

▪ Gradu Amaierako Proiektua ▪

Software Ingeniaritza

Mantentze-kudeaketa softwarea diseinatu eta garatu,
Tolosaldea institutuko mantentze-lanak kudeatzeko.

Alex Aranburu

2017- ekaina

Laburpena

Gradu amaierako proiektu honetan, Tolosaldea Instituturako mantentze-kudeaketa sistemaren diseinua eta inplementazioa egitea izan dugu helburu, beti ere software librean oinarritua. Matxuraren bat sumatzen denean, aplikazioaren bitartez erabiltzaileak mezua bidaltzen dio mantentze-kudeatzaileari, argazki eta guzti eta, honek, konponketa bideratzen du, behar den lana eginez. Horretaz gain, historikoak eta bestelako datuak ere jaso eta esportatzeko aukera ematen du.

Horrela, aplikazioak Tolosaldea Institutuaren mantentze-lanen kudeaketa ahalbidetuko du, mahaigaineko ordenagailuetan, tablet edo mugikorretan atzigarri izanez.

Gaien Aurkibidea

Laburpena	iii
Gaien Aurkibidea.....	v
1. Sarrera	1
2. Proiektuaren kudeaketa plana	3
2.1. Helburuak	3
2.2. Irismena.....	3
2.2.1 Betekizunak	4
2.2.2 Irismen mailak	4
2.2.3 Emangarri nagusiak.....	5
2.3. LDE Diagrama.....	6
2.3.1 Kudeaketa	6
2.3.2 Produktua.....	7
2.3.3 Dokumentazioa	8
2.4. Mugarriak	9
2.5. Metodologia	10
2.6. Gantt diagrama	10
2.7. Kalitate-plana	10
2.7.1 Aplikazioaren kalitatea	11
2.7.2 Memoriaren kalitatea	11
2.9. Interesatuak.....	13
2.9.1 Komunikazioa	13
3. Aurrekariak eta erabilitako teknologia	15
3.1. Aurreikusitako teknologiak.....	15
3.1.1 Aplikazioa	15
3.1.2 Datu-basea	17
3.1.3 Zerbitzaria	17
3.1.4 Bestelakoak	18
4. Analisia eta diseinua	21
4.1 Analisia	21
4.1.1 Eskakizun-bilketa	21

4.2	Aztertutako aplikazioa.....	22
4.2.1	Mantis[24].....	22
4.2.2	Veedor[25]	24
4.3	Diseinua.....	24
4.3.1	Rolak	24
4.3.2	Erabilpen kasuak	25
4.2.3	Datu-basea	47
4.2.4	Klase-diagrama.....	48
5.	Software-garapena	51
5.1	Garatutako ezaugarriak	51
5.1.1	Hasiera orria/Saioa hasi.....	51
5.1.2	Menua.....	52
5.1.3	Abisua bidali	53
5.1.4	Abisuak ikusi.....	53
5.1.5	Lanaren xehetasunak ikusi	54
5.1.6	Lana osatu	55
5.1.7	Datuak esportatu	55
5.1.8	Lan prebentiboak sortu	57
5.1.9	Erabiltzaileak sortu	58
5.1.10	Erabiltzaileak editatu/ezabatu.....	59
5.1.11	Abisuak editatu.....	60
5.2	Arazoak.....	60
5.2.1	Segurtasun arazoak.....	60
6.	Probak eta hobekuntzak	65
6.1	Lazy Loading[35]	65
6.2	Datables[37].....	66
7.	Jarraipena eta kontrola.....	67
7.1	Proiektuaren garapena	67
7.2	Kalitatea	68
7.2.1	Aplikazioaren kalitatea	68
7.2.2	Memoriaren kalitatea	69
7.3	Arriskuak	69
8.	Ondorioak	71
8.1	Lortutako emaitza	71
8.2	Ebaluazio pertsonala	71

8.3	Etorkizunerako aukerak.....	73
	Bibliografia	74
	A ERANSKINA.....	76

Irudi eta Taulen zerrenda

1 irudia. Kudeaketari dagokion Gantt diagramaren zatia.	6
2. irudia. Produktuari dagokion LDE diagramaren zatia.	8
3.irudia. Dokumentazioari dagokion LDE diagrama.	9
4. irudia. Proiektuari dagokion Gantt diagrama.	10
5. irudia. "2017ko apirilean bildutako nabigatzaileen erabilera datuak.	15
6.irudia. Git-en erabilera zuzena azaltzen duen irudia.	19
7. irudia. Mantis-en leiho nagusia, abisuak kategoriatan banatuta.	23
8. irudia. Mantisen abisua bidaltzeko leihoa.	23
9. irudia. Mantis-en abisuaren inguruko datuak ikusteko leihoa.	24
10. irudia. Rolak eta hauei lotutako erabilpen kasuak.	26
11 irudia. Saioa hasi erabilpen kasuaren sekuentzia diagrama.	28
12 irudia. Saioa amaituren sekuentzia diagrama.	29
13 irudia. Abisua bidaliren sekuentzia diagrama.	31
14 esleitutako abisuak sekuentzia diagrama.	33
15 irudia. Abisuaren xehetasunak ikusti erabilpen kasuaren sekuentzia diagrama. .	35
16 irudia. Lana osatu erabilpen kasuaren sekuentzia diagrama.	37
17 irudia. Abisua editatu erabilpen kasuaren sekuentzia diagrama.	39
18 irudia. Datuak exportatu erabilpen kasuaren sekuentzia diagrama.	41
19 irudia. Erabiltzailea sortu erabilpen kasuaren sekuentzia diagrama.	44
20 irudia. Erabiltzailea editatu erabilpen kasuaren sekuentzia diagrama.	46
21. irudia. Datu basearen diagrama.	47
22 irudia. Aplikazioaren orrien egitura.	49
23.irudia. Hasiera orria, saioa hasteko leihoa.	52
24.irudia. Aplikazioaren menu nagusia, administratzailearen ikuspegia.	52
25. irudia. Abisua bidaltzeko leihoa.	53
26. irudia. Abisuak ikusteko leihoa.	54
27. irudia. Abisuaren xehetasunak.	54
28. irudia. Lana osatzeko erlaitza.	55
29 irudia. Datuak esportatzeko leihoa.	56
30. irudia. Abisu prebentiboak sortzeko leihoa.	57
31 irudia. Abisu prebentiboen jakinarazpena.	58
32. irudia. Erabiltzaile berria sortzeko leihoa.	59
33 irudia. Erabiltzaile zerrenda.	59
34.irudia. Abisua editatzeko leihoa.	60
35 irudia. Kode ahula.	61
36 irudia. Sortuko den SQL sententzia.	62
37 irudia. Espero den sententzia zuzena.	62
38 irudia. Aldi bereko sententzia, kode arrotza exekutatu diguna.	63
39 kodea. Irudien formatua zuzena dela bermatzeko kodea.	64
40. irudia. Datatables pluginaren APIari deia egiteko modu ezberdinak.	66
41. irudia. APIari egindako deia.	66
42 taula. Proiektuan estimatutako ordu eta benetako orduen arteko desbideraketak.	68

1

Sarrera

Zer da mantentzea?

Mantentzea eraikin, gailu eta bestelako sistemak erabilera-kondizio egokietan izan daitezen eta matxuren arriskua gutxitzeko egindako lan multzoa da. Sistema horiek hondatutakoan, horiek konpondu eta erabiltzeko prest jartzeko lanak ere barnebiltzen ditu. Esan genezake, gizakia sortu zenean sortu zela mantentzea, bai bizilekuak, baita lehen tresnak ere, nolabaiteko zaintzea edo mantentze lanak behar zituztelako. Ataza horri, gaur egun, mantentze-zuzentzailea deituko genioke, hau da, behin erabilera-kondizioa egokia ez denean egin beharreko mantentzea.

Gaur egun ezagutzen dugun mantentzea, makinaren mantentze-kudeaketa, industria-iraultzarekin sortu zela esan dezakegu. Hasieran, industria sortu zenean, makina bakoitzaz arduratzen zen langileak hartzen zuen konponketaren ardura. Baina, XX. mende hasieran, industrietako makina kopurua handitzen zihoan heinean, geroz eta denbora gehiago eskatzen zuten mantentze-lanek eta, lan hauek makina horretako produkzioa geratzen zenez, produkzioa jaitsi eta, ondorioz, diru galerak ez izateko, soluzioa pentsatzen hasi ziren makinaren jabeak. Horrela, makinekin aritzen ziren langileez gain, mantentze-lanak egiteko beste hainbat langile kontratatzen hasi ziren, mantentzean espezializatutako langileak sortuz.

Baina, beharrak geroz eta handiagoak zirenez, matxurak konpontzeaz gain, horiek aurreikusi eta ekiditeko mantentzea proposatu zuten, matxurak sortutako atzerapenak ekiditearren. Horrela, prebentziozko mantentzea sortu zen. Hartara, makinak denbora luzeagoz funtzionatzea lortuta, produkzioa handitzen zen. Ondoren, gaur arte, hainbat hobekuntza izan ditu mantentze-kudeaketak, produkzio mailak izugarri handituz. Horregatik, gaur egun izugarritzko garrantzia ematen zaio mantentze-kudeaketari eta, beste hainbat arlo bezala, informatizatzeke joera hartu du. Honen bidez, lan kudeaketa arindu eta azkartzeaz gain, bateratzea lortu da.

Beraz, manentze-kudeaketaren garrantzia ikusteak bultzatu gaitu proiektu hau garatzera, benetan hain baliagarria izango den softwarea sortzeak motibazio extra bat ematen baitio lanari.

2

Proiektuaren kudeaketa plana

2.1. Helburuak

Proiektuaren helburuak definitzerakoan, bi zatitan banatuko ditugu. Batetik, helburu zuzenak, gradu amaierako lana osatzeko beharrezkoak eta, bestetik, zeharkako helburuak, lanak berak zuzenean eskatzen ez dituenak, baina modu berean garrantzitsuak.

- Helburu zuzenak:
 - Mantentze-kudeaketarako web-aplikazioa diseinatu eta inplementatzea
 - Proiektuaren nondik norakoak bilduko dituen memoria osatzea
 - Proiektuaren defentsarako beharrezkoa izango den aurkezpena egitea
- Zeharkako helburuak:
 - Inplementazioan erabilitako lengoia/softwarea ezagutu eta menperatzea
 - Softwarea sortzeko prozesua gertutik ezagutzea
 - Enpresa batean lan-esperientzia hartzea
 - Ikasketa autonomoa eta arazoei aurre egiteko gaitasunak frogan jartzea

2.2. Irismena

Proiektu honen helburua, ikastetxe bateko mantentze-kudeaketarako softwarea garatzea izango da.

Merkatuan hainbat eta hainbat mantentze-kudeaketarako software ditugu eskuragarri. Gehienak, ordea, oso konplexuak dira eta, gainera, produkziara zentratutakoak izaten dira. Horrela, ikastetxe batek izan ditzakeen beharrak aztertuta, oinarritzko funtzioak beteko dituen softwarea garatzea izango da gure helburua, produkzioa alde batera utzita. Honetaz gain, proiektuak berak bete beharko dituen beste hainbat betekizun izango ditugu.

