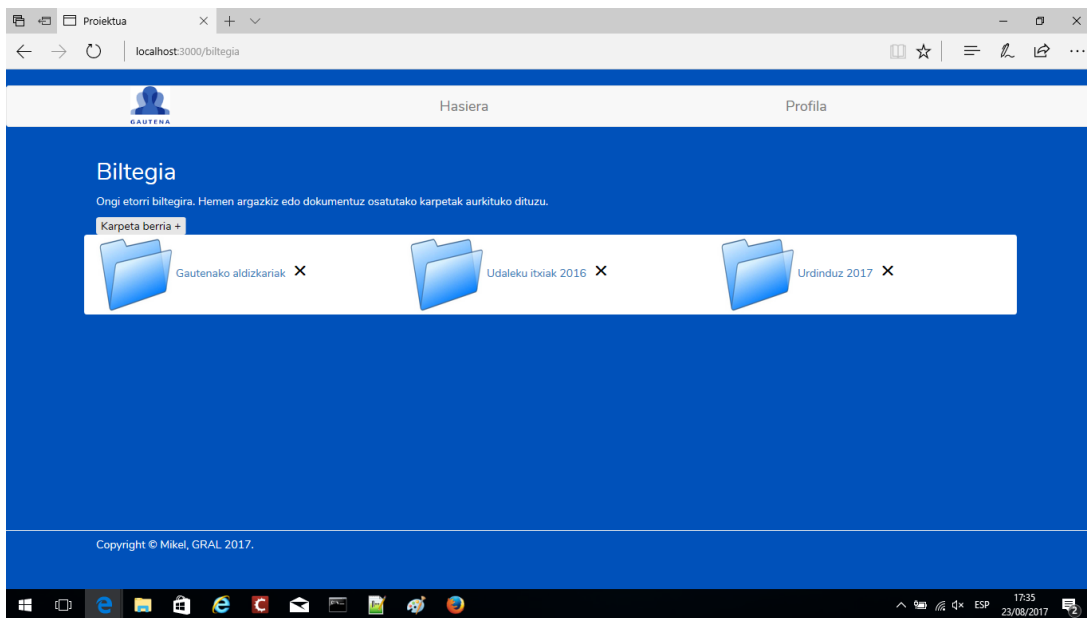
21. irudia: Administrazioaren menu nagusia.

Goiko irudian ikus daitekeen bezala, nabigazio barran (Gautenaren logotipoaz gain) hasiera-orrira itzultzeko eta profila ikusteko estekak kokatzen dira; jarraian, saioa amaitzeko esteka eta bere azpian menu nagusia, zeinak aplikazioaren atal bakoitzerako irudi bat

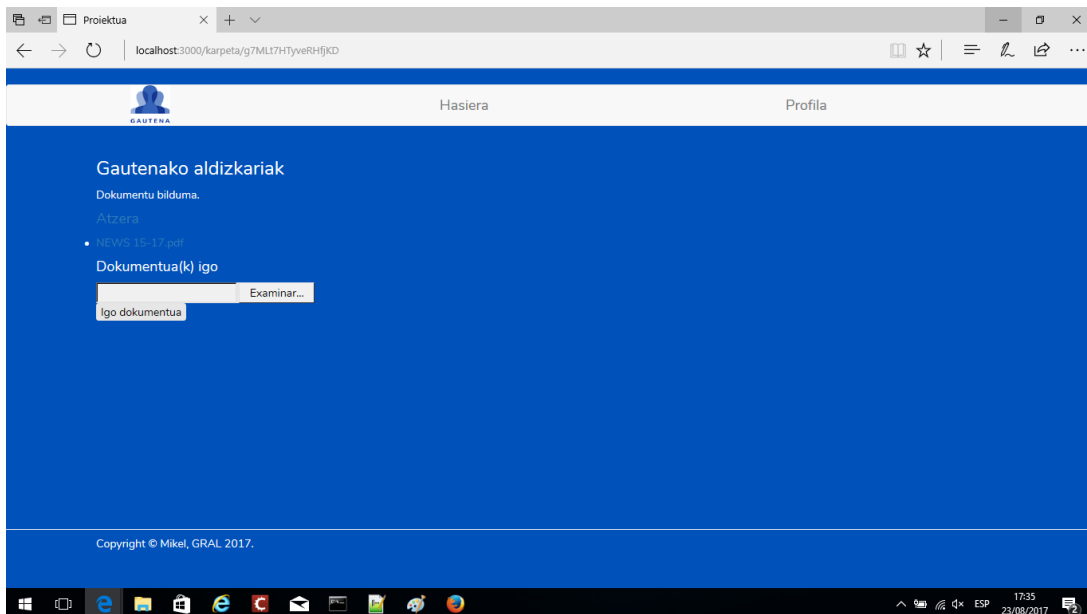
erakusten duen esteka moduan. *Erabiltzaile Kudeaketa* izeneko atala administratzaileentzat soilik dago ikusgai (beste rol bateko erabiltzaile batekin sartuz gero ez da pantailan agertuko) eta gainerako atalak beti bistaritzen dira.

Menu nagusitik aurrera, HTML txantilo guztiak *dashboard.html*-tik routing-aren bitartez atzitzen dira, hau da, bideak definituz *router.js* fitxategian. Egitura honen bitartez, eta Meteor-en Spacebar bereziei esker, itxura eta egitura finkoa duen txantilo bat defini dezakegu web-aplikazioaren frontend-aren oinarri bezala eta jarraian gainerako edukiak bistaratu konfigurazio-arazorik gertatzeko aukerarik gabe.

Jarraian aplikazioaren atal nagusiei dagozkien interfazeen pantaila-irudiak aurkeztuko dira. Irudi bakoitzean interfaze horri dagokion rola adierazten da, txantilo berean aldaketak egon daitezke-eta erabiltzaile motaren arabera.

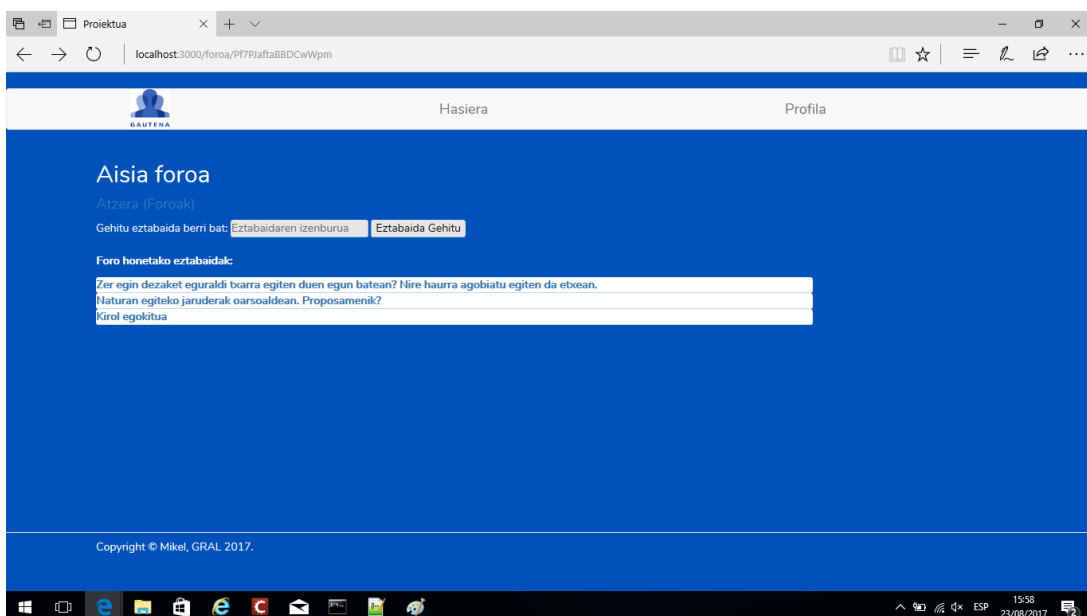


22. irudia: Biltegi digitala (administratzaile-rolean).

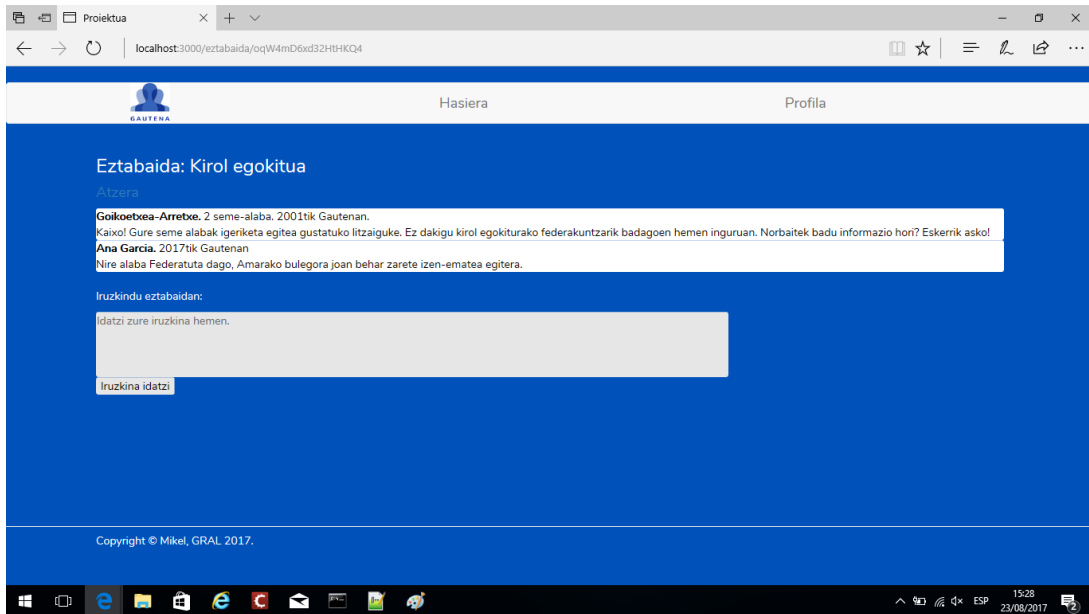


23. irudia: Dokumentu bilduma (administratzaile-rolean).

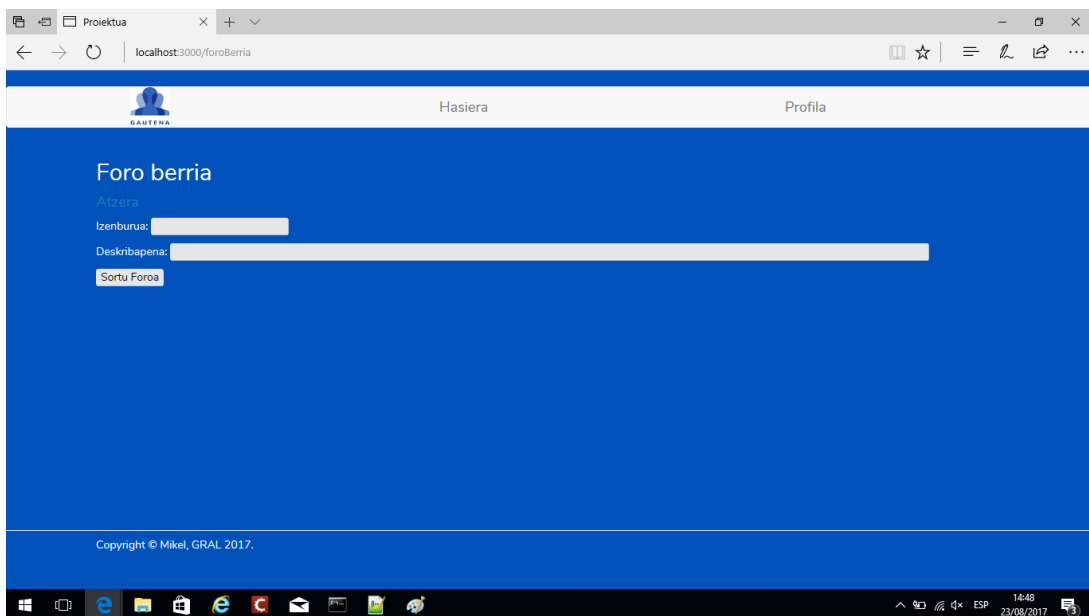
28. irudian ikus daitekeen bezala *Karpeta Berria+* aukeraz gain, karpeta bakoitzaren ondoan haiek ezabatzeko balio duten gurutzeak daude. Aukera haiek administratzaile-rolean dagoen erabiltzaile bati agertuko zaizkio soilik, aktore mota horrentzako eksklusiboak baitira erabilpen-kasu horiek. Berdina gertatzen da 29. irudian, non dokumentuak igotzeko aukera dagoen. Aplikazio osoaren interfazeak horrela daude egituratuta, eta modu honetan nabigazioa eta funtzionalitateen erabilpena intuitiboa da edozein roletzat; aldi berean kodetzea errazten du, ez baitira txantilo ezberdinak egin behar rol ezberdinetarako.