2.2.1 Betekizunak

Esan bezala, gradu amaierako proiektua zuzen ebaluatu dadin, bete beharreko betekizun batzuk izango ditugu. Betekizun horiek, fakultateko orrian izango ditugu eskuragarri, hain zuzen, Gradu Amaierako Lanen araudia[1] fitxategian. Nagusiki, ondorengoak izango dira proiektuak bete beharrekoak:

- Gradu Amaierako Lana defendatu eta ebaluatu ahal izateko, egiaztatu egin beharko da ikasleak ikasketak amaitzeko gehienez ere hiru hautazko ikasgai falta dituela eta hortaz, gradu amaierako lanari dagozkion kredituak izan ezik, graduko titulua lortzeko beharrezkoak diren hautazko 18 kreditu falta dituela kasurik okerrean.
- Informatikaren Ingeniaritzako Graduan, lan hau Gradu Amaierako Lana izeneko derrigorrezko irakasgaiari dagokio, 12 kredituko irakasgaia.
- Ondoko trebetasunak landuko eta egiaztatuko dira lanaren garapenean:
 - Lanean aurreikusitako helburuak erdiestea, hura garatzerakoan agertutako zailtasunak bere kabuz gaindituz.
 - Egin beharreko lana planifikatzea eta kudeatzea.
 - Memoria modu egokian idaztea, eginiko lana zehatz jasotzen duen memoria sortuz. Memoriak ingeniari-txostena bati egokitutako egitura izango du, eta estiloa, berriz, zehatza eta formala izango da.
 - Eginiko lana jendaurrean azaltzea eta defendatzea.
- Matrikulatutako ikasleek ADDI plataforman aurkeztu beharko dute gradu amaierako lanari dagozkion memoria. Memoriaren formatuak fakultateko estandarrari jarraituko dio. Idazkaritzatik zuzendariaren txostena bildu eta memoriarekin batera kopia elektronikoa helaraziko zaio epaimahaikide bakoitzari. Material digital gehigarria beharrezkoa denean (softwarea etab.) memoriari gehituko zaio edo sarean jarriko da atzigarri. Bestelako materialik balego idazkaritzan utziko da.

2.2.2 Irismen mailak

Proiektu honetan, benetako bezeroak ditugu: Tolosaldea institutuko mantentze arloko irakasleak hain zuzen. Horrela, benetako aurreikuspen zuzena egitea zaila izango da, bezeroen eskaerak askotan gehitu eta aldatu egiten baitira. Hortaz, irismen orokorra hainbat mailatan banatu da:

1. Oinarrizko aplikazio funtzionala garatzea: Mantentze-kudeaketarako oinarrizko betekizunak betetzen dituen softwarea osatzea izango da lehen helburua. Horrela, aplikazioak abisuak bidaltzeko balioko du, eta ondoren, abisu hauek ikusi eta modu sinplean osatzeko aukera eskainiko du.
2. Erroreen zuzenketa eta hobekuntzak egitea: Behin oinarrizko aplikazioa funtzional egonda, hau zuzendu eta hobetzea izango da helburua. Hartara, aplikazioak izango dituen erroreak zuzendu, interfazea hobetu eta egokituko da eta kodea txukundu eta komentatuko da.
3. Funtzio osagarriak gehitzea: Behin oinarrizko aplikazioa osatu eta zuzenduta egonda, izan ditzakeen funtzio osagarriak garatuko dira. Adibidez: datuak aplikaziotik zuzenean esportatzea, datu-basean sartu beharrik gabe; mantentze-prebentiborako abisuak sortzea, posta elektronikoarekin lotura ezartzea, *Push notifikazioak* [2] gauzatzea...
4. Interfazea hobetzea: Azkenik, interfazea erabat itxuraldatu genezake, aplikazioari itxura propioa emateko. Horrela, CSS lengoia erabiliz, interfazea eraldatuko dugu, sinplea eta erabilerraza izan dadin.

Esan gabe doa, irismen maila hauen lehentasuna handienetik txikienera doala. Izan ere, azken mailako interfazearen hobetzea egiteko, aurretik beste maila guztiak erabat lortu beharko ditugu, ez baitu zentzurik erroreak dituen aplikazio batean interfazea hobetzen hasteak.

2.2.3 Emangarri nagusiak

Proiektua entregatzeko unean izango dituen emangarri nagusiak ondorengoak izango dira:

- Produktuarekin lotutakoak:
 - Mantentze-kudeaketarako aplikazioa: Lanaren helburu den aplikazioa. Ostatu-emate zerbitzari batetara igoko da, aplikazioa erakutsi eta probatzeko.
 - Aplikazioaren kodea: Kode irekiko filosofiari jarraituz, aplikazioa *Github* [3] plataforman jarriko dugu edozeinentzako eskuragarri, nahi bezala erabili, eraldatu eta zabaltzeko.
 - Aplikazioaren eskuliburua: Aplikazioa erabiltzeko gida-liburutxoa. Hau ere, kodearekin batera, *Github* plataforman eskuragarri egongo da edozeinentzat.
- Proiektuarekin lotutakoak:
 - Gradu amaierako lanaren nondik norakoan memoria. Honek hasierako plangintza, ondorengo jarraipena (erabilitako softwarea, jarraitutako pausoak, garapen datuak...) eta amaieran osatutako dokumentazioa bilduko ditu.
 - Aurkezpena: Proiektuaren defentsan erabiliko den dokumentua.

2.3. LDE Diagrama

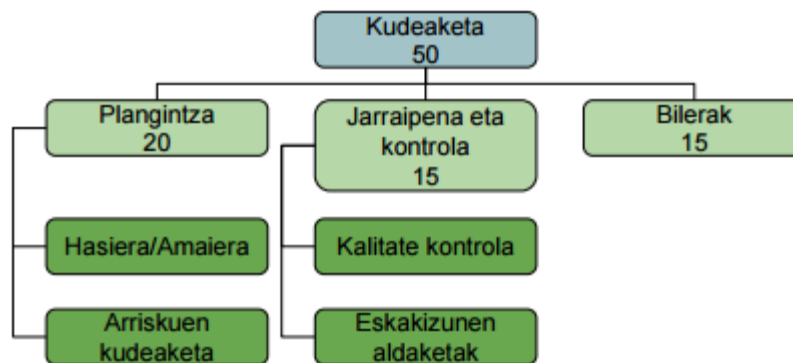
Lanaren deskonposaketa egituraren bidez, gure proiektua hainbat azpiatazetan banatu dezakegu, ondoren azpiataza bakoitza planifikatu eta kosteak estimatzeko. Horrela, hiru ataza nagusitan banatu dugu gure proiektua.

2.3.1 Kudeaketa

Kudeaketan, proiektu osoaren antolaketa eta jarraipena hartuko ditugu kontuan. Proiektu bat hasierakoan, lehenik eta behin, proiektuaren plangintza osatzea ezinbestekoa da. Atzetako bat, hain zuzen ere, plangintza izango da. Horretarako, bilerak egitea ezinbestekoa izango da, bai Tolosaldea Institutuan, bezeroekin, eta baita fakultatean bertan ere, proiektuan esleitutako tutorearekin.

Behin proiektua abian jarrita, jarraipena eta kontrola egin beharko dugu, aurreikuspenekiko aldaketak eta atzerapenak ikusi eta ebazteko.

2.1 irudian, kudeaketa zatiaren LDE diagrama dugu. Bertan, zati nagusi bakoitza agertzen zaigu, eta bertan aurreikusi ditugun ordu kopurua.



1 irudia. Kudeaketari dagokion Gantt diagramaren zatia.

Hain epe luzeko proiektu batean, orduak aurreikustea ez da erraza. Nahiz eta graduan hainbat ikasgaietan plangintzak, orduen aurreikuspenak, jarraipen eta kontrolak landu, *Proiektuen Kudeaketa edo Software Ingeniaritza II* bezalako ikasgaietan, praktika falta somatzen dugu aurreikuspenak egiterakoan. Horrela, beste hainbat proiektu oinarritzat hartuta (beste ikasleen gradu amaierako lanak, graduan zehar egindako hainbat proiektu...), ahal dugun plangintza zuzenena egingo dugu.

- **Plangintza:** Proiektuaren plangintzari 20 ordu eskainiko dizkiogu. Bertan, proiektuaren hasierako eta amaierako data finkatu, proiektua atzetan banatu, eta ataza bakoitzari ordu kopuru bat esleituko diogu. Gutxieneko eskakizun-bilketa ere egingo da, litekeena izanik proiektuaren garapena izatea desbideratze handiena jasango duen arloa.
- **Jarraipena eta kontrola:** Jarraipena eta kontrola behin lanean hasitakoan burutuko dugu. Bertan, planifikazioa zuzen jarraitzen den ala ez ziurtatu behar da, atzerapenik jasan den ala ez eta, ondoren, soluzioak emango dira. Hau lortzeko, Gantt diagraman

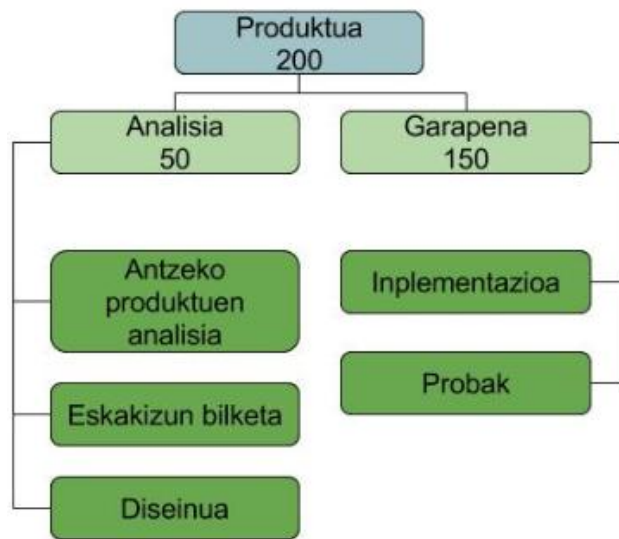
aurreikusitakoarekin, egunero egindako lana eta sartutako ordu kopurua konparatuko ditugu. Jarraipen eta kontrola astean behin egingo da, ostiraletan, astean zehar egindako lanaren jarraipena egiteko. Gantt diagrama gainbegiratu eta sartutako orduak konparatzeaz gain, aurreikusitako atazak bete direla ziurtatuko dugu, atzerapenak garaiz antzemateko. Garaiz ibilita, arriskuak ekiditea eta ezustekoak gutxitzea lortuko baitugu.

- **Bilerak:** Bai proiektua abian jartzeko, baita ondoren jarraipena egiteko ere, bilerak ezinbestekoak izango dira. Bilerak bi arlotan banatuko ditugu. Alde batetik, esleitutako tutorearekin egingo direnak eta, bestetik, Tolosaldea Institutuko irakasleekin, hau da, bezeroekin, egingo direnak. Tutorearekin proiektuaren nondik norakoaz hitz egingo da gehien, eta ez hainbeste garapenaz. Horrela, bezeroekiko lehenengo kontakturako laguntza izango da, proiektua hasteko onespina emango duena eta, noski, hau burutzeko bidean lagunduko gaituena. Bestetik, Tolosaldea Institutuko irakasleak izango ditugu. Hauek proiektuan bezeroaren papera beteko dute. Aplikazioak bete beharko dituen funtzioei buruz hitz egingo da, dudaren bat bada hauei egingo zaie galde, aplikazioaren nondik norakoen inguruan hauekin eztabaidatu beharko baitugu. Beraz, esan liteke tutoreak proiektu osoan gidatuko gaituela, nahiz eta garapenean ere laguntza eskainiko duen, eta bezeroak berriz, aplikazioan bertan, produktuan, zentratuko direla. Bileretan, gutxi-gorabehera, 15 ordu inguru pasako direla aurreikusi da eta ahal bezain azkar eta erakingor izatea komeniko da. Horretarako, bileran jorratuko diren gaiak aurretik ezarri eta lantzea lagungarria izango da.

2.3.2 Produktua

Ohikoa den moduan, produktuaren garapenean pasako dugu denbora gehien. Atal honetan bi ataza nagusi identifikatu ditugu, 2.2 irudian ikusten dugun moduan:

- **Analisisa:** Inplementazioa hasi aurretik egin beharreko lana. Hasteko, bezeroekin bilera egin beharko da. Bertan, softwareak eduki beharreko ezaugarriak bildu beharko dira, alegia, eskakizunen bilketa egin. Behin eskakizunak edukita, antzeko produktuak aurkitu, aztertu eta hauen analisisa egingo da. Azkenik, eskakizunak eta antzeko produktuetatik hartutako ideiak erabiliz, aplikazioaren diseinua egingo dugu eta inplementazioari hasiera emango diogu.
- **Garapena:** Proiektuaren zati nagusia. Proiektuaren helburua Tolosaldea Lanbide Heziketa Instituturako mantetze-kudeaketarako aplikazioa sortzea da, eta, beraz, horri eskainiko diogu denbora gehien. Software garapenari dagokionez, lehenik inplementazioa edukiko dugu. Inplementazioak izango du garrantzi gehien, eskakizunak beteko dituen aplikazioa osatzeak, baina, guztiak ondo funtzionatzen duela frogatzeko, probak egin beharko ditugu. Beraz, garapenean bi zati izango ditugu, inplementazioa eta probak.



2. irudia. Produktuari dagokion LDE diagramaren zatia.

Produktua inplementatzen hasteko, diseinu egokia egitea behar beharrezkoa da. Horrela, inplementazioa modu antolatuan egingo dugu eta, ondorioz, alferrik sortutako kodea, klase bikoiztuak, etab. ekidingo ditugu, hein batean behintzat. Beraz, analisiari 50 orduko aurreikuspena ezarri zaio, inplementatzen hasi aurretik oinarri sendo bat lortzeko. Analisisian bertan sartu dugu diseinua, inplementatzen hasi aurretik egin beharreko lana baita. Bertan, rolak finkatuko ditugu, eskakizunen bilketa zehatza egin, eta bai eraikina eta baita ikastetxeko sailak ere banatu egingo ditugu.

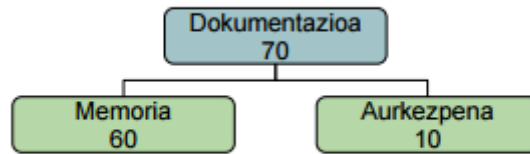
Plangintza egiterakoan, eskakizun-bilketa ez dago oraindik erabat zehaztuta. Produktuak eduki behar dituen ezaugarri nagusiak dagoeneko jakinak dira, baina erabilpen-kasu guztiak ez dira definitu. Beraz, garapenean aurreikusitako ordu kopurua, benetan, asko alda liteke. Horrela ere, 150 orduko dedikazioa aurreikusi da.

2.3.3 Dokumentazioa

Behin produktua inplementatuta, dokumentatzeko garaia heldu da. Izan ere, dokumentazio gabeko aplikazio batek ez baitu ezertarako balio, edo, behintzat, ez du luze iraungo (kodea ulertzeko zailtasunak eguneraketak edo erabilpen gehigarriak egiteko zailtasuna dakar eta, honek ondorioz, bazterketa).

Aplikazioa bera dokumentatzeaz gain, gradu amaierako lanak ere txosten baten beharra du, hasieran egindako planifikazioa, jarraipena eta kontrola, bilerak eta guztia biltzeko. Beraz, dokumentazioari ere ordu dezente eskainiko dizkiogu, berezko garrantziaz gain, epaimahaiak oinarri bezala hartzen baitu txostena, balio gehigarria erantsiz.

Dokumentazioak bi zati nagusi izango ditu; batetik, memoria bera eta, bestetik, aurkezpena edo defentsa (ikus 2.3 irudia).



3.irudia. Dokumentazioari dagokion LDE diagrama.

Lehen esan dugun moduan, memoria txukuna egiteak izugarritzko garrantzia du. Hau lortzeko, 60 ordu eskeiniko diozkiogu txostena burutzeari eta, defentsarako materiala eta aurkezpena bera prestatzeari, 10 ordu.

2.4. Mugarriak

Proiektuaren garapen-prozesuan, lorpen garrantzitsuak lortzen ditugun egunetan, mugarriak finkatuko ditugu. Era honetan, proiektuan aurrera goazela adierazten duten mugak izango ditugu, modu simple batean adierazita. Hau guretzako ez ezik, bezero eta tutorearentzako ere baliagarria izango da, proiektua aurrera doan ziurtasuna izateko. Mugarriak bi zatitan banatuko ditugu; batetik, guk ezarritakoak izango ditugu eta, bestetik, kanpotik ezarritakoak. Azken hauek edukiko dute garrantzi handiena. Izan ere, hemen sartuko ditugu proiektua entregatzeko epea eta proiektuaren defentsa eguna. Beraz, azken mugarri hauek errespetatzea ezinbestekoa izango da.

Guk finkatutakoak ondorengoak izango dira:

- Plangintza bukatzea (2017/03/13): Proiektuaren plangintza bukatu egin beharko dugu egun horretarako.
- Softwarearen analisia bukatu (2017/04/15): Ordurako antzeko produktuen analisia eginda eduki beharko dugu, softwarea diseinatuta eta inplementazioa hasteko prest.
- Inplementazioa bukatu (2017/06/05): Egun honetarako softwarea inplementatuta eta probatuta egongo da.
- Memoria eginda(2017/06/30): Dokumentazio osoa amaituta egon beharko da egun honetarako eta aurkezpena prestatzea besterik ez zaigu geratuko.

Esan bezala, kanpotik ezarritako bi mugarri izango ditugu:

- Proiektua entregatzea (2017/09/03): Egun honetarako proiektuak erabat amaituta egon beharko du eta ADDI plataformara igota [4] [5]
- Proiektuaren defentsa (2017/09/18-21): Hiru egunetan zehar banatuta dago laugarren defentsa epea. Beraz, ordurako proiektuaren aurkezpenak eginda eta landuta egon beharko du.

2.5. Metodologia

Proiektuan zehar lana modu eraginkorrean egingo dela ziurtatzeko, lan metodologia finkatuko dugu. Horrela, metodologiak ordenagailu aurrean alferrik pasatako denbora gutxitzea izango du helburu. Garapena Tolosaldea institutuan bertan gauzatuko da, 9:00etatik 13:00tara arteko lan saioan. Lan egunak ondorengo atalak eduki beharko ditu:

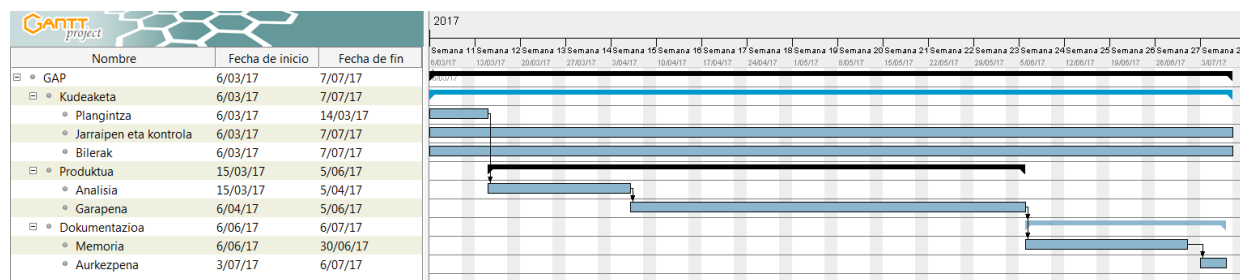
1. Aurreko eguneko lanaren errebaso azkarra: Lanari ekin aurretik aurreko egunear egindako lana errebasatuko da, memoria freskatu eta falta dena identifikatzeko.
2. Eguneko helburuak finkatu: Egunaren bukaeran bukatuta beharko luketen atazak finkatuko ditugu.
3. Garapena: Lanari ekingo diogu.
4. Dokumentazioa: Egindako lana gorde eta dokumentatu egingo dugu, azkenerako utzi gabe. Horrela, bai memoria lantzea, bai aurreko eguneko errebasoa egitea erraztu egingo dugu.
5. Biharamunerako falta dena: Hurrengo egunerako utziko diren lanak identifikatu.

Pauso horiek jarraituz gero, lan saioa optimizatzea lortuko dugu eta, ondorioz, gure portaera txarrak eragin ditzakeen atzerapenak ekiditea.

Honetaz gain, jarraipen eta kontrola atalean adierazi dugun moduan, astean behin, asteko lanaren jarraipena egingo dugu. Hartara, egindako lanaren jarraipen orokorragoa izango dugu, hain zehatza izango ez dena, baina hala ere oso garrantzitsua.

2.6. Gantt diagrama

Jarraian, proiektuaren Gantt diagrama daukagu, proiektuan egin beharko dugun lanaren bilakaeraren aurreikuspena adierazteko balio diguna. Bertan, lanen arteko erlazio eta interdependentziak ikusi ditzakegu, lan bakoitzaren hasiera eta amaiera data finkatzeaz gain. Horrela, proiektuaren iraupen osoa kalkulatu da.



4. irudia. Proiektuari dagokion Gantt diagrama.

2.7. Kalitate-plana

Garatutako produktuak, bere eginkizuna betetzeaz gain, hau da, softwarearen kasuan funtzionatzea eta, memoriaren kasuan egindakoa dokumentatzea, kalitate minimo bat bete

beharko du. Izan ere, ez baitu zentzurik funtzionatzen duen software bat sortzeak, ondoren erabiltzaileentzako interfazea ulertezina bada, edo izugarrizko karga denborak baditu. Bestalde, memoria erabat desordenatuta idazteak edota aurkibiderik ez izateak, ez digu batere lagunduko. Horregatik, kalitate-plan bat ezartzea beharrezkoa izango dugu, minimo bat lortzeko.

2.7.1 Aplikazioaren kalitatea

Aplikazioaren kalitatea bermatzeko hainbat faktore hartuko ditugu kontuan:

- **Arintasuna:** Aplikazioan gabiltzala karga-denborak ahalik eta gehien murrizten saiatuko gara eta, elementuak kargatzen diren heinean, sartu daitezkeen animazio-akatsak saihesten.
- **Interfaze erabilerraza:** Interfazea ahalik eta sinpleena egingo da, gehiegi kargatu gabe, beharrezko elementuak bakarrik erabiliz. Honetaz gain, bai mugikorretan bai ordenagailuetan funtzionatuko duenez, *responsive* web diseinua erabiliko da [6], elementuak gainjartzea, desitxuraketak eta abar ekiditeko.
- **Kode-garbitasuna:** Aplikazioak funtzionatzeaz gain, hau da, kodea zuzena izateaz gain, garbia izatea ezinbestekoa izango da, ondoren sortu daitezkeen arazo-konponketak, aldaketak, hobekuntzak... programatzea errazteko. Horrela, kodea zuzen tabulatuko, izen esanguratsuak erabili eta beharrezko iruzkinak idatziko dira.