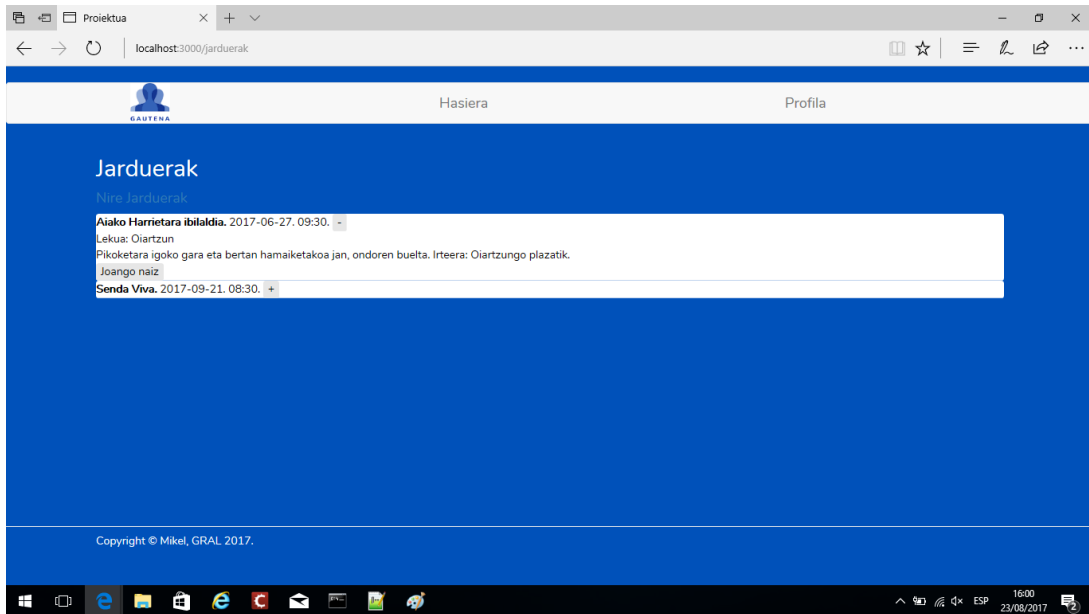
24. irudia: Foroa (erabiltzaile-rolean).



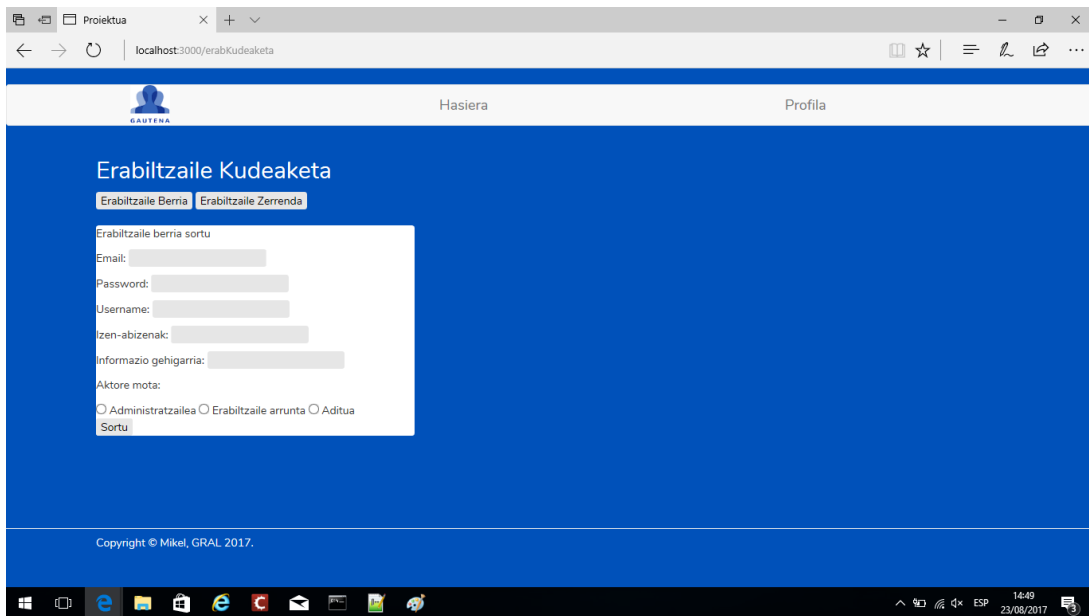
25. irudia: Eztabaida (erabiltzaile-rolean).



26. irudia: Foro berria sortzeko menua (administratzaile-rolean).



27. irudia: Jarduerak (erabiltzaile rolean).



28. irudia: Erabiltzaile-kudeaketa (administratzaile rolean).

7. JARRAIPEN ETA KONTROLA

Atal honetan, proiektuaren hasieran aurreikusitako denbora-estimazioak eta lan-plangintzak amaieran erabilitako denbora eta egindako lanarekin alderatuko dira.

7.1 LAN-KARGA ETA DESBIDERAKETAK

Proiektuaren hasieran, lana pakete ezberdinetan banatu eta pakete bakoitzak izango zuen karga estimatu zen, bakoitzean erabili beharreko denbora aurreikusteko. Normalean gertatzen den bezala, aurreikusitakoaren eta amaieran erabili den denboraren artean aldaketak egon dira proiektuan zehar. Jarraian erakutsiko den taulan adierazten dira desbideraketa horiek.

LAN-PAKETEAK	ESTIMAZIOA (ordutan)	DEDIKATUA (ordutan)	Desbideraketa (%)
KUDEAKETA	50	18,5	-63
JARRAIPEN ETA KONTROLA	35	7,5	-78,6
PLANGINTZA	5	4,5	-10
BILERAK	10	6,5	-35
PROIEKTUA	250	265,75	6,3
PRODUKTUAK	210	186,5	-11,2
AZTERKETA	42	26,25	-37,5
INFORMAZIO-BILKETA	12	11,5	-4,2
KONPARAKETA	30	14,75	-50,8
WEB-APLIKAZIOA	168	160,25	-4,6
ESKAKIZUNEN BILKETA	10	7,25	-27,5
DISEINUA	38	34	-10,5
DISEINUA: ERABILPEN-KASUAK	12	5,25	-56,3
KLASE-DIAGRAMAK	13	6,5	-50
INTERFAZEAK	5	5,25	5
SEKUENTZIA-DIAGRAMAK	8	17	112,5
GARAPENA	120	119	-0,8
TEKNOLOGIAREN AZTERKETA	15	11	-26,7
GARAPENA: ERABILPEN-KASUAK	83	91,75	10,5
FRONTEND-A	22	16,25	-26,1
MEMORIA	32	87,25	172,7
DEFENTSA	8	8	0
GUZTIRA	300	300,25	0,08

5. taula: Lan-paketeen denbora estimatuaren eta dedikatuaren arteko alderaketa.

Taulan ikusten den modura, GAPeen dedikatutako denborara ia estimatutakoaren bera da (% 0,08ko desbideraketa, soilik). Hala ere, orokorrean desbideraketa esanguratsuak izan dira hainbat atazatan, positiboak batzuetan eta negatiboak beste batzuetan. Guztiak banan-banan aipatu gabe, azpimarratu behar da *memoria* lan-paketeen izandako desbideraketa izugarria (% 172,7), non ataza honi eskainitako denborak estimatutakoa ia hirukoiztu duen. Desbideraketa honetatik ondorioztatzen da ataza honen estimazioa ez zela batere zuzena izan eta kontuan izan behar zela proiektuak atal teoriko garrantzitsua hartzen zuela bere barnean, memoriaren erredakzioa luzatuko zuena. Aipatutako datu hori hobeto ulertzeko *konparaketa* atazaren desbideraketa behatzea komeni da, izan ere, frameworken arteko konparaketa soila egiteak (ezaugarriak ikusi eta alderatu) ez du aparteko lan-kargarik

eskatu (-% 50,8, estimatutakoaren erdia), baina konparaketa hau janztea eta dokumentu batean txukun erredaktatzea izan da denbora-kostu handia eskatu duen jarduera. Konparaketaz gain, amaierako hausnarketa eta ondorioek ere dedikazio dezente eskatu dute.

Memoriaren lan-paketeak izandako desbideraketa positiboak beste ataza batzuetan izandako desbideraketa negatiboak konpentsatu ditu, hala nola *Jarraipen eta kontrola, Diseinua: erabilpen-kasuak* edo *Klase-diagramak* atazenak. Diseinuaren parte diren bi horietan, estimazioak luze geratu dira eta azkenean denbora gutxiago behar izan da lan horiek burutzeko. Jarraipen eta kontrolaren kasuan, honetan ere desbideraketa oso handia izanik (-% 63), jarraipena egiteko testu-dokumentu bat eta Excel taula bat erabiltzeak izugarri erraztu du lana, aurrerapenak oso modu azkarrean erregistratu ahal izan baitira. Kasu honetan, beraz, antolakuntza onak denbora aurrezten lagundu du.

Aipatu behar dira ere estimatutakoa baina denbora gehiago behar izan duten beste ataza batzuk: *Sekuentzia-diagramak* (% 112,5eko desbideraketa) eta *Garapena: erabilpen-kasuak* (% 10,5), hain zuzen. Lehenak, ikusten den bezala, denbora bikoitza baino gehiago behar izan du amaitzeko, aurrez beste atal batean aipatu den arrazoiarengatik izan du horrelako desbideraketa, hasieran, garapen-teknologia oso ezaguna ez zenean, planifikazio desegokia egin zelako. Horren ondorioz, aldaketa asko egin behar izan dira diagrametan eta denbora gehiago erabili da. Garapenean, erabilpen-kasuak inplementatzeak ere denbora gehiago eskatu du aurreikusitakoa baino, espero zitekeen desbideraketa bat izanik.

Azken esaldi horren harira, garrantzitsua da aipatzea ez direla inplementatu diseinuan proposatu ziren erabilpen-kasu guztiak, hain zuzen, desbideraketa izugarria eragingo zutelako eta baita atzerapen handia ere proiektuaren entregan. Ikus daitekeen bezala, erabaki egokia izan zen, *Ikastaroak eta formakuntza* ataleko 15 erabilpen-kasuak (eta besteren bat) inplementatu izan balira ezingo baitzen proiektua iraileko deialdian entregatu seguruenik, eta horiek guztiak inplementatu gabe utzita ere, proiekturako estimatuta zegoen denbora oso antzekoan amaitu delako lana.

Azken ohar bezala, txosten honen entrega-datan oraindik ez da defentsa egiteko prestaketarik abiatu; beraz, ataza horri dagokion lan-paketearen dedikatutako denboratzat estimatutako bera hartu da, desbideraketa osoa kalkulatu ahal izateko.

Hausnarketa modura eta gertatutako desbideraketa aintzat hartuz, esan daiteke proiektuaren hasieran egin ziren estimazioak ez direla zuzenak izan, ataza batzuetan askogatik denbora gutxiago eta beste batzuetan askoz gehiago erabili delako azkenean. Zortez, desbideraketa batzuek besteak konpentsatu dituzte eta orokorrean proiektuak ez du denbora-desberdintasun handirik izan, baina arazo handiak egoteko arriskua egon dela ezin da ukatu.

7.2 ALDAKETAK

Denbora-estimazioekin gertatu den bezala, proiektuan egon dira aurreikusiak egon arren garatu ezin izan diren atalak. Kasu honetan, proiektuaren alde praktikoak bakarrik jasan ditu aldaketa horiek, alde teorikoan definitutako irismena guztiz bete den bitartean. Web-aplikazioaren garapenean izandako aldaketak azalduko dira jarraian:

- *Ikastaroak eta Formakuntza* ataleko erabilpen-kasuen inplementazioa bertan behera uztea erabaki zen, multzo handia osatzeaz gain, konplexutasun

handieneko erabilpen-kasuak zirelako. Gainerako ataletako erabilpen-kasu gehienak garatuta zeuden mementoan jada erabilpen-kasuentzat esleitutako denbora gairatuta zegoen, eta memoriaren zati handi bat osatzeko; beraz, memorian zentratu eta erabilpen-kasu hauek baztertzea erabaki zen. Inplementatu ez izanak ez du eraginik izan teknologia-azterketaren frogapenean, modu berean aplikatu ahal izan baita alde teorikoa praktikan. Bezeroaren aldetik ere zatiren bat inplementatu gabe uzteko aukera eman zen, gainera.

- Profil-argazkia aldatzeko erabilpen-kasua inplementatzea ez da lortu. Ez da lortu argazkien miniaturak bistaratzea, eta hori dela eta erabilpen-kasu hori garatu gabe utzi da. Arazo berdina sortu da biltegi digitalean, non argazkiak atzitu daitezkeen baina ez den posible hauek irudi txikiagoz bistaratzea aurretik. Arazo honek zuzenean eragin du web-aplikazioaren itxuran eta kalitatean.

8. ONDORIOAK

Atal honetan proiektua gauzatu ondoren ateratako ondorioak eta web-aplikazioari begira dauden etorkizuneko aukerak azalduko dira.

8.1 HELBURU OROKORREN KALITATE MAILA

Azpiatal honetan proiektua burutu ondoren lortu den kalitatea aztertuko da. Proiektu honen helburu nagusia teknologia-azterketa bat egitea izan arren, web-aplikazioaren kalitate mailan ere erreparatuko da, bigarren mailako helburua izanagatik ez baitu esan nahi haren kalitateak garrantzirik ez duenik.

8.1.1 Teknologia-azterketa

Teknologia-azterketaren helburu nagusia web-aplikazioak garatzeko gertaeretan oinarritutako zerbitzariak dakartzaten hobekuntzak edo aukerak, hobeto esanda, ezagutzea eta aztertzea izan da. Horretarako, lehenik gertaeretan oinarritutako programazioaren paradigma aztertu da, teknologiak duen funts teorikoa ezagutzeko. Alde horretatik egindako analisia egokia izan da, eta ondoren kontzeptu eta teknologia zehatzagoetara pasatzeko oinarriak finkatu ditu.

Teknologia zehatzagoetara igarota, aurrez ezagutzen zen NodeJS-z gain gertaeretan oinarritutako zerbitzariekin lan egiten duten beste hainbat aukera daudela ikusi da, beste lengoia batzuetan idatziak daudenak. Argi finkatu dira hari bakarraren erabilerak ohiko zerbitzarien funtzionamenduari dituen ezberdintasunak eta eskaintzen dituen abantailak. Bigarren pauso horretan, beraz, web-aplikazioetan paradigmak duen aplikazioa ezagutu eta ulertu da, modu egokian berriz ere.

Azken atalean, irizpide eta ezaugarri ezberdinen artean garrantzitsuenak identifikatu dira, hainbat eta hainbat aztertu ondoren. 7.2 azpiatalean honi buruzko xehetasun gehiago emango diren arren, aipatu behar da irizpide horien (eta irizpide horiek betetzen zituen garapen-ingurunearen) aukeraketak emaitza onak ekarri dituela bere aplikazio praktikoan. Laburbilduz, beraz, teknologia-azterketa oso gogobetekotzat jotzen da, hainbat kontzeptu ikasi eta alde praktikoa errazago abiatzeko balio izan baitu.

8.1.2 Web-aplikazioa

Web-aplikazioan lortutako kalitate mailari buruz hitz egiterakoan, erabilpen-kasuen kopurua azpimarratu behar da lehenik, 45 izan baitira guztira diseinuan proposatu direnak. Erabilpen-kasuen kopuru handi horretatik, azkenean, denbora kontuengatik eta proiektuko beste emangarriak osatzeko ere denbora behar zelako, 30 inguru inplementatu dira. Beraz, ez da hasieran nahi zen web-aplikazio osoa garatu, baina gure prototipoa dezente hurbiltzen da aurreikuspenetara, funtzionalitatearen aldetik.

Bezeroari eskatu zituen funtzionalitate guztiak barneratzen dituen web-aplikazioa ezin izan zaio entregatu, baina malgutasuna izateko aukera eman zuenez gero, ez da arazo handia izan berarentzat. Etorkizunean garatzen jarraitzeko aukera ematen du garatutako web-aplikazioak, oinarritzko egitura eta funtzionamendua egokia baititu.

Interfazei erreparatuz, Gautenaren kolore korporatiboak eta web-orriaren estiloak (menuak, letra mota...) errespetatzeko ahalegina egin da, gune pribatua web-orri nagusian

txertatu behar denez, itxura arrotzik ez izateko. Interfazeak oinarrizko modu batean garatu dira, CSS soila erabiliz. Lortutako emaitza ikusita, badago zer hobetu interfazeen aldetik, baina nahikotzat jotzen da prototipo baterako.

Amaitzeko, aipatu beharra dago proiektu honen garatzailearentzat garapen-ingurunea guztiz berria zela, eta horrek zenbait mementotan inplementazioa moteldu eta zaildu duela.

Laburbilduz, web-aplikazioaren kalitate maila nahikoa da eraiki nahi zen prototiporako. Hori baino garrantzitsuago, aztertutako teknologia praktikan aplikatzeko balio izan du, proiektuaren helburu nagusia betetzeko, alegia. Honi buruzko azalpen gehiago 7.2 atalean emango dira.

8.1.3 Ikasitako lezioak

Behin proiektua amaitua dagoela, hilabete hauetan ikasitako hainbat lezio zerrendatuko dira jarraian:

- **Kanpo-faktoreak ere arrisku bezala kontsideratu behar dira.** Arriskuen identifikazioa egin zen unean, hainbat arrisku-faktore azpimarratu ziren, hala nola, ordenagailua hondatzea, garapenean trabatzea... Hauek guztiak (bezeroari lotutako bat izan ezik) garatzailearen esku egon daitezkeen ezustekoak dira, hau da, kontrola daitezkeenak. Baina, proiektu honetan izan da espero ez zen kanpo-faktore bat, arriskueta kontsideratu ez zena: tutoreak izan ditzakeen arazoak. Proiektu honen kasu zehatzean, faktore honek tutore-aldaketa bat ekarri zuen eta ia astebeteko atzerapena proiektuaren garapenean, hainbat gauza berriro lotu ziren arte (tutore berria, hasierako lana berriro azaldu behar izatea...). Gerta zitezkeen ere tutore-aldaketarekin irizpide-aldaketa handi bat etortzea eta lana berregin behar izatea, baina kasu honetan ez da horrela izan. Beraz, hainbat kanpo-faktorek eragin dezakete proiektuan modu zuzenean, tutore-aldaketa batek eragin duen moduan: familia-ezbeharrak, ezusteko konpromisoak, sare-konexioa galtzea... Gertaera hauek ez daude proiektuaren egilearen esku eta ezer gutxi egin daiteke prebentzio-neurri bezala, baina kontsideratu behar dira behintzat, gertatzeko aukera egon baitago.
- **Ezinbestekoa da plangintza on bat egitea.** Gradu Amaierako Proiektua lan profesional bat bezain handia ez den arren, denbora eta dedikazio asko eskatzen du, graduan egindako beste edozein lanek baino gehiago. Horregatik, ezinbestekoa izan da ondo planifikatzea eta hasieratik egin beharreko lana ondo finkatzea. Kasu honetan, plangintzak asko lagundu du orokorrean eta ondo bete da, baina zehaztasun gutxiagoz definitu zen atalen batean asko nabaritu da gero plangintza horren falta. Esaterako, sekuentzia-diagramak behin baina gehiagotan berregin behar izan dira, hasieran aplikazioa nola egituratuko zen ez baitzen guztiz argi planifikatu.
- **Garrantzitsua da garapena inoiz ez etetea.** Web-aplikazioaren inplementazioan, normala den bezala, arazoak sortzen dira eta atalen batean aurreratu ezinik geratzea nahiko ohikoa da. Horrelako kasuetan, eta berdin aplika daiteke proiektu

honen helburu nagusira (teorikora), oso garrantzitsua da denbora kudeatzen jakitea eta beste lan bati ekitea, aurreko arazoa konpontzeko modua aurkitzen den bitartean. Horrelako arazoek hainbat etenaldi sortu zituzten web-aplikazioaren hasierako garapen-fasean, eta beste modu batera kudeatuta denbora askoz gehiago aprobeztatzen dela ikusi da.

8.2 WEB-APLIKAZIOAREN GARAPENETIK ATERTAKO ONDORIOAK

Ondorioen barnean, azpiatal hau da garrantzitsuena, hemen azalduko baitira teknologia-azterketaren emaitzak praktikan aplikatu ondoren lortutakoa: zer izan den lagungarria, zer ez, zer desberdintasun egon diren beste garapen-ingurune batzuekin alderatuz, eta abar.

Lehenik eta behin, Javascriptek web-aplikazioan izan duen eragina izango dugu aztergai. Aurreko ataletan azalduakoa berriro hartuz, Javascript bezeroaren aldeko kodetzea egiteko erabiltzeaz gain, zerbitzarian ere erabili da NodeJS barneratzen duen Meteor frameworkari esker. Alde horretatik, oso erabilgarria izan da lengoia bakar baten erabilera, produktuaren garapena asko sinpletu baitu. Ez da beharrezkoa izan beste lengoia batekin trebatzea, eta, gainera, Javascript ezaguna izanik kodetzea nahiko modu automatizatuan egin da. Noski, egon dira ere Javascripten ezagunak ez ziren aukerak eta hauek ikasi behar izan dira, baina behin ikasita nahiko mekanikoa izan da kodetze-prozesu guztia. Beraz, bezero eta zerbitzarian Javascript erabiltzen duen garapen-ingurune bat erabiltzea onuragarria izan da garapenerako.

Meteor garapen-ingurunea aukeratzeko beste arrazoi garrantzitsu bat aurredefinitutako pakete eta txantiloak erabiltzeko aukera izatea zen. Ikuspegi honetatik, aldeko eta kontrako ezaugarriak ikusi dira. Aurredefinitutako paketeekin hasiz, esan beharra dago laguntza handia eskaintzen dutela eta erabiltzeko oso errazak direla. Komando-lerro bat exekutatzuz instalatzen dira eta erabiltzeko prest daude, erabiltzaileak instalazio-mementotik kodean pakete bakoitzeko hitz gakoak erabilgarri dituelarik. Pakete hauek hainbat funtzionalitate modu oso sinplean garatzeko balio izan dute, hala nola, login-sistema, routing-a, rolak eta beste hainbat. Txantiloiekin arazo gehiago izan dira. Angular-en erabilerak aurredefinitutako txantiloak aukera zabala eskaintzen du, baina aplikazio guztia bere egiturara egokitu beharra dago, hau da, aurredefinitutako pakete askok talka egiten dute Angular-ek erabiltzen dituenekin. Proiektu honetan, ez da posible izan Angular baliatzea routing-erako erabiltzen zen paketeak talka egiten zuelako, esaterako. Beraz, Angular erabili nahi bada hasieratik oso argi aztertu eta definitu behar da zein pakete erabiliko diren, bateraezintasunak ekiditeko, eta hori plataforman berria den garatzaile batentzat ez da erraza. Laburbilduz, aurredefinitutako paketeen erabilera oso positiboa izan da proiektuaren garapenean, baina txantiloak ere erabili ahal izateko oso ondo aukeratu behar da zein pakete erabili eta zein ez.

Beste ezaugarriei erreparatuz, denbora errealeko aplikazioa sortzeko aukera ematen zuen Meteor-ek eta aukera horrek oso ondo funtzionatzen du, oso erabilgarria izateaz gain. MVC patroia erabiltzeko aukerak ere garapena erraztu du, nahi bezala egituratu ahal baita aplikazioa, gainera. Hasieran nahiko nahasgarria suertatzen den arren egitura finko bat ez izatea, segituan osatzen du garatzaileak bere gustukoa den bat, eta horrela eroso egin dezake lan.

MongoDB datu-base ez-erlazionalaren erabilera, hasieran nahasgarria dirudien arren (klase- eta atributu-ezagatik), aplikazio sinpleak sortzeko oso egokia suertatzen da ez baita ia ezer definitu behar. Hala ere, kontuz ibili beharra dago bilduma eta “atributuen” artean zuzentasuna mantentzeko (ezaugarri errepikatuak, izen okerrak...) ez baitu inolako murriztapenik ezartzen.

Meteor-en berezkoak diren Spacebar-en erabilera ere aipatu nahi da atal honetan, asko sinplifikatzen baitu interfazeetan zerbitzariko datuak aurkezteko modua, aurreko atal batean azaldu den moduan.

Amaitzeko, aipatu beharra dago gertaeretan oinarritutako zerbitzarien erabileraren ikuspegitik, programatzeko era oso intuitiboa dela, gertaera jakinetarako bakarrik sortzen baita kodea. Javascript lengoia bakarria erabiliz asko sinplifikatzen da kodetu beharrekoa baina funtzionalitate aukerarik galdu gabe. Ez da harrizkoa teknologia honen hazkunde handia web-aplikazioen garapenean.

Laburpen modura, esan daiteke Meteor garapen-ingurunearen aukeraketa oso egokia izan dela eta asko erraztu duela web-aplikazioaren garapena, bere puntu txarrak ere badituen arren. Beraz, teknologia-azterketan egindako lana berresten du aplikazio praktikoak, gertaeretan oinarritutako zerbitzaria erabiltzen duen NodeJS teknologiaren eta hau barneratzen duen Meteor framework-aren onura eta abantailak frogatu ahal izan baitira.