2.7.2 Memoriaren kalitatea

Memoriaren gutxieneko kalitatea bermatzeko ere, hainbat faktore hartuko ditugu kontuan:

- **Luzera:** Nahiz luzerak ez duen zuzenean kalitatea bermatuko, tamaina honetako proiektu batean, gauza asko lortuko dira. Beraz, memoriak gutxieneko neurria izatea ezinbestekoa izango da.
- **Zuzentasuna:** Memoria idazteko garaian, EHUko GAP araudiak esaten duena beteko da (2. artikulua, 3 puntuko 3. araua)[1][7]: “Memoria bat modu egokian idaztea, eginiko lana zehatz jasotzen duen memoria bat sortuz. Memoriak ingeniari-tza-proiektu bati egokitutako egitura izango du, eta estiloa, berriz, zehatza eta formala izango da”. Akatsak zuzentzeko berriz, Xuxen 5 euskarazko zuzentzaile ortografiko eta gramatikala erabiliko da. Terminologiari dagokionez, Elhuyar Hiztegia eta Euskalterm terminologia-banku publikoa erabiliko dira, batez ere.

2.8. Arrisku-plana

Proiektuaren garapenean aurrera goazen heinean, hainbat arazo sor daitezke, bai ezustekoak gertatu zaizkigulako, edo baita guk arduragabekeriaz jokatu dugulako ere. Horregatik, proiektuan sor daitezken arriskuak identifikatu ditugu, aurreikuspenen bidez, sor daitezken arazoak ekidin eta gertatuz gero, soluzioa emateko.

- **Plangintza okerra egitea:** Proiektu guztien oinarria plangintza sendoa osatzean datza. Izan ere, orduen estimazio kaxkarra egiten badugu, betebeharrak kontuan hartu gabe, ondoren arazoak izango ditugu epe mugak betetzean. Hau gerta ez dadin, lehenik eta behin orduen gutxi-gorabeherako bat osatuko da, proiektuaren garapen osorako. Hemen ordea, zati handi bat garapenarena izango da, eta eskainiko dizkiogun orduen estimazio zuzen bat egiteko asmoz, bezeroekin bilera egingo dugu, softwareak bete beharko dituen eskakizunak bildu, eta horren arabera plangintza osatzeko. Horrela, estimazio zorrotzago bat osatuko dugu, ondoren garapenean lagunduko diguna.
- **Plangintza ez jarraitzea:** Plangintza txukun eta landu bat egiteak ez du ezertarako balioko, berau jarraitzen ez bada. Horregatik, ahal den heinean plangintza jarraituko da, atzerapenak eta ezustekoak ekiditeko.
- **Eskakizunen aldaketa:** Bezeroek hasiera batean eskakizun batzuk eska ditzakete, baina, ondoren, baita iritiz aldatu eta eskakizun batzuk aldatu, gehitu edo ezabatu ere. Hau, neurri handi batean, ez da gure esku egongo, baina, zenbait neurri hartu ditzakegu sortu daitezken atzerapenak handiak ez izateko. Eskakizun bilketa egiten den egunerako bezeroei jada eskakizunak prestatuta edukitzea eskatzea egokia izan liteke. Horrela, aurrez landu dela ziurtatuko dugu, eta baita hainbat buelta eman zaizkiola ere. Hartara, etorkizunean egon daitezken aldaketak murriztuko dira. Gainera, orduen estimazioan badaezpada ordu gehiago sartuko dira produktuaren garapenean, aldaketak baleude atzerapenak saihesteko.
- **Aplikazioaren garapenean arazoak:** Software-garapenean ohikoak dira errore txikiak sortutako atzerapenak. Hauek, askotan, ez dira oso zailak izaten konpontzen, baina arazoa zein den konturatu arte orduak edo egunak pasa daitezke. Horretaz gain, teknologia berriak erabiliko ditugunez, hauen ikasketa-prozesua ere kontuan hartu beharko da. Horrela, software-garapen denbora luzatu egingo dugu, ezustekoak ekiditeko.
- **Aplikazioaren bateragarritasuna:** Web-aplikazio bat garatu behar dugunez, bai mugikorretarako, bai ordenagailuetarako, garrantzitsua izango da interneteko nabigatzaile ezberdinekin bateragarria izatea. Horregatik, ahal den moduan bateragarriak diren teknologiak erabiliko ditugu.
- **Datuen galera:** Tamaina honetako proiektu batean ezinbestekoa izango dugu badaezpadako babes-kopiak sortzea, ordenagailua, zerbitzaria... arazoak ematen hasten badira, bigarren aukera izan dezagun. Horrela, txostena *Driven* eta ordenagailuan bertan gordeko dugu eta aplikazioa, berriz, *Github* plataformara eta *ostatu-emate* zerbitzura igoko ditugu, ordenagailuan edukitzeaz gain. Babes-kopia hauek eguneko lana bukatzean egingo dira, egunero.
- **Erabilitako software-elementuen eguneratzeak:** Software bat garatzeko hainbat teknologia eta liburutegi erabiltzen dira. Hauetan ohikoak izaten dira eguneraketak, hobekuntzak gehitu eta erroreak konpontzeko. Orokorrean, aldaketak txikiak izaten dira eta ez da ezer aldatu behar izaten, garatzaileak eguneraketa bateragarriak egiten saiatzen baitira. Hala ere, batzuetan, aplikazioak zuzen ez funtzionatzea ekarri lezake, liburutegiaren oinarria aldatu, edo erabiltzeko modua aldatzen bada. Beraz, garrantzitsua izango da erabiltzen ditugun teknologiak gertutik jarraitzea, horrelako eguneraketak aurreikusi, eta arazorik badago ahal den azkarren soluzioa emateko.

2.9. Interesatuak

Proiektu honek, ohikoak diren interesatuez gain, hau da, proiektua garatuko duena (Alexander Aranburu Peñagarikano) eta GAPeko zuzendariak (Ana Arruarte) gain, enpresa batean egingo denez, enpresako langileak ere izango ditu kontuan. Gure kasuan, enpresa Tolosaldea Institutua da, eta bertako bezeroak Mantentze Alorreko bost irakasleak izango dira, hain zuzen ere mantentze-kudeaketarako aplikazioa sortzea baita helburu. Lehenago esan bezala, azken hauek beteko dute aplikazioaren bezero rola. Azkenik, proiektua ebaluatuko duten epaimahaikideak izango ditugu, gure kasuan software-espezialitatekoak.

2.9.1 Komunikazioa

Proiektua enpresan egingo denez, bi atal nagusitan banatuko dugu komunikazio-plana: batetik aplikazioaren bezeroekin izango duguna eta, bestetik, proiektuko zuzendariarekin izango duguna.

- **Zuzendaria:** Proiektuaren zuzendariarekin komunikatzeko lehentasuna aurrez aurreko bilerek izango dute. Bilerak hauek, aurrez, posta elektronikoko bidez finkatuko dira, baina ondoren fakultatean bertan, zuzendariaren bulegoan, bilduko gara. Nahiz lehentasuna honek eduki, duda edo arazo puntualen bat sortuz gero, posta elektronikoko bidez komunikatzea ere aurreikusten da.
- **Aplikazioaren bezeroa:** Aplikazioaren garapena Tolosaldea Institutuan bertan egingo da. Beraz, bilerak zehazteko, irakasleekin hitz egingo da zuzenean data finkatzeko. Horretaz gain, duda puntualen bat sortuz gero, momentuan libre dagoen irakasleari galdetuko diogu, besterik gabe. Azkenik, Google Drive plataforman karpeta konpartitu bat egingo ba, beharrezkoa edo interesgarria izan daiteken informazioa trukatzeko.
- **Epaimahaikideak:** Hauekin ez da komunikazio plan berezirik izango, proiektuaren defentsa egunera arte ez baita elkar komunikatu beharrik izango.

3

Aurrekariak eta erabilitako teknologia

Kapitulu honetan, garatu beharreko aplikazioa osatzeko aztertutako antzeko aplikazioak eta hautatutako teknologien inguruko informazioa bildu dugu. Ondoren, garapenean erabilitako teknologiak eta hauek hautatzeko zergatia azalduko ditugu.

3.1. Aurreikusitako teknologiak

Aplikazioa hasieratik garatzea erabaki dugunez, hainbat lengoia eta teknologia erabili beharko ditugu. Ondoren, erabiliko diren programazio-lengoia eta liburutegiak azalduko ditugu, eta baita garapenean eta ondoren aplikazioa martxan jartzeko erabilitako beste hainbat teknologia ere. Horrela, hiru atal nagusitan banatu ditugu teknologiak:

3.1.1 Aplikazioa

3.1.1.1 Nabigatzaileak

Smartphone zein ordenagailuetan funtzionatu dezakeen aplikazio bat sortu nahi da. Beraz, helburua, ahalik eta nabigatzaile gehienetan funtzionatzen duen aplikazioa sortzea izango da. Nabigatzaile guztietan probak egitea, ordea, ezinezkoa zaigunez, 4 nabigatzaile nagusitan funtzionatzeko helburua jarri dugu. Horrela, Mozilla Firefox [8], Google Chrome [9], Internet Explorer/Edge [10] eta Safari [11] nabigatzaileetan funtzionatuko duela bermatuko da, hain zuzen ere apirilean StatCounter eta Wikimedia orrialdeek bildutako datuen arabera erabilienak diren nabigatzaileak (ikus 3.1 irudia) [12].

Usage share of all browsers for April 2017

Source	Chrome	Safari	UC	Firefox	IE	Opera	Android	Edge	Others
StatCounter	53.69%	14.64%	8.41%	6.31%	3.91%	3.57%	2.65%	1.71%	5.14%
Wikimedia	45.7%	21.4%	0.6%	7.1%	9.8%	1.3%	1.4%	2%	10.7%

5. irudia. 2017ko apirilean bildutako nabigatzaileen erabilera datuak.

Irudiko datuak aztertzen baditugu, ikusiko dugu lau nabigatzaile horietan bateragarria izatea lortzen badugu, merkatuaren % 80arentzako funtzionamendua bermatuta egongo dela. Hau,

noski, ez da erabat zuzena izango. Izan ere, nahiz eta nabigatzaile horietako bat erabili, bertsio oso zahar edo eguneratu gabea erabiltzen bada, funtzionamendua ezingo baita erabat bermatu. Probak egiteko, gainera, Mozilla, Chrome eta IE garapenean erabili den ordenagailu bertan instalatu dira. Safari nabigatzaileak arazo gehiago sortu dizkigu, macOS eta iOS sistema eragileetarako bakarrik baitago. Hala ere, bezeroetako bik iPhone smartphoena dutela aprobe txatu dugu, bertan noizean behin probak egin eta erroreak konpontzeko.

3.1.1.2 Programazio-lengoiak

Web-aplikazioa garatzeko, arlo honetan nagusi diren hiru programazio-lengoiak erabili dira; HTML5, PHP eta Javascript.

3.1.1.2.1 HTML5[13]

HTML5 aski ezaguna den HTMLren bostgarren bertsioa da. HTML etiketatan oinarritzen den lengoia da, web-orriak sortzeko egina. Honen bidez, web-orrialdearen oinarritzko egitura definitzen da eta kodea idatziz edukia definitzen da, hala nola, testua, irudiak, bideoak eta abar.