8.3 ETORKIZUNeko AUKERAK

Web-aplikazioaren etorkizuneko aukerei begira, lehenik eta behin aipatu behar da ez dela zuzenean funtzionamenduan jartzeko moduko aplikaziorik lortu garapenean (ez zen hori helburua ere). Etorkizunean web-aplikazioari jarraipena emateko aukeren artean honakoak daude:

- “Ikastaroak eta formakuntza” atalaren inplementazioa gauzatzea, familiek on-line ikastaroak egin ahal izan ditzaten adituen laguntzarekin.
- Interfazeak web-orri nagusira egokitzea. Oraingoan lortu dena kolore korporatiboak eta tipografia mantentzea da, baina interfazeaz askoz gehiago sakontzeko aukera dago bi web-orri bereizi gabe orri bereko atal ezberdinen itxura izan dezaten bai orri nagusiak bai gune pribatuak.
- Argazkien miniaturak bistartzeko aukera gehitzea eta horrekin batera profil argazkik ikusi eta aldatzeko aukerak.
- Web-aplikazioko interfazeak *responsive* bihurtzea gailu ezberdinetara egokitzeko.
- Gune pribatua Gautenaren zerbitzarian kokatzea familiei zerbitzua eman ahal izateko. Memento honetan hori gauzatzea ez da posible erakundeak bere zerbitzariarekin dituen barne-arazo batzuk direla medio, bezeroak azaldu duenez.

Etorkizunean aurrera egiteko aukera izanez gero, hau izango litzateke puntu garrantzitsuena.

- Gune pribatuari funtzionalitate gehigarriak eta hobekuntzak egitea. Familiei zerbitzu hobea eskaintzeko beste hainbat funtzionalitate garatzea posible izango litzateke, sortzen diren beharren arabera.

Erreferentzia nagusiak

Gertaeretan oinarritutako programazioa:

<https://www.technologyuk.net/computing/software-development/event-driven-programming.shtml>

Gertaeretan oinarritutako programazioa Javascripten:

<https://www.24hr.se/event-driven-programming-in-javascript/>

NodeJS erabiltzen duten garapen-inguruneak:

<https://www.devsaran.com/blog/10-best-nodejs-frameworks-developers>

Meteor-i buruzko azalpen orokorrak:

<https://www.meteor.com/> eta <https://www.meteor.com/tutorials/blaze/creating-an-app>

Meteor-en trebatzeko azalpen eta adibideak:

<https://www.es.discovermeteor.com>

A Eranskina: Proiektuaren jarraipen-dokumentua

Eranskin honetan, proiektuaren jarraipen eta kontrola egiteko erabili den dokumentuan hartutako ohar guztiak aurkeztuko dira.

PLANGINTZA

- 2017-04-26 (3h 30min): Proiektua lan-paketeetan deskonposatu da eta denbora-estimazio bat egin da pakete bakoitzeko. Dokumentua osatu da lan-paketeen diagramarekin. Gantt diagrama osatu da lan-pakete bakoitzaren epeak adieraziz.
- 2017-05-08 (30min): Lan-paketeen denbora-estimazioak zuzendu dira eta Gantt diagramako paketeak kronologikoki ordenatu. Sekuentzia-diagramen paketea gehitu da.
- 2017-05-08 (45min): Tutorearekin bilera.
- 2017-05-09 (15min): Sekuentzia-diagramak Gantt diagraman gehitzea falta zen. Denbora kontrola eramateko taulan ere aldaketa hori egin da.
- 2017-05-30 (30min): Tutorearekin bilera.
- 2017-06-14 (1h): tutorearekin bilera.
- 2017-06-19 (15min): Tutorearekin bilera.
- 2017-06-23 (30min): Tutorearekin bilera.

WEB APLIKAZIOA

ESKAKIZUNEN BILKETA

- 2017-01-16 (2h, bilketa+azterketa): Bezeroarekin bildu eta eskakizunak jaso ondoren, hauek aztertu dira lehen ideia bat egiteko.
- 2017-01-17 (2h15min): Aurreko eguneko lana jarraituz lehen zirriborroak egin dira: erabilpen-kasuen lehen zerrenda egin, klase nagusiak irudikatu...
- 2017-02-08 (2h): Tutorearekin hitz egin ondoren, ideia batzuk gehitu zaizkio zirriborroari: foroaren egitura eta formakuntza atala. Oraindik ez dago dena argi eta dezente garatu behar da.
- 2017-05-09 (1h): Bezeroak eskakizun berriak jakinarazi ditu: *aditu* profil baten beharra, formakuntza-atalaren astekako egituraketa edo administrariaren partetik adituen esleipena.

DISEINUA

- 2017-04-27 (3h): Erabilpen-kasuen diagrama osatu da eta formatu digitalera pasaTU da. *Online ikastaroak* erabilpen-kasua ez da osatu zalantzak daudelako. Bezeroari galdetu behar zaio zein egitura izatea nahi duen erabilpen-kasu honek.
- 2017-04-27 (2h): Interfazeen paperezko diseinua egin da (guztira 15 interfaze diseinatu dira). Oraindik amaitzeko.
- 2017-04-28 (2h 30min): Interfazeen diseinua amaitu da *Online ikastaroak* atalekoak izan ezik. Bezeroak eskakizun zehatzagoak eman arte ez da atal honekin ezer aurreratuko.
- 2017-04-28 (1h 45min): Klaseen diseinua abiatu da, aktoreak eta foroa definituz. Zalantzak agertu dira diseinatzeko garaian eta garapen-teknologian gehiago sakontzea erabaki da ideia argiago bat edukitzeko.
- 2017-05-02 (2h 15min): Klase-diagrama osatu eta formatu digitalera pasatu da. Klaseei metodoak gehitzea falta da. Ikastaroen atala ez da osatu bezeroarengandik eskakizun zehatzagoak jaso behar direlako oraindik. Davidi bidaliko zaio dokumentua oniritzia eman dezan.

- 2017-05-05 (2h): Bezeroarekin hitz egin ondoren, ikastaroen atala gehitu da erabilpen-kasuen diagraman eta klase-diagraman. Aktore mota berri bat (aditua) gehitu da. Hainbat akats konpondu dira (loturak, agregazioa...).
- 2017-05-08 (1h 45min): Erabilpen-kasuen diagraman moldaketak egin dira dokumentua trinkotzeko. Bukatu gabe geratu da.
- 2017-05-09 (1h 45min): erabilpen-kasu eta klase-diagramak zuzendu dira. Aldaketa nagusiak, EK: diagrama trinkotua, 3 aktore nagusiak *aktorea*-ren espezializazio dira, formakuntza-ataleko erabilpen-kasuak gehituta. KD: iruzkin batek beste bat erantzun dezake, administratzaileak aktoreak sortu, ikastaroak, unitatea, autoebaluazioa, mezu pribatuak, klase berriek aktoreekin dituzten loturak gehitu dira eta oker zeudenak zuzendu, atributu gehiago gehitu dira.
- 2017-05-10 (2h): Sekuentzia-diagramekin hasteko, aurreko urteetako memorietan egindakoa aztertu da. Zalantzak agertu dira: erabilpen-kasu gehiegi izango ote diren memorian jartzeko, egitura orokorra edo zehatza adierazi behar den, eta kodetzen hasi aurretik beharko diren HTML eta JS fitxategi guztiak zehaztu beharko ote diren. Tutoreari email bat bidali zaio zalantza hauek argitzeko asmoz eta HAREN erantzunaren zain nago.
- 2017-05-10 (1h 30min): Tutorearen aholkuak jarraituz, erabilpen-kasuen diagramari `include` eta `extends` loturak gehitu zaizkio beharrezkoa zuten klaseen artean.
- 2017-05-11(2h): Klase-diagraman beharrezko metodoak gehitu dira. Aktore-klaseetan egitea soilik ikusi da beharrezkoa.
- 2017-05-12 (1h 45min): Interfazeen paperezko diseinua amaitu da. Ikastaroei dagozkien interfazeak gehitu dira, 3 aktoreentzako.
- 2017-06-01 (1h 45min): Klase-diagraman mementoz inplementazioan egin diren aldaketak eguneratu dira (metodoak eta aktoreen atributuak). Erabilpen-kasuen diagraman kasuak sistematan multzokatzeko ahalegina egin da baina ez da batera txukuna geratzen, beti gainjartzen baitira geziak. Mementoz, koloreekin bereizi dira multzoak eta tutorearekin eztabaidatuko da nola komeni den egitea.
- 2017-08-15 (8h): Zirriborroan eginda zeuden sekuentzia-diagramak *starUML* programarekin egin dira memorian sartzeko.

INPLEMENTAZIOA

- 2017-05-15 (5h): Aplikazioaren garapena hasi da, emaitzarik lortu gabe. Login-a egiteko aurredefinitutako paketea ez da nahikoa (atributu gehiago behar ditu, aktore mota desberdinak ez ditu kudeatzen). Atributuak gehitzeko moduak aurkitu dira baina ez erabiltzaile mota bat baina gehiagorentzat. Login-a eskuz programatu beharko da. Routing-a egiten saiatu da login-a egin ondoren beste orrialde batera pasatzeko baina ez da lortu funtzionatzea. `onClick` metodoa erabiliz egin da baina pantailan konfigurazio-arazo bat dagoela esaten du. Ez da sarean honentzat konponbiderik aurkitu mementoz. Konponbideak saiatzen Meteor-en egitura gehiago aztertu da eta zalantzak sortu dira benetan MVC patroia jarraitzen duen edo ez ez dagoelako argi. Fitxategi bakoitza non kokatu argiago geratu da behintzat.
- 2017-05-16 (5h): Ez da lortu oraindik login-a egitea, baina gaur teorikoki nola egin beharko zen ulertzen da. Hasierako formulario sinplea eta *klaseak* (bildumak, `mongodbn`) sortuta daude baina ez da lortzen funtzionatzea.

Iron router-ak errore-mezu bera erakusten du beti eta ez dago konpontzeko modurik mementoz. Meteor-en informazio oso gutxi dago login-a egiteko pakete “ofizialaz” aparte eta zaila da lan egitea. Agian lan egin ahal izateko paketea erabili beharko da modu ez oso egoki batean: *user* bera izatea aktore guztientzat baina batzuei atributu batzuk ematea eta beste batzuk besteei (itsusia, baina egin daiteke MongoDB-n). Tutorearekin komentatuko da osteguneko bileran. Hala ere, rol ezberdinak nola eman daitezkeen aztertu beharko da.

- 2017-05-18 (3h): Routing-aren funtzionamendua ondo ulertu da dagoeneko baina ez da lortzen behar bezala inplementatzea.
 - 2017-05-19 (3h): Login-aren inplementazioarekin jarraitu da. *Layout*-ak beste txantiloia dinamikoki bistaratzeko *yield* parametroa erabili da, baina *iron:router* berriro instalatu behar izan da funtzionatzeko. Ez da lortzen erabiltzaileak formulariotik erregistratzea, ezta kontsolatik erregistratu eta login-a egitea, nahiz eta tutorial honen pauso guztiak jarraitzen diren (<http://meteortips.com/second-meteor-tutorial/user-accounts/>). *Accounts* paketea berriro instalatu da baina ez da ezer aldatu.
 - 2017-05-20 (3h): Hainbat *import* gehitu dira horrela login-aren arazoa konpontzen zen ikusteko baina ez da ezer lortu. Proiektu berri bat sortu da Meteor-en berezko login-arekin lan egiteko. Login egiteko ez du balio kontsolatik erabiltzailea sartzeko, beste klase batzuekin ez bezala. Sistema hori hedatzeko eta erabiltzeko aukerak aztertu beharko dira.
 - 2017-05-22 (5h 30min): Erregistroa eta login-ak funtzionatzea lortu da. Arazoa Firefox nabigatzailearena zen (*websocket*-ekin arazoa ematen zuela zioen jakinarazpena ikusi da). Microsoft Edge-ra aldatuz ez du arazorik ematen. Tutorial hau jarraitu da oraingoan aurrekoa baino argiagoa delako: <https://www.sitepoint.com/creating-custom-login-registration-form-with-meteor->. Formulario bat gehitu da atributu guztiak dituzten erabiltzaileak sortzeko eta horretarako *onCreate* metodoa erabiltzen zuen JS funtzio sortu da, hurrengo estekako argibideak jarraituz. Rolak esleitzeko argibideak ditu ere (hurrengo pausoa): <https://medium.com/all-about-meteorjs/extending-meteor-users-300a6cb8e17f>
 - 2017-05-23 (6h 15min): Errore-kudeaketa sistema simple bat gehitu da login eta aktore sorrerarako. Rolak erabiltzeko paketea gehitu da eta inplementatu egin da atal hori. Orain 3 rol ezberdin daude eta rolaren arabera pantaila nagusiko edukia aldatzen da.
 - 2017-05-24 (4h 45min): Erabiltzaile-kudeaketa atala moldatu da dinamikoki *Erabiltzailea gehitu* edo *Zerrenda ikusi* aukeratzeko eta txantiloia bistaratzeko. *Session* atributua erabili da, iturriak:
 - <https://stackoverflow.com/questions-38251084-meteor-show-hide-element-on-button-click>
 - <https://docs.meteor.com/api-session.html>
- Erabiltzaile zerrenda ikusteko *template*-a gehitu da, eta baita aktoreak ezabatzeko aukera ere. *Simple-todos* tutorialerako pausoak jarraitu dira. Aktorea sortu eta Aktorea ezabatu erabilpen-kasuak amaituta daude. Erabiltzaile bakoitzak bere profila ikusteko beharrezko elementuak inplementatu dira (ez da lortu argazkia bistaratzeko). Argazkia eta pasahitza aldatzeko aukerak inplementatzea falta da. Hasiera orrira itzultzeko esteka gehitu da ere.
- 2017-05-25 (4h): Profilean, pasahitza aldatzeko aukera inplementatu da. Argazkiarekin lotutako erabilpen-kasuak Biltegi Digitalarekin batera egingo

direla erabaki da. Berrien atala sortu, datu basean txertaketak egin ditzake *admin* rola duen aktoreak baina ez da lortu berriak bistaratzea. Usersekin egin bezala jokatu da baina ez da arazoa aurkitzen.