3.1.1.2.2 PHP[14]

PHP zerbitzari -aldeko programazio-lengoaia da, eduki dinamikoa zuen web-garapenerako sortua. Gaur egun, PHP lengoiak hainbat hobekuntza jasan ditu eta orain komando bidezko interfazea du, aplikazio grafikoetan erabili daitekeena. Horretaz gain, web-zerbitzari eta sistema-eragile/plataforma gehienetan erabili daiteke, inolako kosturik gabe.

3.1.1.2.3 Javascript[15]

Javascript Java programazio-lengoiarekin antzekotasun handia duen, baina hau ez bezala web-orrien garapenerako zuzenduta dagoen, programazio-lengoaia da. Normalean, erabiltzaile-aldean erabiltzen da eta erabiltzaile-interfazean hobekuntzak eta web-orri dinamikoak sortzeko balio du. Nahiz eta, esan bezala, erabiltzaile-aldean erabili, zerbitzari-aldean ere erabili daiteke (Server-side Javascript) eta webetik kanpoko aplikazioetan ere, adibidez, PDF dokumentuetan nahiko erabilia da. Gaur egun, nabigatzaile moderno guztiek interpretatzen dute Javascript, inongo plugin beharrik gabe.

3.1.1.2.4 CSS[16]

CSS diseinu grafikorako lengoia da, dokumentu egituratu baten aurkezpena definitu eta deskribatzeko erabiltzen den markaketa-lengoaia. Nahiz eta gehienetan HTML eta XHTML web-orrien itxura ezartzeko erabiltzen den, beste hainbat dokumenturi ere aplikatu dakiok, hala nola, XML, SVG edo XUL motako dokumentuei. HTML eta Javascript-ekin batera, CSS teknologia web-orri askok erabiltzen duten teknologia da, ikusgarriak diren orrialdeak sortzeko, web-aplikazioentzako erabiltzaile-interfazeak sortzeko edo mugikorrentzako interfaze-grafikoak sortzeko (Firefox OS esaterako).

3.1.1.3 Liburutegiak

Programazio lengoiaez gain, askotan laguntza handia eman diezagukete liburutegiek. Ondoren, erabilitako liburutegiak ditugu:

3.1.1.3.1 JQuery Mobile[17]

Jquery Mobile ukipenerako optimizatuta dagoen web-framework-a da, zehazki esateko *Javascript* lengoaiaren liburutegia. *jQuery* framework-aren gainean dago eraikia. Hau HTML dokumentuen interakzioa, DOM elementuak aukeratzea, event-en tratamendua, animazioak sortzea eta *AJAX* aplikazioen garapena errazteko sortu zen. Horrela, *jQuery Mobile* azken urteetan izugarriko gorakada jasan duten smartphone eta tabletekin erabat bateragarria izateko helburuarekin sortu zen. Bateragarritasun horren adibide, ondorengo plataformak ditugu: *Android*, *iOS*, *Windows Phone*, *Blackberry*, *WebOS* eta *Symbian*. Hau da, mahaigaineko eta smartphonetan erabiltzen diren nabigatzaile nagusiak.

3.1.2 Datu-basea

Web-aplikazioan erabili eta sortuko ditugun datuak nonbait gorde beharko ditugu. Horretarako, datu-base bat beharko dugu.

3.1.2.1 MySQL[18]

MySQL datu-baseak kudeatzeko sistema da. Oracle eta Microsoft SQL Server-ekin batera, web-garapenerako datu-base erabiliena da. MySQL datu-basea oso azkarra da irakurketak egiten, baina, konkurrenteki aldaketa asko daudenean, integritate edo osotasun arazoak sor daitezke. Hala ere, web-aplikazioetan konkurrentzia gutxi izaten da aldaketetan eta, irakurketak berriz, oso maiz egiten dira. Beraz, gure kasurako, MySQL datu-basea ideala izango da.

3.1.3 Zerbitzaria

Web-aplikazioa eskuragarri eduki nahi badugu, ezinbestekoa izango zaigu zerbitzari batean martxan jartzea. Gure kasuan, aplikazioa momentu puntualean atzitu beharko denez, hau da, proiektuaren defentsa egitean bakarrik beharko denez (enpresak berak martxan jartzeko ulean bere zerbitzariak erabiliko ditu), ostatu-emate-sistema doakoa izatea erabaki da, erabiliko ez diren gehigarriak ez ordaintzearen. Garapenaren prozesu ezberdinetarako, hiru zerbitzari ezberdin erabiliko dira, momentuan ditugun beharrak asetzeko.

3.1.3.1 Garapenean erabilitako ordenagailua

Softwarea garatzen goazen heinean, web-aplikazioa lokalki exekutatzea nahikoa izango dugu, ez baitugu edozeinentzako eskuragarri jartzeko beharrik izango, aplikazioa osorik garatu arte. Horretarako, *XAMPP*[19]plataforma erabiliko dugu. *XAMPP*ek datu-baseen kudeaketarako *MySQL* sistema eskaintzen digu, Apache zerbitzaria eta *PHP* eta *Perl* scrip-lengoaien interpretatzailea. Modu horretan, *XAMPP* instalatu ondoren beharrezko guztia gure ordenagailuan izango dugu, garapenari ekiteko prest. *Apache* HTTP web-zerbitzaria da, kode irekikoa, *Unix*, *Windows* eta *Macintosh* sistema-eragileetan funtzionatzen duena, beste batzuen artean.

3.1.3.2 Hostinger[20]

Behin aplikazioa garatuta, ostatu-emate zerbitzari batera igo beharko dugu, defentsaren egunean edozein tokitatik eskuragarri edukitzeko. Interneten hainbat doako ostatu-emate zerbitzu daude eskuragarri, baina horietatik *Hostinger* aukeratu da.

Hau hautatzea, zerbitzari honekin aurretik izandako esperientzia onagatik erabaki dugu. Izan ere, *Hostingerrek* doako plana eskaintzen digu, inongo publizitaterik gabe. Horretaz gain, *MySQL* datu-basea eta *PHP* interpretatzailea ditu. Doakoa denez, zenbait muga ere baditu, adibidez, gehienez bi datu-base eduki daitezke, datuentzako 2GBeko muga daukagu, eta domeinuaren izena, aukeratu daitekeen arren, *Hostingerrek* .esy gehitzen dio, domeinuaren itxura narrastuz. Hala ere, momentu puntual batean web-aplikazioa eskuragarri edukitzeko denez, gure beharretarako oso egokia da.

3.1.3.3 Enpresako zerbitzaria

Lehenago esan bezala, enpresan martxan jartzea erabakitzen bada, bertako zerbitzaria erabiliko da. Ikastetxeko sare-kudeatzaileekin hitz egin ostean, XAMPP plataforma instalatzeko arazorik ez litzatekeela egongo ziurtatu digute, ondorioz gure garapen lokaletik zerbitzarira pasatzea ez litzateke zaila izango.

3.1.4 Bestelakoak

Aplikazioa garatzeko softwarea eta ondoren eskuragarri jartzeko erabilitako teknologiez gain, beste hainbat software ere erabili dugu. Batetik, aplikazioa bera idazteko softwarea edukiko dugu eta, bestetik, bertsio kontrola kudeatzeko softwarea.

3.1.4.1 Notepad++[21]

Notepad++ testu eta iturburu kode editorea da. Hasiera batean “*Bloc de Notas*” simple baten antza izan dezake, baina web-aplikazioen garapenerako oso erabilgarri bihurtu daiteke, hainbat gehigarriekin. Horien bidez, *HTML* etiketak automatikoki ixteko aukera izango dugu, lerro eta zutabe zenbakia ikusi, bi fitxategi konparatzeko aukera, “*autocompletar*” funtzioa, etiketak non hasi eta bukatzen diren ikusteko aukera, kolore bidezko elementuen bereizketa... Guztiarekin, web-aplikazioa garatzeko tresna oso erabilgarria bihurtzen da.

3.2.4.2 Git[22]

Git bertsio-kontrol software bat da; hain zuzen, produktu edo konfigurazio batean egin daitezkeen aldaketak kudeatzeko programa. Iturburu-kode askoko aplikazioen bertsioen mantentzea egiteko diseinatu zen, eraginkortasunean eta fidagarritasunean pentsatuz. Horretaz gain, kodearen adar ezberdinak izatea ahalbidetzen du, garapen ez lineala bultzatuz. Horrela, software baten garapenean hainbat programatzaile izatea ahalbidetzen du, bakoitzak adar bat erabiliz funtzionaltasun ezberdinak sortzeko aukera izan dezaten. Programatzaile bakoitzari kopia lokal bat sortzen dio, garapenaren historia osoarekin. Aldaketak adarretan banatzen dira, eta adarrak modu lokalean gordetzen; adar lokaleko aldaketak ere urruneko zerbitzarietan erregistratzen dira. Horrela, adar hauetako aldaketak adar nagusiarekin fusionatu daitezke, funtzionalitateak gehitu edo erroreak zuzentzeko.

Erabiltzaile bakoitzak nahi bezala erabil dezake software hau, hala ere, erabilera egokia honako hau da (ikus 3.2 irudia):

Master:

Adar nagusia da. Erabiltzailearen esku dagoen programa baten gordailuari deritzo. Beraz, bertsio egonkor bat izatea beharrezkoa izango da.

Development:

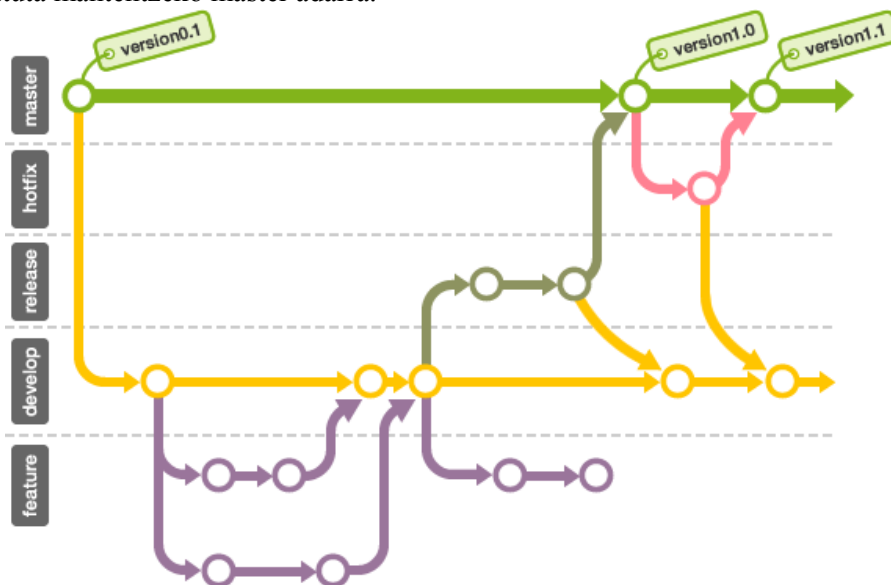
Adar nagusitik ateratako bigarren bat da. Integrazio adar bat da, funtzionaltasun berri guztiak bertan integratuko dira.

Features:

Funtzionaltasun berri bakoitzarekiko adar espezifiko bat sortzen da. Hauek *development* adar batetik garatutakoak izango dira. Behin funtzionaltasuna garatuta, bestearekin integratzeko *development* adarrarekiko merge bat egingo da, hau da, bateratu egingo dira..

Hotfix:

Hotfix-ak programa garatzerako orduan sortzen diren bug-ak edo arazoak, masterretik ateratako adarrak dira. Horiek lehenbailehen argitaratu eta konpondu beharko dira, sor ditzaketen arazoengatik. Behin arazoa konponduta, master adarrarekin bateratu beharko da, eguneratuta mantentzeko master adarra.



6.irudia. Git-en erabilera zuzena azaltzen duen irudia.

4

Analisia eta diseinua

Atal honetan, softwarea garatzen hasi aurretik egindako azterketa eta diseinua azaltzen dira. Lehenik, bezeroekin erabakitako eskakizun-bilketa azalduko da, funtzionalitate bakoitza azalduz eta, ondoren, betebeharrak osatzeko garatuko den aplikazioaren diseinua.

4.1 Analisia

Proiektuaren oinarria izango da aplikazioa eta, ondorioz, garatu aurretik analisi sakon bat egitea ezinbestekoa izango da, ondoren ezusteko eta atzerapenak ekiditeko. Gainera, kasu honetan, mantentze-kudeaketa ez dugu gertutik ezagutzen eta, beraz, ezezaguna egiten zaigu erabat bertako funtzionamendua. Horregatik, eskakizun-bilketarekin hasi aurretik, bi txango egin dira, mantentze-kudeaketa gertuagotik ezagutzeko.

Lehen pausoa, ikastetxeak berak dituen tailerrak ezagutzea izan da. Irakasle pare baten laguntzaz, ikastetxeko tailer ezberdinak ikusi eta bertan nola funtzionatzen duten azaldu zaigu, ideia bat hartzeko. Horretaz gain, Leitzako paper-fabrika batera ere bisita egin dugu, bertako mantentze-lanen kudeaketa gertutik ezagutzeko. Nahiz eta azken hau ikastetxeko beharretatik pixka bat urrundu, interesgarria izan zaigu bisita, “benetako” enpresa baten mantentze-kudeaketa ezagutzeko.

4.1.1 Eskakizun-bilketa

Behin mantentze-kudeaketa ezagutzen delarik, eskakizun-bilketari ekin diogu. Horretarako, Tolosaldeako mantentze alorreko lau irakaslerekin bildu gara. Bileran, aplikazioak eduki beharko lituzkeen funtzionalitateez galdetu zaie, inongo mugarik jarri gabe, “*brainstorming*”[23]teknikaren antzekoa simulatuz, botatako ideia guztiak apuntatzen joan gara. Normala denez, hainbat funtzionalitate gure irismenetik kanpo geratu dira; denbora, diru edo garatzailez oso mugatuak baikaude. Hala ere, hainbat ideia interesgarri eta egingarri lortu ditugunez, lortutako ideiak hiru mailatan banatu ditugu:

- **Oinarrizko funtzionalitateak:** Aplikazioak beteko dituen funtzionalitate minimoak izango lirateke.
 - Abisuak bidaltzeko aukera:
 - Bidalitako abisuak ikusteko aukera:
 - Esleitutako abisuak ikusteko aukera:
 - Egindako lanaren inguruko informazioa gorde
 - Erabiltzaileak sortu editatu eta ezabatzeko aukera

- **Funtzionalitate gehigarriak:** Behin oinarrizko funtzionalitateak osatuta, denbora izanez gero gehituko direnak.
 - Abisuak editatzeko aukera
 - Datuak esportatzeko aukera
 - Lan prebentiboak

- **Irismenetik kanpo daudenak:**
 - Posta elektronikoarekin lotura
 - Mugikorretarako aplikazio “natiboa”
 - Push notifikazioak
 - Datuen grafikoak
 - Material/makinak datu-basean txertatu

4.2 Aztertutako aplikazioa

Garatu nahi dugun aplikazioa ez da erabat berritzailea izango. Sarean, hainbat aplikazio ditugu mantentze-kudeaketa eramateko balio dutenak. Hauek, oso erabilgarri izango zaizkigu gure aplikazioa sortzeko unean, beraien abantaila eta desabantailak identifikatuz, aplikazioak nolakoa izan behar duen ideia bat egiteko. Nahiz eta hainbat aplikazio egon, aplikazioaren bezeroak ez daude gustura probatutako aplikazioekin. Izan ere, aplikazio hauek oso garestiak dira eta aukera gehiegi eskaintzen dituzte, beraien beharretarako. Hortaz, aplikazio hauek aztertuko ditugu, indarrak eta ahulguneak identifikatzeko, baina beti ere bezeroen iritzia kontuan hartuta.

Ondoren, aztertutako bi aplikazio nagusiak azalduko ditugu, bakoitzaren abantaila eta ahuleziekin.

4.2.1 Mantis[24]

4.1 irudian ikusi dezakegu gaur egun erabiltzen duten Mantis-en menu nagusia. Bertan intzidentziak kategoriaren arabera banatzen dira, hala nola: esleituak, bidaliak, berriki editatuak, ebatziak... Modu hau interesgarria izan daiteke gure aplikaziorako, intzidentziak modu garbi batean ikusten baitira, kolore eta guzti.

Bestalde, 4.2 irudian abisua bidaltzeko leihoa ikusi dezakegu. Kasu horretan, nahiko egokia iruditzen zaie abisua bidaltzeko modua baina, berriz ere, aukera batzuk soberan ikusten dituzte, zehazki: Gertaera, Zorroztasuna eta Profila.

Azkenik, 4.3 irudia daukagu, intzidentziaren informazioa jasotzen duena. Bertan, berriz ere, aurreko ataleko informazio bera soberan daukagu, etiketak ezartzeko aukerarekin batera.

Mantis aplikazioak badauka, ordea, arazo nagusi bat. Aplikazioa, berez, ez da mantentze-kudeaketa egiteko sortua, baizik eta abisuak bidali eta kudeatzeko sortua da. Horregatik, bertan ezin dira egindako lanorduak sartu, eta ezta mantentze prebentiborako abisuak sortu ere. Beraz, nahiz eta nagusiki abisuak bidaltzeko egokia izan, beste hainbat behar ez ditu asetzen aplikazioak.

Nagusia | My View | Intzidentziak ikusi | Intzidentzia jakinarazi | Txosten Orokorra | Nire kontua | Irten

Issue # Intzidentziara Jauzi

Recently Visited: 0000655, 0000636

berria | egiaztatua | esleitu | ebatzia | itxia

Issue Status Percentage

8% 92%

Assigned to Me (Unresolved) [^] (1 - 4 / 4)

- 0000655 Frogak egiten ari naiz berokuntza-hotza atala - 2017-04-05 09:31
- 0000636 bigarren solairuko erradiadorea berokuntza-hotza atala - 2017-03-08 08:44
- 0000581 Mekanika laugarren tailerlean aerotermo izorratua berokuntza-hotza atala - 2017-01-10 09:53
- 0000529 A14 ETA A13 berokuntza-hotza atala - 2016-11-03 09:19

Unassigned [^] (0 - 0 / 0)

Reported by Me [^] (1 - 1 / 1)

- 0000655 Frogak egiten ari naiz berokuntza-hotza atala - 2017-04-05 09:31

Resolved [^] (0 - 0 / 0)

Recently Modified [^] (1 - 10 / 51)

- 0000655 Frogak egiten ari naiz berokuntza-hotza atala - 2017-04-05 09:31
- 0000597 E21 gelan erradiadore balbula izorratuta berokuntza-hotza atala - 2017-04-03 13:50
- 0000646 Frontniko konpresoreak ez du eniten

Monitored by Me [^] (0 - 0 / 0)

7. irudia. Mantis-en leiho nagusia, abisuak kategoriatan banatuta.

Nagusia | My View | Intzidentziak ikusi | Intzidentzia jakinarazi | Txosten Orokorra | Nire kontua | Irten

Issue # Intzidentziara Jauzi

Recently Visited: 0000655, 0000636

Txostenaren zehaztapenak sartu.

*Atala (select)

Gertaera beti

Zorroztasuna erdikoa

Lehentasuna normal

Profila aukeratu

Edo eremu hauek bete

Plataforma

Sistema Eragilea

Bertsioa

Honi esleitu

*Laburpena

*Deskripzioa

8. irudia. Mantisen abisua bidaltzeko leihoa.

View Issue Details [[Oharretara Jauzi](#)] [[Oroit-oharra bidali](#)] [>>] [[Intzidentzia historia](#)] [[Inprimatu](#)]

ID	Saila	Atala	Sarbidea	Bidalketa-data	Azken Eguneraketa
0000655	05 - Berokuntza-Hotza Mant.	berokuntza-hotza atala	publikoa	2017-04-05 09:31	2017-04-05 09:31
Informatzailea	aizkierdo				
Honi esleitua	aizkierdo				
Lehetasuna	normal	Zorroztasuna	erdikoa	Gertaera	beti
Egoera	esleitua	Ebazpen-informazioa	intzidentzia irekia		
Plataforma		Sistema Eragilea		Bertsioa	
Laburpena	0000655: Frogak egiten ari naiz				
Deskripzioa	manti				
Tags	No tags attached.				
Attach Tags	(Separate by ",") <input type="text"/> Existing tags <input type="button" value="Attach"/>				
Denbora (h)					

9. irudia. Mantis-en abisuaren inguruko datuak ikusteko leihoa.

4.2.2 Veedor[25]

Aplikazio hau, *Mantis* ez bezala, mantentze-kudeaketara dago zuzendua. Honek, ordea, beste sistema gehienek duten arazo bera du, konplexutasuna. Sistema, web-orrian dioten moduan, makina-erreminta industrialen mantentze lanak egiteko dago pentsatuta, produkzioa ere kontuan hartuz. Ikastetxe batek, ordea, makina kantitate handirik ez du izango eta, gainera, ez du produkziarik sortuko. Beraz, *Veedor* aplikazioak eskaintzen dituen ezaugarrien zati handi bat (exekuzio denborak, beharrezko erremintak, sortutako soberakinak, makina ezberdinen datu basea...) ez dira beharrezkoak ikastetxe batean. Honetaz gain, aplikazioa erabiltzeko harpidetu egin behar da eta, gutxieneko tarifa, 30€/hileko da. Hortaz, guztia kontuan hartzen badugu, aplikazioa gure beharretarako ez dela egokia konturatuko gara.

4.3 Diseinua

Behin analisia egin ostean, aplikazioko aktoreak eta erabilpen-kasuak identifikatu dira, eskatutako funtzionalitateetatik abiatuta. 4.4 irudian erabilpen-kasuen diagrama ikusi ditzakegu, erabilpen-kasuak eta hauek burutuko dituzten aktoreak.

4.3.1 Rolak

Lehenik eta behin, hiru aktore ezberdin definitu ditugu: irakaslea, arduraduna eta administratzailea. Erabiltzaile gehienak, irakasle moduan sartuko dira. Irakaslea aktoreak, abisuak bidali eta bidalitako abisuen egoera ikusteko aukera edukiko du, besterik ez. Beraz, esan genezake, erabiltzaile arruntena eta sinpleena izango dela.