- 2017-05-26 (2h): Berriak bistaratzea lortu da azkenean. Arazoa `import {Berriak} from minlib-berriak.jsmin;` lerroa kenduta desagertu da (nahiz eta aurkitu den dokumentazioan beharrezkoa zela jarri). Aurkezpen txukunagoa egitea falta da, interfazea lantzerakoan osatuko da. Hainbat orri egoteko aukera ere inplementatu beharko da, baita berri baten lehen lerroak ikusi eta klikatzerakoan informazio gehiago lortzeko aukera ere. Admin-ak berriak ezabatzeko aukera ere gehitu da. Jarduerak gehitzeko orria gehitu da eta datu basean jarduerak txertatu ditzake administrariak jada.
- 2017-05-29 (1h30min, frontend-a): bootstrap paketea gehitu da eta aplikazioari nahi den itxura ematen hasi da (nabigazio barra, lerro-zutabeak...). Ez da lortu argazkiak jartzea esteken tokian, ezta ikonoak txertatzea ere, eta oso argi ez zegoenez nola lan egin, beste memento baterako utzi da. Iturriak:
 - <http://lineadecodigo.com/html-imagen-con-enlace-en-html->
 - <http://getbootstrap.com/css->
 - <http://getbootstrap.com/components->
- 2017-05-29 (4h30min, inplementazioa): Jarduerak ikusteko orria inplementatu da. Hasieran jarduera bakoitzaren xehetasunak beste orrialde baten erakustea pentsatuta eta ia lortuta bazegoen ere (nahikoa denbora eman du ibilbideak nola konfiguratu asmatzeak), azkenean zerrendako orrian bertan erakustea logikoagoa eta erosoagoa dela pentsatu da, erabiltzailearen ikuspegitik, ez baitaude beste datu asko erakusteko. Arazoak egon dira jarduera konkretu baten xehetasunak soilik eta ez guztienak bistartzeko. Baina azkenean lortu da, xehetasunak ireki eta ixteko sistema hobetu behar da, bat irekita dagoenean beste bati bi aldiz klikatu behar baitzaio hau irekitzeko.
- 2017-05-30 (1h30min): Jardueran izen ematea egitea lortu da. Hobetzeko geratzen da erabiltzaile bati bi aldiz izena ematen ez uztea. Administratzaileak jarduera bakoitzeko izen-emandakoen zerrenda ikus dezake.
- 2017-05-32 (3h30min): Erabiltzaileak bere jarduerak ikusteko orria inplementatu da. Xehetasunak ikusterako garaian + eta - ikurren bidez bereizten da orain. Ez da lortu izen emate bikoiztuen arazoa konpontzea, datu basetik jarduerak ez ditu ondo hartzen eta `null` balioa itzultzen du. Datu basetik bertatik probak egin dira eta MongoDB zintzilik geratzen da kontsulta egiterakoan. Arazo honentzako konponbide bat bilatu beharko da.
- 2017-06-01 (30min): Jarduerak ezabatzeko aukera gehitu da eta hasieran sortu diren erroreak konpondu dira. Hainbat erabiltzaile sortu dira eta errorerik ez dagoela konprobatu da. Kontsultaren arazoa amaierarako uztea erabaki da.
- 2017-06-08 (2h): Administratzaileak foro berriak sortu ditzake eta hauek zerrendatu egiten dira. Hasieran aurkitu zen tutorialak ez du balio *Angular*-ek barneratzen dituen txantilo- eta ibilbide-kudeaketa sistemak talka egiten dutelako orain arte erabili direnekin, errorea ematen du bi modu aldi berean erabiltzean. Beste modu bat bilatu beharko da foro bakoitzeko bere orrira joateko.

- 2017-06-09 (2h 30min): Lortu da foro bakoitzak bere helbidea izatea baina ez da lortzen helbide horietan edukia bistaratzea. Irudiak gehitu dira hasiera-orrian eta moldaketak egin dira goiburukoan ikonoak sartu ahal izateko. Ez da lortu *logout* aukera nabigazio barran kokatzea eta funtzionatzea, ez du ezer egiten eta ez da ulertzen zergatik.
- 2017-06-12 (5h): Foroa alde batera utzita (emaitzarik ez delako lortzen), biltegi digitalarekin hasi da lanean. lortu da argazkiak modu egokian igotzea eta bistaratzea klikatzerakoan, baina ezin dira “txikian” ikusi. Dokumentuak igotzen dira baina ez modu egokian, gero ezin dira atzitu. Karpeta sistema egiteko modu bat bilatu da ezagutzen ziren iturrietan baina ez da baliozko ezer aurkitu. fitxategi mota zehatzak bakarrik igo daitezkeela zehaztu arren, edozein mota igotzen uzten du.
- 2017-06-13 (4h, bestelakoa): Proiektua ordenagailu batetik bestera pasatzeko ahalegina egin da emaitzarik lortu gabe. Hainbat modu ezberdinetara probatu arren, martxan jartzerakoan (asko itxaron ondoren), ulertzen ez diren erroreak ematen ditu.
- 2017-06-14 (3h45min): Administratzaileak karpetak sortzeko aukera gehitu da eta karpetak ikus daitezke biltegian. Orain karpeta barruko funtzionamendu egokia inplementatu behar da. Ez da lortzen karpeta barruan sartu eta bere ezaugarriak (galeria den edo ez) hartzea.
- 2017-06-15 (5h): Karpeta bakoitzaren barruan bere atributuak bistaratzea lortu da. Arazoa `this.params` ez erabiltzea zen, horrela hartzen baitira estekan pasatzen diren datuak (segitzen ari ziren azalpenetan ez zen agertzen). Orrialde honetako azalpenekin lortu da egitea, kode sinpleagoarekin gainera: <http://meteortips.com/second-meteor-tutorial-iron-router-part-2->. Galeria den edo ez desberdintzen da orain eta administratzaileari dagokion igoera botoia agertzen zaio. Aldaketa berdinak egin dira foroaren *routing*-arentzako. Foroaren atal handienaren inplementazioa burutzea lortu da. Atal honetako klase umeez identifikatzaile bat izango dute, klase gurasoa lortzeko balio duena eta klase gurasoek ez dute klase umeez osatutako arrayrik izango. Horrela inplementazioa sinple eta argiagoa da. Diseinu aldetik hainbat hobekuntza egin dira ere. Foroaren atala biribiltzea falta den arren, erabilpen-kasu garrantzitsuenak burutzen ditu aplikazioak (foroak, eztabaidak eta iruzkinak sortu+bistaratu eta administratzaileak ezabatu). Erabiltzaileek bere iruzkin eta eztabaida propioak ezabatzea falta da.
- 2017-06-16 (45min): Foroan iruzkin propioak ezabatzeko erabilpen-kasuarekin jardueretan bezala ez da lortzen datu basetik beharrezko datuak ondo atzitzea. MongoDB sintaxia erreparatu da baina ez da soluziorik aurkitu.
- 2017-06-19 (1h): argazkiak eta dokumentuak modu egokian igotzen dira eta atzitu daitezke. Ez da lortu oraindik argazkiak aurreikustea. Argazkiak dagokion karpetan azaltzea falta da, `beforeWrite` bezalako funtzioaren bat erabili beharko da. Hasiera-orria txukundu da.
- 2017-06-20 (5h): Dagoeneko fitxategi bakoitza bere karpetan agertzen da. Karpeta eta bertako edukia ezabatzeko aukera gehitu da. Ez da lortu fitxategi igoera anitzak inplementatzea.
- 2017-06-21 (1h 45min): Arazo hauek konpontzen saiatu da, emaitzarik gabe: interfazeak egokitu, foro eta biltegian orria birkargatzekoan datuak ezabatzea, jardueretan 2 aldiz ezin izena eman izatea.

- 2017-06-23 (4h 45min frontend-a): Gautenako kolore korporatiboak eman zaizkio webguneari eta edukiak modu txukunagoan erakusteko CSS aginduak idatzi dira. Mantso doa aurrera.
- 2017-06-24 (6h 30min, frontend-a): Interfazean hobekuntzak egin dira `div`-ak egokituz eta parekoak jarriz orri guztietan. *Relative* aukeraren bitartez dinamiko handitu eta txikitzen diren edukiak lortu dira. Berriak, *Erabiltzaile kudeaketa* eta *Jarduera* atalak amaituta daude interfazeen aldetik. Testuak formatua ez galtzea falta da. Foroan eta login-ean ere hobekuntzak egin dira. Hasierako menuan eta biltegian ez da lortu nahi ziren aldaketak egitea.
- 2017-08-23 (1h, frontend-a): Interfazeetan hainbat txukunketa egin dira, pantaila-irudiak egiteko (zutabeak ondo kokatu, testuak txukundu...).

GARAPEN TEKNOLOGIAREN AZTERKETA

- 2017-01-11 (1h): Web-aplikazioak garatzeko inguruneei buruzko informazio askorik ez da lortu, aipatzen diren gehienak bereziki mugikorrenako aplikazioak sortzen baitira.
- 2017-01-13 (2h): NodeJS erabiltzen duten hainbat framework aurkitu dira (<https://www.devsaran.com/blog-10-best-nodejs-frameworks-developers>). NodeJS erabiltzea aukera ona izango litzateke teknologia ezaguna delako. Zerrendatik interesgarriena Meteor frameworka dela dirudi, “lengoaia” ezagunak erabiltzen dituelako (HTML, JS, CSS..). Sare sozialak eraikitzeko erabilia izan dela dirudi aurreko urteetako memoriak begiratu. 2017-01-15 (1h30min): Meteor instalatzeko saiakera egin da baina errorea ematen du, ez da konpontzea lortu.
- 2017-04-30 (2h): Meteor berriro instalatzeko saiakera egin da. Instalatzen da baina ez da proiektua abiatzen, *starting app* azaltzen denean blokeatu egiten baita. Sarean soluzioak bilatuz arazoa erabiltzaile gehiagori gertatu zaiela ikusi da baina ez da konponbiderik aurkitu. Windows7 ordenagailutik Windows10-era pasatzea probatu da, eta Windows10-en badabil. Meteor garapen- ingurunearen xehetasunak eta funtzionamendua hobeto ezagutzeko, web ofizialetik ateratako tutorial bat jarraitu da. 5-12 atalera arte osatu da. (<https://www.meteor.com/tutorials-blaze-creating-an-app>)
- 2017-05-01 (2h): Tutorialaren gainerako zatia amaitu da. Zalantzak geratu dira bezeroan eta zerbitzarian jarri behar diren dokumentuen aldetik baina teknologia eta “lengoaia” aldetik ikusitako guztia argi dago.
- 2017-05-02 (1h): Meteor garapen-ingurunearen egitura hobeto ulertzeko, gida ofizialetik dagokion atala aztertu da (<https://guide.meteor.com/structure.html>).
- 2017-05-09 (1h): Meteor-en aurredefinitutako paketeen artean (puntu indartsuetako bat) Foroa eta Biltegitratze sistema garatzeko lagun dezaketen batzuk bilatu dira. Foro bat inplementatzeko hainbat tutorial (dagokion kodeaz eta githubeko estekaz lagunduak) aurkitu dira eta biltegitratze sistemarako Meteor-en hainbat pakete batuta lortutako soluzio bat baino gehiago ere aurkitu dira. Informazio guztia gorde da *meteor forum+storage* dokumentuan.
- 2017-05-11 (30min): Sekuentzia-diagramak osatze aldera, Meteor-en arkitektura erreparatu da. Ikusi da errazagoa izango dela arkitektura sakontasunean ulertzea eta klase-diagrama zehatzak egitea garapenean trebatzen joaterakoan, beraz, klase diagramen zirriborro bat egingo da

hasiera batean eta, garapenak irauten duen bitartean osatuko dira sekuentzia-diagramak.