Alor bakoitzak, dagokion mantentze-arduradun bat edo gehiago dauzka. Horiei, arduraduna moduan sartzeko aukera emango zaie. Horrela, arduradunek, abisuak bidaltzeaz gain, berauei esleitutako abisuak ikusi eta egindako lana osatzeko aukera edukiko dute. Modu horretan,

arduradun bakoitzak egindako lanorduak gordetzeko aukera edukiko dugu, alor bakoitzean denbora epe batean sartutako ordu kopurua. Hori, etorkizunerako interesgarria izan daiteke, irakasle edo arduradun bakoitzari kurtsuan zehar esleitu behar zaizkion orduen estimazioa egiteko, adibidez.

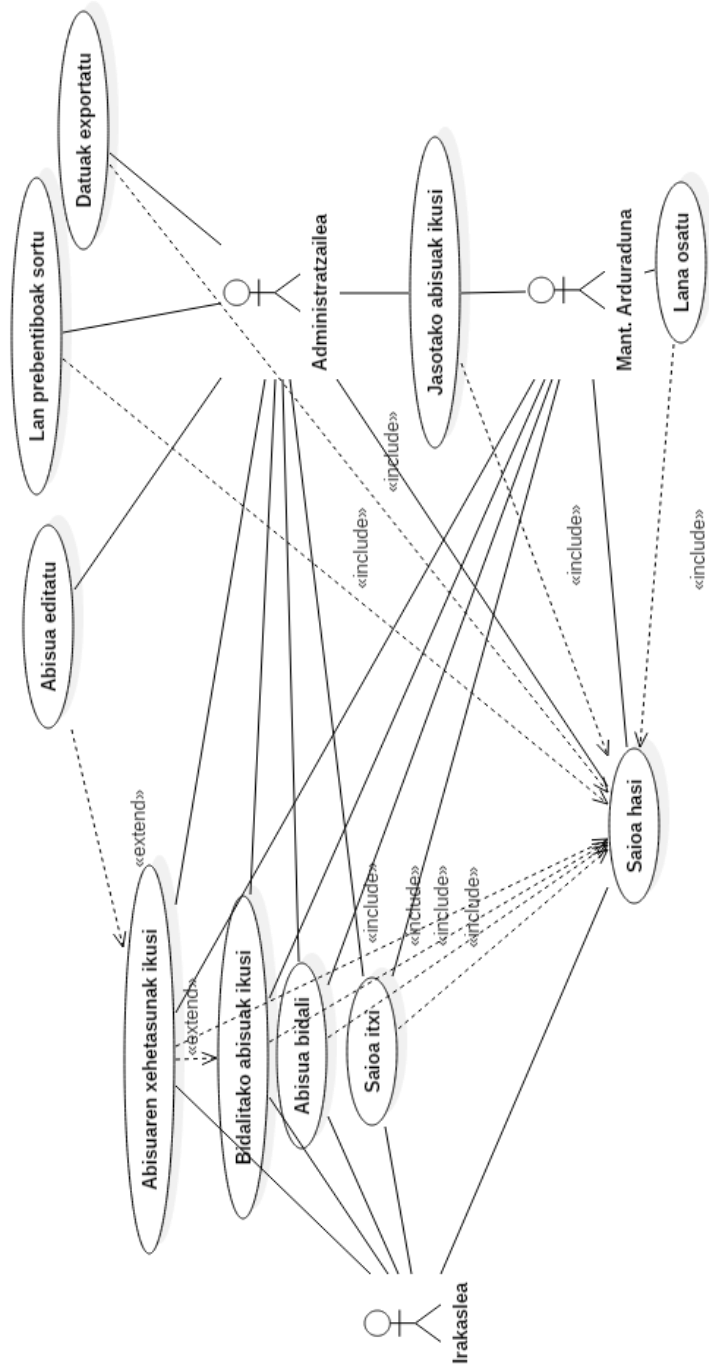
Azkenik, administratzailea izango dugu. Egun, ikastetxean, arduradun nagusi bat daukate, Mantis programan oker bidalitako abisuak, abisu orokorrak...kudeatzen dituena. Beraz, horri ere beste aktore bat esleitu diogu, aplikazioaren kudeaketa egiteko hain zuzen.

4.3.2 Erabilpen kasuak

Beraz, esan bezala, 3 aktore ezberdin sortu ditugu. Erabiltzaile arruntentzat, irakaslea aktorea deitu diogunak, ondorengo erabilpen kasuak edukiko ditu eskuragarri: Aplikazioa sartu eta irteteko saioa hasi eta ixteko aukera izango du. Honetaz gain, abisuak bidaltzeko aukera edukiko du eta, noski, abisu hauen egoera ikusteko aukera.

Abisu horiek aztertu eta konpontzeko ordea arduradunak beharko ditugu. Horiek, Irakasleak dituen aukerez gain, jasotako abisuak ikusteko aukera edukiko dute eta, ondoren, behin lana eginda, egindako lanaren informazioa osatzeko aukera.

Azkenik, administratzailea aktoreak, beste aktoreen aukerak edukitzeaz gain, beste hainbat aukera ere edukiko ditu. Adibidez, aplikazioko datuak exportatzeko aukera izango du, abisu prebentiboak sortzeko aukera, bidalitako abisuak editatu edota erabiltzaileak sortu, editatu eta ezabatzeko aukera. Azken honek izango du aplikazio osoaren kontrola emateko aukera. Beraz, komenigarria izango da aplikazioan administratzaile bat bakarrik izatea, edo, behintzat, arduratsu erabiltzea eskaintzen dituen aukerak.



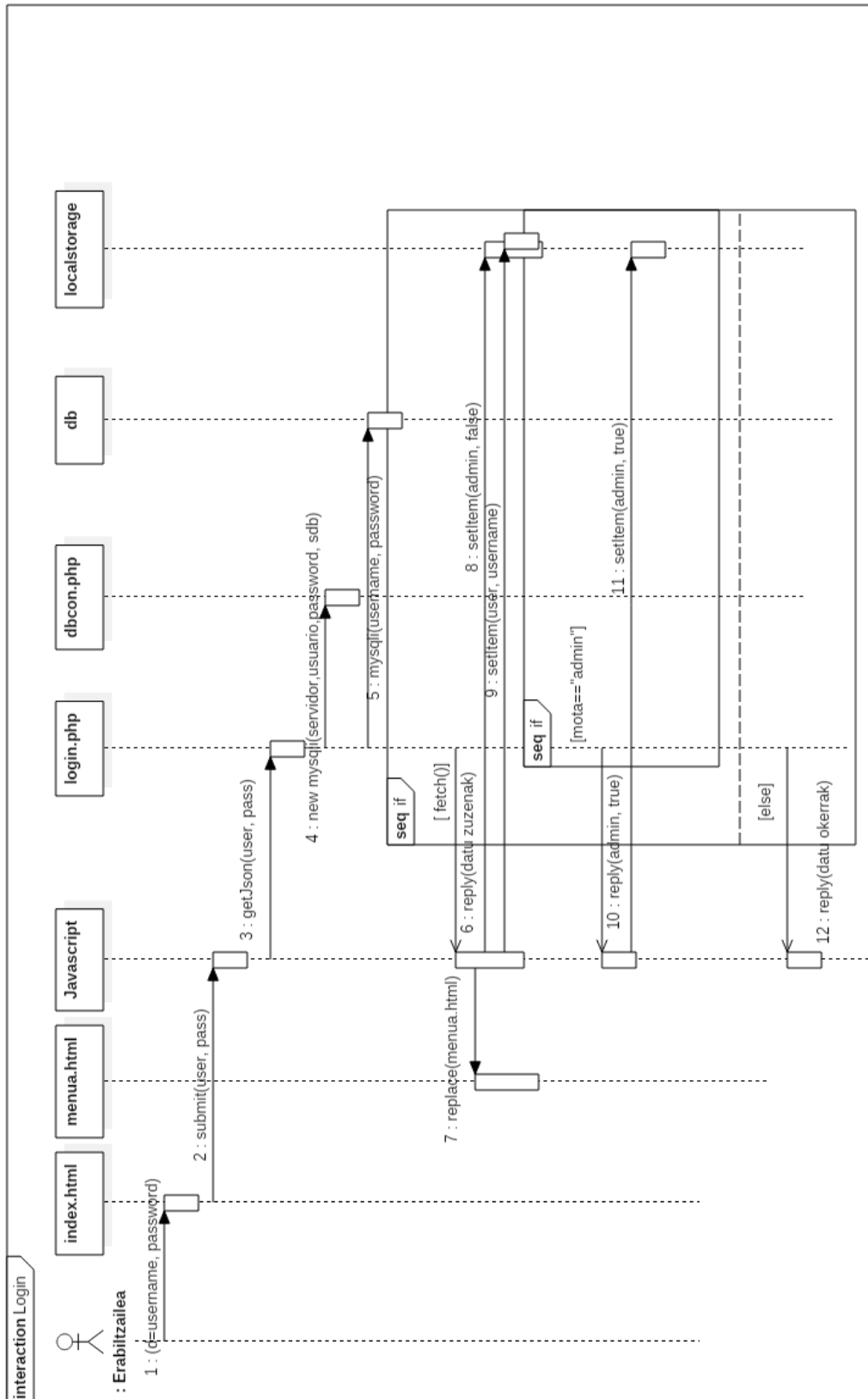
10. irudia. Rolak eta hauei lotutako erabilpen kasuak.

Behin aplikazioaren ikuspegi orokorra daukagula, erabilpen-kasu bakoitza azalduko dugu, bakoitzaren aktoreak, deskribapena eta gertaera- fluxua azalduz.

4.2.2.1 Saioa Hasi

Aktoreak: Guztiak (Irakaslea, Mantentze arduraduna eta Administratzailea)
Deskribapena: Aplikazioaren hasierako leihoan erabiltzaile eta pasahitza sartzeko aukera izango dugu. Kontua zuzena bada, aplikazioaren menura joango gara. Zuzena ez bada berriz, aplikazioak errore-mezua itzuliko digu eta berriz saiatu beharko dugu.
Aurrebaldintzak: Erabiltzailea aplikazioaren hasiera orrian egon beharko da.
Gertaera-fluxua: <ol style="list-style-type: none">1. Erabiltzaileak erabiltzaile izena eta pasahitza idatziko ditu.2. <i>Saioa Hasi</i> botoiari sakatuko du.3. Sistemak, <i>Ajax</i> eskaera baten bidez, datuak datu-basera bidali eta zuzenak direla ziurtatzen du. Administratzailea den ala ez konprobatzen da.4. <i>PHPk</i> eskaintzen digun <i>Session</i>[26]bat hasiko da, aldagaietan erabiltzailearen datuak gordetzeko.5. Erabiltzailea aplikazioaren menura birbidaltzen da.
Ordezko fluxua: <ol style="list-style-type: none">3.b Sistemak datuak okerrak direla konprobatzen du.4.b Erabiltzaileari datuak okerrak direla jakinarazten zaio.5.b Hasiera orrian mantentzen da
Post-baldintza: <ol style="list-style-type: none">A Erabiltzailea aplikazioa sartzen da.B Erabiltzailea hasierako leihoan geratzen da.