AZTERKETA

INFORMAZIO BILKETA

- 2017-02-4 (3h): Bezeroan eta zerbitzarian JS erabiltzen duten plataformak (isomorfitxoak) bilatu dira. Badirudi NodeJS dela aukera bakarra horretarako.
- 2017-02-5 (3h): NodeJS-ren pareko beste aukera batzuk aurkitu dira. Konparaketan, lehenik Event Driven web server kontzeptua azaltzea izan daiteke interesgarria, ondoren teknologia isomorfitxoak konparatzeko (NodeJS, JXCore...). Amaitzeko, NodeJS erabiltzen duten garapen-inguruneak konpara daitezke.
- 2017-02-6 (1h): Konparatuko diren teknologia eta inguruneak zehaztu dira: teknologiak: NodeJS, Twisted (Python, lengoia ezberdina izateagatik), Vert.x (lengoia aniztatan erabili ahal izateagatik); Frameworkak: Meteor, Mean, Mojito, Socket.io, Total.
- 2017-05-10 (2h 30min): Meteor eta Mean konparatzen dituen informazioa aurkitu da. Meteor-en alde onak nabarmentzeko hainbat ezaugarri ere. Mojitorentzako informazio gehigarria aurkitu da.
- 2017-06-03 (2h): Twisted eta Mojitoren informazioa hobetzeko xehetasun gehigarriak aurkitu dira

KONPARAKETA

- 2017-05-10 (2h 15min): *konparaketa: ideia nagusiak* fitxategian frameworken arteko konparaketa idazten hasi da. Atal hauek osatu dira: sarrera (osatzeko), Meteor, Mean eta Mojito.
- 2017-05-10 (1h 30min): aurrez aipatutako dokumentuaren lehen bertsioa amaitu da.
- 2017-06-10 (4h): Azken bileretan tutorearekin hitz egindakoa aintzat hartuz, teknologia-azterketaren atala izango denaren lehen bertsioa idazten hasi da. *Event-driven* eta teknologia atalak amaitu dira. Teknologiaren aldetik ez da informazio gehiegi aurkitu eta behar bada motzegia geratu da. Frameworken atala ere idatzi da baina hainbat ezaugarri gehiago jartzea (eta informazioa bilatzea) falta da. Taula ere erdi osatu da. *AZTERKETA-v1* dokumentuan idatziko da, ondoren Memoria-dokumentura pasatzeko.
- 2017-06-11 (3h): Frameworken atala amaitu da eta hainbat zuzenketa egin dira atzo idatzitakoan.
- 2017-06-16 (2h): Tutoreak adierazitako hainbat zuzenketa egin dira: paragrafoak osatu, izenburua aldatu, informazioa gehitu...
- 2017-07-16 (2h 30min): Azterketa osatu da irizpideak eta informazio gehigarria sartuz. Ondorioak ere sakondu dira.

MEMORIA

- 2017-04-29 (2h 30min): *Aurrekariak* eta *Testuingurua* atalak idatzi dira memoriarako. Agian zerbait gehiagorekin osatu daitezke.
- 2017-05-12 (30min): *Arriskuen identifikazioa* eta *Interesatuak* atalak idatzi dira.
- 2017-06-01 (1h): Memoriak izango dituen atalak zehaztu dira eta dokumentua egituratu da atalka.
- 2017-06-11 (30min): Orain arte idatzitako memoriara pasa da eta formatuak egokitu dira. Sarrera atalean pare bat paragrafo gehitu dira.
- 2017-07-16 (45min): Tutoreak aipatutako zuzenketa ortografiko eta gramatikalak egin dira. Sarrera atalean aipatutako moldaketak egin dira.

- 2017-07-21 (2h 15min): Memoriako 2. atala osatu da (kudeaketa-plana) aurrez osatuta zeuden gauzak txertatuz eta falta zirenak eginez. LDEa moldatu behar izan da bertikalean sartzeko. GANTT erabatekoa sartzea falta da.
- 2017-07-22 (6h): Gantt diagrama bukatu eta memorian sartu da. Diseinu-atala erdi-osatu da baina hainbat zalantzarekin (zalantzak iruzkin modura adierazi dira). Erabilpen-kasuen diagrama zati sartu da hainbat azalpenekin eta bezeroaren eskakizuna adierazi dira.
- 2017-07-23 (3h): Ondorioak atala egin da.
- 2017-07-24 (3h): Inplementazioa atala osatu da. Ez dira interfazeak eta arazoak sartu proiektua oraindik ez dagoelako amaituta. 4. atala lantzen hasi da. Tutoreari memoria bidali zaio, berrikusi dezan.
- 2017-07-25 (1h 30min): Meteori buruzko atala osatu da azalpen orokor batzuk emanaz eta kode zatiak gehituz.
- 2017-08-03 (2h): Tutoreak emailaz helarazitako zuzenketak hasi dira. Gramatika eta ortografia zuzenketetan 5.2 atalera arte aurreratu da.
- 2017-08-04 (2h30min): Falta ziren zuzenketa ortografiko eta gramatikalak egin dira eta 1.3 atala berridatzi da.
- 2017-08-07 (4h): 4. atalean tutoreak adierazitako aldaketak egin dira, informazio orokorreko hainbat paragrafo gehituz.
- 2017-08-08 (2h15min): 3. atalean egin beharreko zuzenketak egin dira eta Vert.x eta Twisted atalak zabaldu dira.
- 2017-08-09 (2h30min): Erabilpen-kasuen diagraman eta klase-diagraman beharrezko aldaketak egin dira dokumentuan modu egokian adierazteko (diagramak moldatu hobeto sartzeko, bitan banatu...). Extends eta include klausulak errepasatu behar dira oraindik.
- 2017-08-13 (2h): Aurkibidea osatzeko konfigurazioak egin dira.
- 2017-08-23 (7h): Interfazeen harrapaketak sartu dira memorian euren azalpenekin batera, arazoak eta konponbideak azaldu dira bere atalean. 6.2 eta 6.4 atalak amaituta daude.
- 2017-08-24 (8h): *Lan-karga eta desbideraketak* atala osatu da eta baita aldaketen atala ere. Excel dokumentuan errore bat zegoela ikusi da eta kalkuluak eskuz berriz egin behar izan dira. Klase-diagraman azalpena gehitu da. Orrialde bakoitiak beste aldera pasatu dira eta kapitulu hasierak ondo kokatu dira. Zenbaki erromatardun orriak aurkibidean sartzea lortu da *agregar texto* aukera erabiliz.
- 2017-08-25 (8h): Sekuentzia-diagramak osatu eta asko berregin dira, ez baitzeuden modu zuzenean eginda.
- 2017-08-26 (4h): Sekuentzia-diagrama guztiak osatu, zuzendu eta memorian gehitu dira gertaera-fluxuekin batera.
- 2017-08-27 (8h): Taula eta irudi zerrendak egin dira, eta erreferentziak sartu. Memoria guztia errepasatu da tutoreari entregatu aurretik. Zuzenketa gramatikal batzuk egin dira.
- 2017-09-01 (8h): Tutoreak helarazitako zuzenketak egiten hasi da. Ortografia eta gramatika zuzenketak egin dira baina tarte pare bat aldiz funtzionatzeari uzti dio Word-ek. Honek lanaren zati bat berregin behar izatea eragin du eta, ondorioz, atzerapena izan da. Sekuentzia diagrametatik 6 eranskinetara pasa dira eta ondorioen atalaren aurretik jarraipen eta kontrola kokatu da.
- 2017-09-02 (4h): Aldaketeekin jarraituz, orriak ondo kokatuta daudela konprobatu da (bakoitia eskuinaldean) eta PDF-ra esportatuz deskonfiguraziorik gertatzen ez dela konprobatu da. Erabilpen-kasuen diagrama eta "Eztabaida sortu" erabilpen-kasuaren sekuentzia-diagrama horizontalean jarri dira irakurgarriagoak izateko.

Egindako aldaketak ondorioetan aipatu dira. Testu-formatuak arauetan eskatutakoetara egokitu dira.

- 2017-09-03 (4h): Erreferentzia nagusiak gehitu dira eta jarraipen eta kontrola eguneratu. Dedikazioa eta desbideraketak eguneratu dira ere. Dokumentu osoa goitik behera errepasatu da ADDI bitartez entrega egin aurretik.

B Eranskina: Bilera-aktak

BILDUTAKOAK: Mikel Agirre eta David Buenestado	DATA: 2017-01-10	TOKIA: Informatika Fakultatea
BILERA ZK.: 1	ORDUA: 15:00	IRAUPENA: 30 min
JORRATUTAKO GAIK: <ul style="list-style-type: none"> • Proiektuaren hasierako egoera. • Eman beharreko lehen urratsak. • Proiektuaren inguruko zalantzak argitzea. • Proiektuaren ezaugarri teknikoak 		
ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK: <ul style="list-style-type: none"> • Interesgarria izango litzateke teknologia berritzaileren bat erabiltzea proiektua burutzeko. Aukera ezberdinak aztertu behar dira. • Ahalik eta azkarren bezeroarekin bilera bat zehaztuko da eskakizunak aztertzeko. • Bezeroarekin hitz egin ondoren bilera berri bat egingo da. • Aurreko urteetako proiektuen memoriak begiratzea, egitura eta jorratu ziren atalak ezagutzeko. Antza duen proiektu bat bada, hobe. 		

BILDUTAKOAK: Mikel Agirre eta Jimmy Brossa (Gautenako arduraduna)	DATA: 2017-01-16	TOKIA: Gautenako bulegoak (Aiete auzoa, Donostia)
BILERA ZK.: 1	ORDUA: 10:00	Iraupena: 45 min
JORRATUTAKO GAIK: <ul style="list-style-type: none"> • Dagoeneko garapenean dagoen webgunearen egoera. • Gune pribaturako eskakizunak. • Gautenari buruzko hainbat datu. 		
ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK: <ul style="list-style-type: none"> • Garapenean dagoen webgunea enpresa batek Wordpress bidez garatzen du. Enpresak musu-truk eta denbora dutenean burutzen du lan hori; beraz, horren osagarria den gune pribatua sortzea da proiektuaren helburua. • Gautenako bazkide direnentzat izango da gune pribatua. Batzuk ez dira bazkide Osakidetza bidaltzen dituelako edo zerbitzu bera dutelako ez izanda ere. Bazkideek edukiko dute pasahitza sartzeko. • Webaren garatzaileak (doan eta denbora dutenean egiten dute) gune pribatu 		

minimo bat egingo dute. Zerbait osatuagoa eta bateragarria dena egitea izango litzateke nire lana

- Gunearen edukiak (eskakizunak):
 - Foroa. Ohiko foro bat bezalakoa, familiek hitz egin ahal izateko.
 - Biltegi digitala. Gidak eta argazkiak biltegitzeko. Asko (batzuk tamaina handikoak). Biltegitze-tamaina handia beharko da.
 - Formakuntza, Online ikastaroak familientzat. Dokumentuak jaitsi eta igotzeko aukera. Arituak berrikuspena egiteko aukera (Moodle baten modukoa).
 - Jarduerentzako izen-ematea (Moodle modukoa).
 - Jakinarazpenak posta bidez zerbait berria dagonean gunean.
 - Responsive izatea ondo legoke baina ez da beharrezkoa. Erabiliko duten gehienek ordenagailutik egingo dute
- Jimmyk Gautenari buruzko hainbat datu praktiko bidaliko dizkit emailaz, aurrekarietan erabiltzeko.

BILDUTAKOAK: Mikel Agirre eta David Buenestado	DATA: 2017-01-30	TOKIA: Informatika Fakultatea
BILERA ZK.: 2	ORDUA: 11:15	IRAUPENA: 15 min
JORRATUTAKO GAIK: <ul style="list-style-type: none"> • Bezeroarekin izandako bileraren berri ematea. • Proiektua osatzeko ideiak aztertzea. 		
ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK <ul style="list-style-type: none"> • Bezeroak eskakizunak nik egin dezakedanaren arabera malguak direla ikusita, proiektua osatzeko teknologia eta framework konparaketa bat egingo da, non web-aplikazioa garatzeko erabiliko den tresna aztertuko den ere. • Konparaketak javascript lengoia izango du oinarritzat eta berau bezero eta zerbitzarian erabiltzen duten teknologia eta frameworkak aztertuko dira. • Informazioa bildu behar da gai honen inguruan existitzen diren aukera ezberdinei buruz. 		

BILDUTAKOAK: Mikel Agirre eta David Buenestado	DATA: 2017-05-08	TOKIA: Informatika Fakultatea
BILERA ZK.: 3	ORDUA: 15:00	IRAUPENA: 45 min
JORRATUTAKO GAIK: <ul style="list-style-type: none"> • Zuzenketak: klase-diagramak, erabilpen-kasuen diagrama, Gantt diagrama... • Proiektuaren egoera eta denbora. • Proiektuaren irismena. 		

- Interfazeak.
- Garapen-inguruneari buruz eztabaidatu.
- Hurrengo bilera zehaztu.

ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK:

- Gantt diagraman lan-paketeak kronologikoki ordenatuak egon behar dira (*waterfall*).
- Erabilpen-kasuen diagrama: 3 aktoreak espezializazioak direnez, "Aktorea" aktorea adierazi behar da bertan. Multzokatze-laukiak ezabatu behar dira.
- Klase-diagrama erreparatu behar da, erabilpen-kasu guztiak betetzeko beharrezko klase eta loturak adierazita egon daitezten (Erabiltzaile-zeregin lotura, esaterako).
- Sekuentzia-diagramak osatu behar dira inplementatzen hasi aurretik, erabilpen-kasuen arteko dependentziak erakusteko (ez dira erabilpen-kasuen diagraman adierazten).
- Lan-paketeen deskonposaketa egokia da baina pakete bakoitzari esleitutako denbora ez. Web-aplikazioak % 80 eta Azterketak % 20a (gutxi gora behera) izango dutela erabaki da.
- Foroa: Bezeroari alderdi honen egitura nolakoa izatea nahi duen galdetu behar zaio (iruzkin zehatzei erantzun etab.). Foro moduko bat (elkarrekintza-sistema) inplementaturik ba ote dagoen begiratu behar da, proiektuan berrerabilgarria izan daitekeena. Gauza bera egin behar da biltegiatze-sistamarako.
- Gautenak zerbitzari propiorik baduen galdetu behar da, aplikazioak non egon behar duen zehazteko beharra dago eta. Agian beharrezkoa izango da webgunea garatzen duen enpresarekin hitz egitea. Proiektua aurkezteko soluzioak aztertuko dira Node.js-rako (zerbitzari iraunkor batean jartzeko beharrik gabe).
- Interfazeen aldetik, egokia izango litzateke Meteor-ek erabiltzen dituen aurredefinitutako txantiloak aztertzea eta horietako bat moldatuz lan egitea.
- Proiektuaren irismena aplikazio mantengarri bat egitea izango litzateke, bezeroak ebaluatzeko. Proiektu pilotoa egitea baztertu da, denbora faltagatik.
- Garrantzitsua izango da diseinua egiten den bitartean frameworken azterketa egitea, Meteor-en aukeraketa ondo justifikatzeko.
- Hurrengo bileraren data zehaztu da: maiatzak 18, osteguna, 15:00etan (leku berean).

BILDUTAKOAK: Mikel Agirre eta David Buenestado	DATA: 2017-05-18	TOKIA: Informatika Fakultatea
BILERA ZK.: 4	ORDUA: 15:00	IRAUPENA: 1h
<p>JORRATUTAKO GAIK:</p> <ul style="list-style-type: none"> • Login sistemaren arazoak. • Zuzenketak: klase-diagrama, erabilpen-kasuen diagrama. • Konparaketaren ikuspuntua (bidalitakoaren zuzenketa). • Bezeroaren zerbitzaria. • Meteor-en aukerak foro eta biltegiatzerako. • Hurrengo bileraren data. 		

- MongoDB-ren egokitasuna.

ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK:

- Meteor-en berezko login-ak hainbat erabiltzaile mota ez onartu arren, ez dago arazorik erabiltzaile bera erabili eta atributu ezberdinak esleitzen badizkiogu (posible da egitea). Erabiltzaile bakoitzari dagokion rola esleituko zaio (nola egin aztertu behar da, baina egin daiteke).
- Klase-diagrama: Mezua klasea aldatu behar da erabiltzaile eta adituak beste klase baten espezializazio bihurtuz eta hau Mezua klasearekin erlazionatuz; soberan dauden atributuak erlazio zuzenengatik aldatu behar dira.
- Erabilpen-kasuak multzokatzea komenigarria da sistemak erabiliz, falta diren `include`-ak jarri behar dira. Login-a beste guztiak burutzeko beharrezkoa denez, ez da erlazioa marraztuko baina azalpenean argi eta garbi aipatuko da. Geziak ez gainjartzea eta bestelako txukuntasun kontuak garrantzitsuak dira.
- Bidalitako konparaketan ez da modu egokian azaldu proiektuaren ikuspuntua, hau da, teknologia jakin baten azterketa eta framework ezberdinen konparaketa objektiboa. Konparaketa amiatzerakoan taula batean ezaugarri ezberdinak (konparazio-dimentsioak) sailkatzeko moduan egotea da ideia. Konparaketa hasteko, ingurune isomorfikoen kontzeptua eta ezaugarri nagusiak azalduko dira, ondoren abantaila eta desabantailak aztertuko. Ikuspuntu hori aldatuz berriro idatzi behar da.
- Askok eztabaidatu ondoren, ondorioztatu da MongoDB ez-erlazionala izateak ez duela zertan web-aplikazioaren funtzionamendu egokia oztopatu. Aplikazio nahiko estatikoa izanik ez da transakzioetan konkurrentzia handirik egotea espero.
- Bezeroak zerbitzariarekin dituen arazoak ikusita, ez gara horretan sartuko eta proiektua aurkezteko garaian *host* batean egotea nahikoa izango litzateke.
- Foro eta biltegiatzerako aukerak oso itxura ona dutela erabaki da eta egokiak izan daitezkeela proiekturako.
- Hurrengo bilera maiatzaren 30ean izango da leku eta ordu berean. Bilera horretarako, lehenetsun nagusia kodetzea da, eta komenigarria izango litzateke login-a eginda edukitzea eta baita aktore bakoitzarentzako oinarritzko erabilpen-kasu ahalik eta gehienak.

BILDUTAKOAK: Mikel Agirre eta David Buenestado	DATA: 2017-05-30	TOKIA: Informatika Fakultatea
BILERA ZK.: 5	ORDUA: 15:00	IRAUPENA: 30 min
JORRATUTAKO GAIK: <ul style="list-style-type: none"> • Proiektuaren egoera aztertzea: egindako erabilpen-kasuak, arazoak... • Proiektuaren entrega-deialdia zehaztea. • Hurrengo bilera zehaztu. 		
ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK: <ul style="list-style-type: none"> • Web-aplikazioaren erabilpen-kasuen implementazioa modu egokian doa. Aparteko arazorik ez bada sortzen, bi asteren buruan implementazioa amaituta egon 		

daitekeela aurreikusi da (betiere kasu hoberenean eta aurkezpenaren inplementazioa kontutan hartu gabe).

- Oraingo bai ikasleak bai tutoreak erabilpen-kasuak inplementatzeari lehentasuna ematen diote, ondoren aurkezpena lantzeko.
- Aurkezpenaren inguruko xehetasunak bezeroarekin komentatzea lagungarria izango da, proiektu osatuagoa lortzeko.
- Orain inplementazioak erritmo ona hartu duela, egokia izango litzateke dokumentazioarekin batera lantzea (sekuentzia-diagramak, diagrametan aldaketak...).
- Proiektuaren egoera eta eskatzen duen denbora ikusita, bi parteak ados daude proiektua iraileko deialdian entregatuko dela. David Buenestado informatuko da matrikulazioaren protokoloari buruz.
- Hurrengo bilera bi aste barrurako zehaztu da: ekainak 13, asteartea, 15:00etan (leku berean).

BILDUTAKOAK: Mikel Agirre eta David Buenestado	DATA: 2017-06-13	TOKIA: Informatika Fakultatea
BILERA ZK.: 6	ORDUA: 15:00	IRAUPENA: 1h
<p>JORRATUTAKO GAIK:</p> <ul style="list-style-type: none"> • Inplementazioaren egoera. • Memorian zuzenketak, memoriaren atalak. • Sekuentzia-diagrametan zalantzak. • Erabilpen-kasuen diagramaren diseinua. • Plangintzaren moldaketa. • Hurrengo bileraren data. 		
<p>ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK:</p> <ul style="list-style-type: none"> • Memorian egin beharreko zuzenketak dokumentuko iruzkinen bidartez adieraziko ditu Davidek eta Mikeli helaraziko dizkio. Egin beharreko aldaketen ideia nagusia hau da: lehentasuna ematea teknologia-azterketari web-aplikazioaren garapenari baino. • Memorian garapenerako baliabideei buruzko atal bat jarriko da, web-aplikazioaren diseinuaren atalaren aurretik. Erabiltzaile-gida bat ere egingo da, eta garatzaile-gida bat denbora egongo balitz. • Erabilpen-kasuen diagrama osoa erakutsiko da txostenean, baina gero sistemaka zatituko da xehetasunak modu egokian ikusteko. Koloreak mantenduko dira. • Sekuentzia diagrametan ez dira aurredefinitu paketeen fitxategiek osatzen dituzten funtzionalitateak xehetu behar. • Hurrengo bilera ekainaren 27an izango da, leku eta ordu berean. 		

BILDUTAKOAK: Mikel Agirre eta David Buenestado	DATA: 2017-06-19	TOKIA: Informatika fakultatea
BILERA ZK.: 7	ORDUA: 12:00	IRAUPENA: 15 min
JORRATUTAKO GAIK: <ul style="list-style-type: none"> • GRALaren zuzendari-aldaketa. 		
ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK: <ul style="list-style-type: none"> • David Buenestadok jakinarazi du arazo pertsonalengatik ezingo duela proiektua zuzentzen jarraitu. Xabier Artolarekin hitz egitea erabaki da, berak proiektua zuzen dezaken galdetzeko. Ostegunean 16:30ean bilera bat egingo da Xabier Artolaren bulegoan gaiari buruz hitz egiteko. 		

BILDUTAKOAK: Mikel Agirre, Xabier Artola eta David Buenestado	DATA: 2017-06-22	TOKIA: Informatika Fakultatea
BILERA ZK.: 1	ORDUA: 16:30	IRAUPENA: 30 min
JORRATUTAKO GAIK: <ul style="list-style-type: none"> • Tutore berriari proiektuaren egoera jakinaraztea. • Tutore berriari proiektuaren inguruko asmoak jakinaraztea. • Tutore-aldaketari buruzko burokrazia. 		
ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK: <ul style="list-style-type: none"> • Xabier Artola, tutore berriari, proiektua inplementazio eta memoria aldetik zein puntutan dagoen azaldu eta erakutsi zaio. Xabier Artolari memoria bidaliko zaio berrikus dezan. • Proiektuaren egoera, denbora eta lehenetsunak ikusita, tutoreari Online Ikastaroei lotutako erabilpen-kasuak garatzea bertan behera uztea proposatu zaio. • Xabierrek memoria berrikusiko du eta ondoren Mikelekin jarriko da kontaktuan. • David Buenestadok idazkaritzan jakinaraziko du tutore-aldaketa eta ofizialki Xabier Artola izango da tutore berria. 		

BILDUTAKOAK: Mikel Agirre eta Xabier Artola	DATA: 2017-07-17	TOKIA: Informatika Fakultatea
BILERA ZK.: 2	ORDUA: 12:00	IRAUPENA: 1h
JORRATUTAKO GAIK: <ul style="list-style-type: none"> • Proiektuaren egoerari buruz hitz egitea: memoria, inplementazioa, epeak... 		

- Hurrengo asteetan komunikatzeko modua zehaztu.

ATERATAKO ONDORIOAK - HARTUTAKO ERABAKIAK:

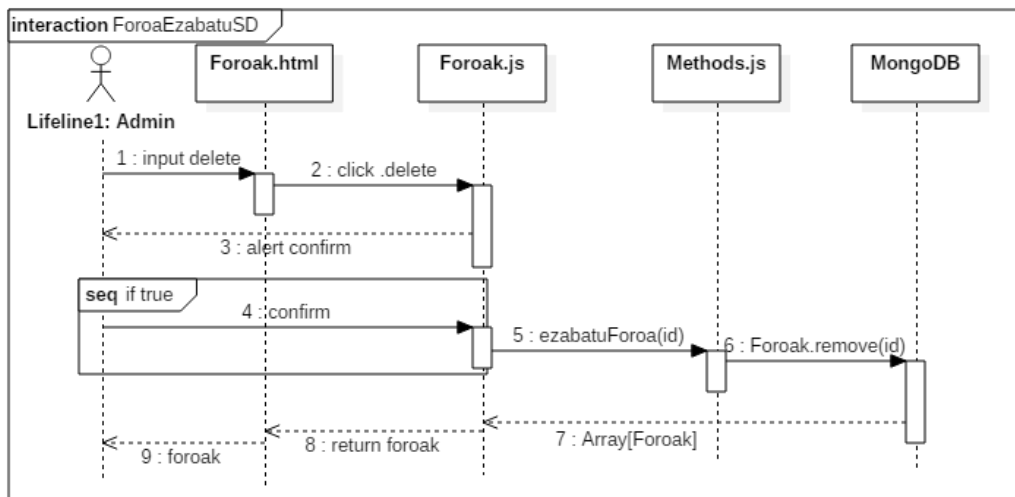
- Memoriaren egiturari buruz, tutoreari ondo iruditu zaio eginda dagoen guztia eta egitekoa antolatzeko modua.
- Proiektua irailean entregatuko da. Mikelek defentsa-eskaera egingo du eta Xabierri jakinaraziko dio. Xabierrek aldeko txostena helaraziko du, ondoren.
- Datorren astelehenerako memoria ahalik eta gehien osatua bidaliko zaio tutoreari berak begiratu dezan. Ondorioak, diseinu- eta plangintza-atalak eginda egon beharko lirateke.
- Sekuentzia-diagramak eranskin modura jarriko dira.

C Eranskina: Sekuentzia-diagramen bilduma osoa

Eranskin honetan, gainontzeko erabilpen kasuen sekuentzia-diagramak eta euren gerater-fluxuak aurkeztuko dira.

FOROA EZABATU

Erabilpen-kasu honen sekuentzia-diagrama baliokidea da hurrengo erabilpen-kasuenekin: "Jarduera ezabatu", "Karpeta ezabatu", "Iruzkina ezabatu", "Dokumentua ezabatu", "Argazkia ezabatu", "Aktorea ezabatu" eta "Berria ezabatu".



29. irudia: "Foroa ezabatu" erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Administratzaile batek foro bat ezabatzen du.

Aktorea: Administratzailea.

Aurrebaldintzak: Saioa hasita eta foroak atalean egotea.

Postbaldintzak: Foroen zerrenda eguneratuko da.

Ohiko fluxua:

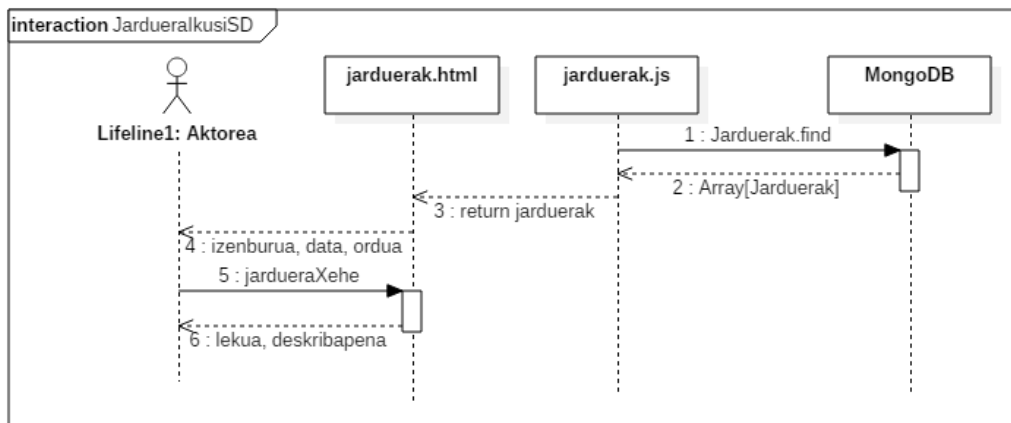
1. Administratzaileak ezabatu nahi duen foroan gurutzea sakatu.
2. Mezu bat bistaratzen da pantailan ezabatzea baieztatu edo ezeztatzeko.
3. Baieztatuz gero, foroa datu-basetik ezabatu eta zerrenda eguneratua bistaratuko da.

Ezohiko fluxua:

4. Ezeztatuz gero, ezabatzea bertan behera geratuko da eta orria zegoen bezala erakutsiko da.

JARDUERA IKUSI

Erabilpen-kasu honen sekuentzia-diagrama baliokidea da hurrengo erabilpen-kasuenekin: “Izena emandakoen zerrenda ikusi” (administratzailea) eta “Izena emandako jarduerak ikusi” (erabiltzaile arrunta, “Nire jarduerak” atalean).



30. irudia: “Jarduera ikusi” erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Aktore batek erabilgarri dauden jarduerak eta bakoitzaren xehetasunak ikusten ditu.

Aktorea: Aktorea (erabiltzaile arrunta, administratzailea edo aditua).

Aurrebaldintzak: Saioa hasita eta jarduerak atalean egotea.

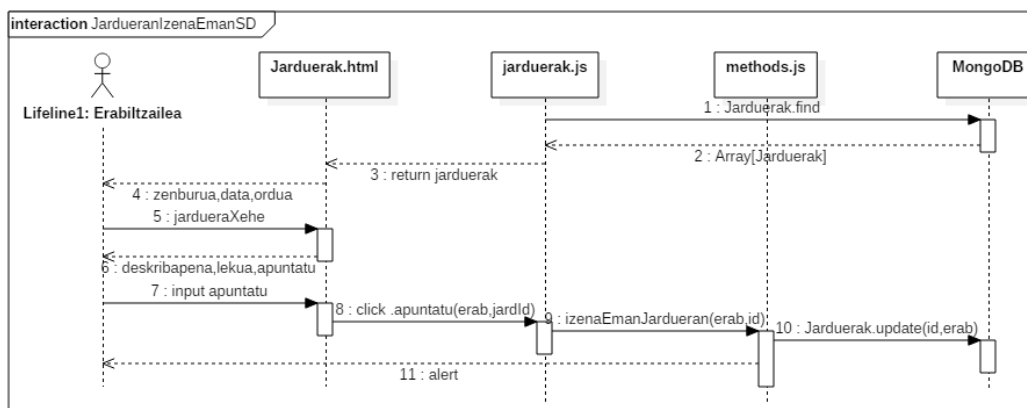
Postbaldintzak: Jarduerak bistaratuko dira.

Ohiko fluxua:

1. Erabiltzailea erabilgarri dauden jarduerak ikusten ditu (izenburua, data eta ordua).
2. Xehetasun gehiago ikusi nahi baditu, batuketa ikurra duen “+” botoia sakatu eta lekua eta deskribapena bistaratuko dira.
3. Xehetasunak berriro botoi bera sakatzen ezkuta daitezke.

Ezohiko fluxua: ez dago ezohiko fluxurik.

JARDUERAN IZENA EMAN



31. irudia: “Jardueran izena eman” erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Erabiltzaile batek jarduera batean izena ematen du.

Aktorea: Eerabiltzaile arrunta.

Aurrebaldintzak: Saioa hasita eta jarduerak atalean egotea.

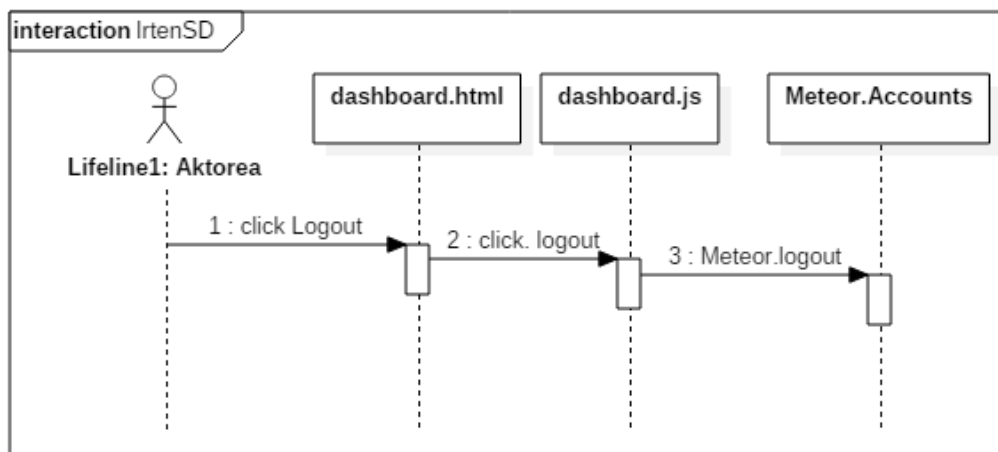
Postbaldintzak: Datu-basea eguneratuko da.

Ohiko fluxua:

1. Erabiltzailea erabilgarri dauden jarduerak ikusten ditu (izenburua, data eta ordua).
2. Xehetasun gehiago ikusi nahi baditu, batuketa ikurra duen “+” botoia sakatu eta lekua, deskribapena eta izena emateko botoia bistaratuko dira.
3. Xehetasunak berriro botoi bera sakatzen ezkuta daitezke.
4. “Joango naiz” botoia sakatzean, izen-ematea burutuko da eta mezu baten bitartez jakinaraziko da.

Ezohiko fluxua: ez dago ezohiko fluxurik.

IRTEN



32. irudia: “Irten” erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Aktore batek saioa amaitzen du.

Aktorea: Aktorea (erabiltzaile arrunta, administratzailea edo aditua).

Aurrebaldintzak: Saioa hasita eta menu nagusian egotea.

Postbaldintzak: Saioa amaituko da eta “Login” orrira birbideratu.

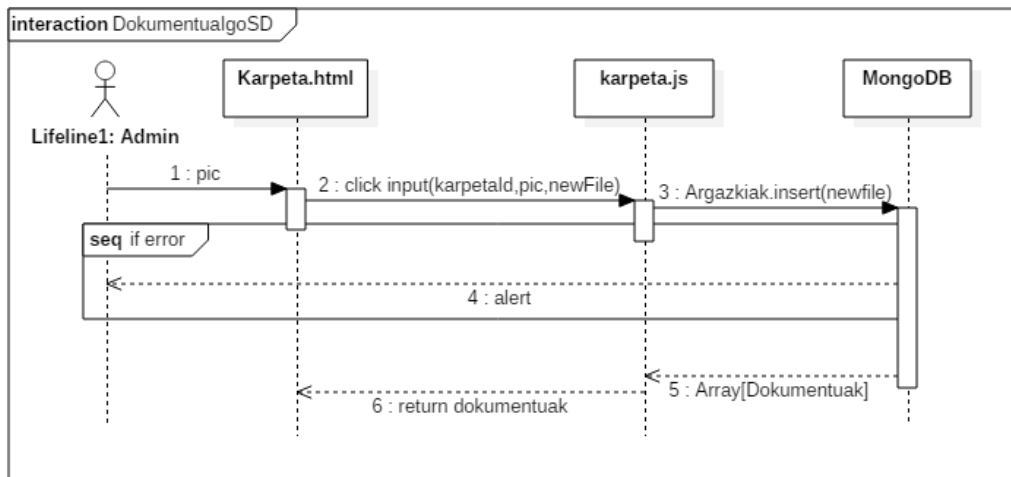
Ohiko fluxua:

1. “Logout” esteka sakatzean, saioa amaitu eta “Login” orrira birbideratuko da erabiltzailea.

Ezohiko fluxua: ez dago ezohiko fluxurik.

DOKUMENTUA IGO

Erabilpen-kasu honen sekuentzia-diagrama baliokidea da hurrengo erabilpen-kasuenekin: "Argazkia igo".



33. irudia: "Dokumentua igo" erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Administratzaileak dokumentu bat igotzen du karpeta batera.

Aktorea: Administratzailea.

Aurrebaldintzak: Galeria motakoa den karpeta batean ez egotea.

Postbaldintzak: Dokumentu bilduma eguneratua bistaratuko da.

Ohiko fluxua:

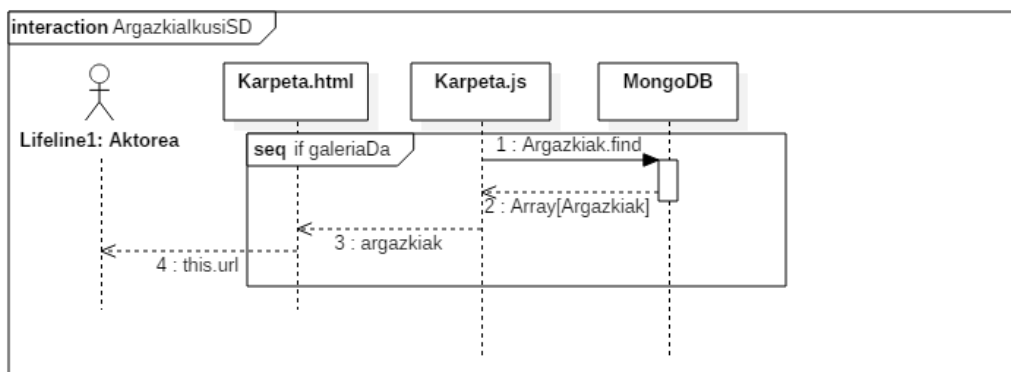
1. "Arakatu" botoia sakatuta dokumentu bat aukeratzeko da.
2. "Dokumentua igo" botoia sakatuta, datu-basean gehituko da dokumentua eta bilduma eguneratu.

Ezohiko fluxua:

3. Ez bada fitzategirik aukeratzeko aplikazioak ez du ezer egingo.

ARGAZKIA IKUSI

Erabilpen-kasu honen sekuentzia-diagrama baliokidea da hurrengo erabilpen-kasuenekin: "Dokumentua jaitsi".



34. irudia: "Argazkia ikusi" erabilpen-kasuaren sekuentzia-diagrama.

Deskribapena: Aktore batek argazki bat ikusten du tamaina osoan.

Aktorea: Aktorea (erabiltzaile arrunta, administratzailea edo aditua).

Aurrebaldintzak: Galeria motakoa den karpeta batean egotea.

Postbaldintzak: Leiho berri batean argazkia bistaratuko da tamaina osoan.

Ohiko fluxua:

1. Argazkiaren estekan sakatu eta leiho berri bat irekiko da, argazki osoarekin.

Ezohiko fluxua: ez dago ezohiko fluxurik.