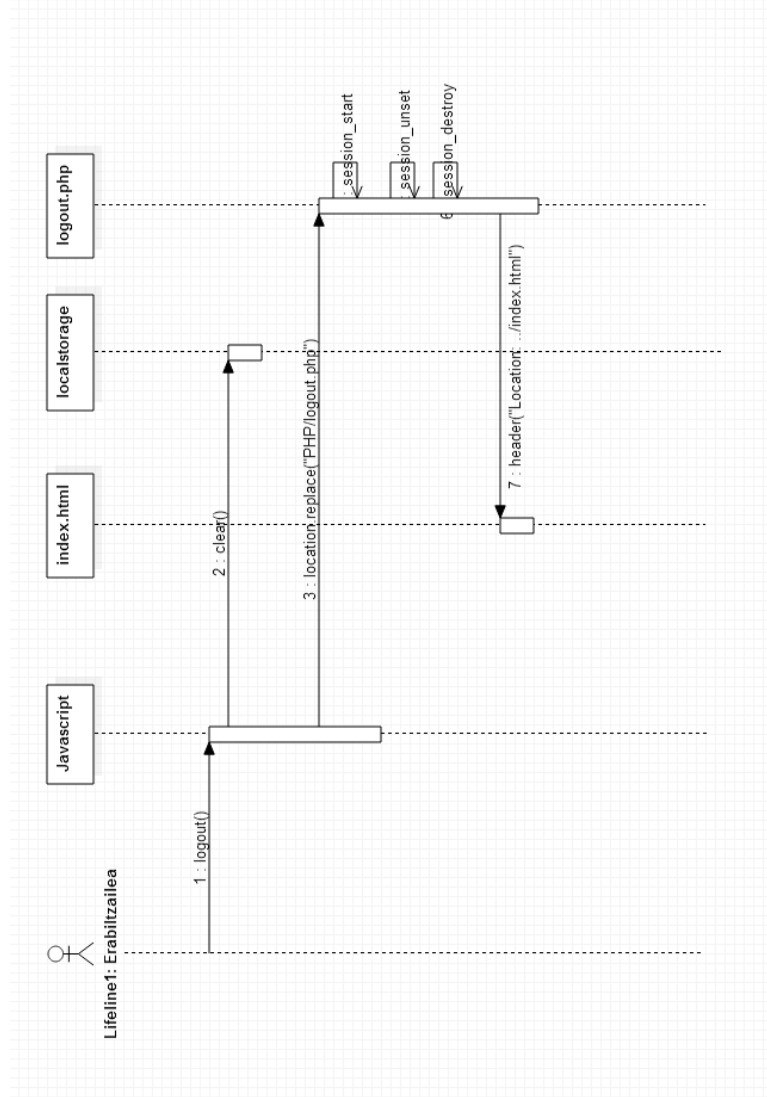
4.5 irudian, saioa hasi erabilpen kasuan sekuentzia diagrama ikusi dezakegu.



11irudia. Saioa hasi erabilpen kasuaren sekuentzia diagrama.

4.2.2.2 Saioa amaitu

Aktoreak: Guztiak (Irakaslea, Mantentze arduraduna eta Administratzailea)
Deskribapena: Aplikazioaren edozein leihotan, goran eskubitara, saioa amaitzeko botoia egongo da. Botoia sakatuz gero, erabiltzailearen saioa amaitu egingo da eta hasiera orrira bideratuko da, automatikoki.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea.
Gertaera-fluxua: <ol style="list-style-type: none"> 1. Erabiltzaileak goran eskuinaldean dagoen <i>Saioa itxi</i> botoia sakatuko du. 2. Botoi honek <i>logout()</i> funtzioari deituko dio. 3. Funtzio honetan <i>localStorage</i> eta <i>session</i>-eko datuak ezabatuko dira. 4. Hasiera orrira birbideratuko gaitu.
Postbaldintza: <p>A Hasiera orrira joango gara.</p>



12 irudia. Saioa amaituren sekuentzia diagrama.

4-2-2-3 Abisua bidali

Aktoreak: Guztiak (Irakaslea, Mantentze Arduraduna eta Administradorea)
Deskribapena: Aplikazioaren menuan dagoen <i>Abisua bidali</i> aukeran sartuz, intzidentzia bat sortzeko aukera izango dugu. Bertan, intzidentziaren inguruko zenbait datu osatu eta dagokion arduradunari bidaliko diogu abisua.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, abisua bidaltzeko leihoan.
Gertaera fluxua: <ol style="list-style-type: none">1. Erabiltzaileak intzidentziaren inguruko datuak beteko ditu (arazo mota, arduraduna, eraikina, solairua, gela, lehentasuna, laburpena, deskribapena eta, aukeran, argazkia)2. <i>Bidali</i> botoia sakatzen du.3. Sistemak bidalketa jaso eta argazkia (igo bada) eta abisua gordeko ditu.4. Erabiltzaileak abisua zuzen bidali dela mezu jasoko du eta menura birbideratuko da.
Ordezko fluxua: <ol style="list-style-type: none">3.b Daturen bat bete gabe badago, jakinarazpen-mezua erakutsi eta eskaera ez da bidaliko.4.b Sistemak datuak bete eta berriz ere bidaltzeko aukera emango digu.
Postbaldintza: <ol style="list-style-type: none">A Abisua zuzen bidali eta menura itzuliko gara.B Leihoan bertan mantenduko gara, datuak zuzen betetzeko.

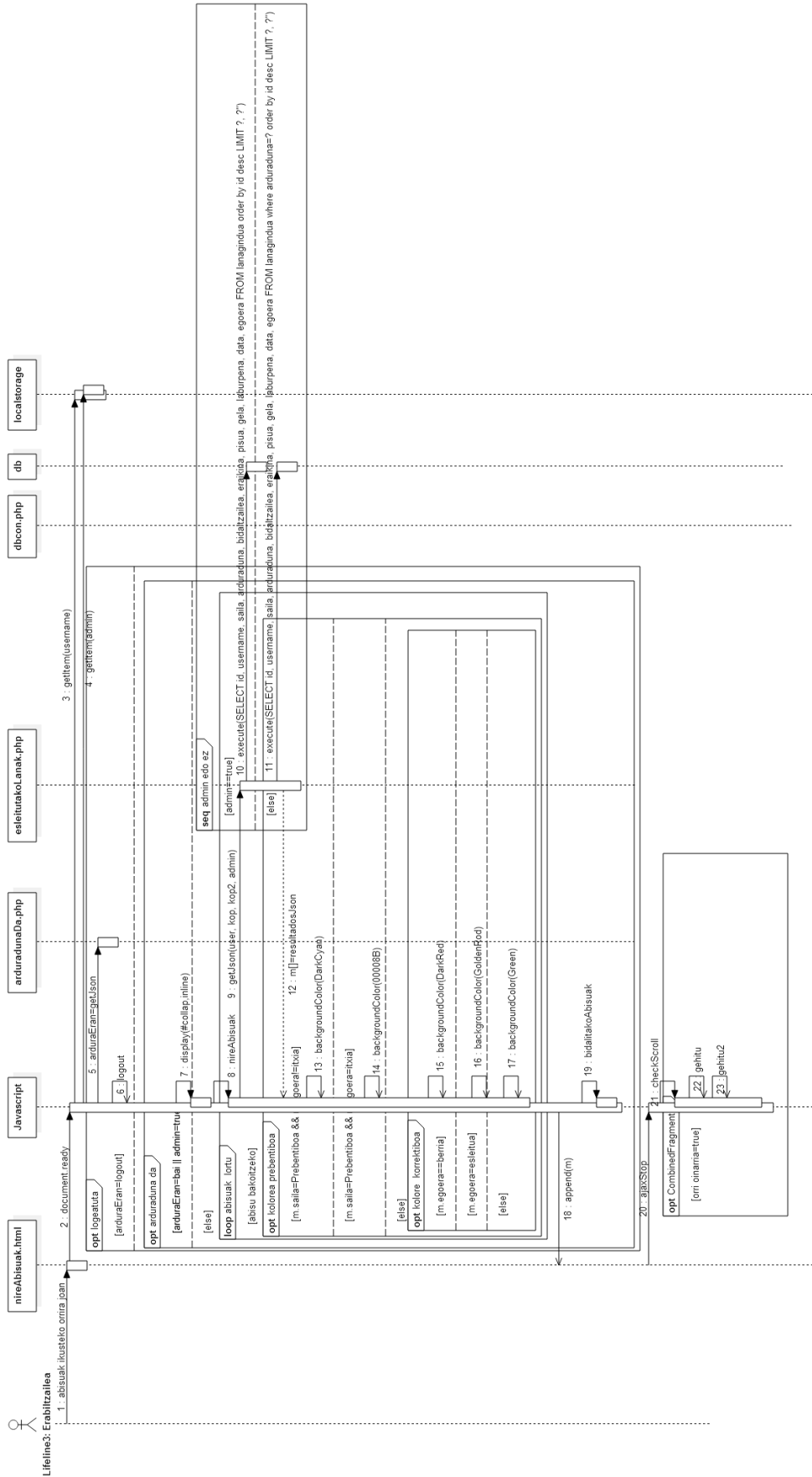
4-2-2-4 Bidalitako abisuak ikusi

Aktoreak: Guztiak (Irakaslea, Mantentze Arduraduna eta Administrazioa)
Deskribapena: Aplikazioaren menuan dagoen <i>Nire abisuak</i> aukeran sartuz, bidalitako abisuak ikusteko aukera izango dugu. Bertan, <i>Bidalitako abisuak</i> tolesgarrian klik egin eta bidalitako abisuen inguruko informazioa ikusiko ditugu, dataren arabera ordenatuta.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, <i>Nire Abisuak</i> leihoan.
Gertaera fluxua: <ol style="list-style-type: none">1. Erabiltzaileak <i>Bidalitako abisuak</i> tolesgarria irekiko du.2. Sistemak saioko erabiltzailea hartu eta datu-basean bidalitako abisuak aurkituko ditu, lehenengo 10ak.3. Abisuen informazioa pantailan erakutsiko da, lista moduan.4. Erabiltzailea, abisu gehiago ikusteko, zerrendan behera jaitsiko da, hondora heldu arte5. Hondora heltzean sistemak beste <i>Ajax</i> eskari bat egingo du, hurrengo 10 abisuak zerrendara gehituz. 4. eta 5. pausuak nahi adina aldiz errepikatuko dira, datu-baseko abisu guztiak erakutsi arte.
Postbaldintza: A Tolesgarrian abisuak ikusiko ditugu.

4.8 irudian ikus dezakegu esleitutako abisuen sekuentzia diagrama. Sekuentzia diagrama bera izango da, baina esleitutako kasuan administrazioa den ala ez konprobatuko da, abisu guztiak erakutsi ala ez jakiteko. Bestalde, ze abisu tarte erakutsiko den erabakitzen duen baldintza ere ez dugu sekuentzia diagraman adierazi, jada nahiko konplexua dela iruditu baitzaigu.

4-2-2-5 Esleitutako abisuak ikusi

Aktoreak: Mantentze arduraduna eta administrazioa
Deskribapena: <i>Bidalitako abisuak ikusi</i> kasuaren modu berean funtzionatzen du. Kasu honetan ordea, arduradunak eta administrazioa bakarrik edukiko dute tolesgarria ikusgarri. Bertan, arduradunari esleitutako abisuak, edo administrazioa izanez gero abisu guztiak, ikusteko aukera izango da.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, <i>Nire Abisuak</i> leihoan.
Gertaera fluxua: Bidalitako abisuen bera izango da.
Ordezko fluxua: 1.b Saioa administrazioa moduan hasiz gero, sistemak abisu guztiak erakutsiko dizkigu, erabiltzailearen arabera esleitutako abisuak iragazi beharrean. Beste pausu guztiak berdinak izango dira.
Postbaldintza: A Tolesgarrian abisuak ikusiko ditugu.

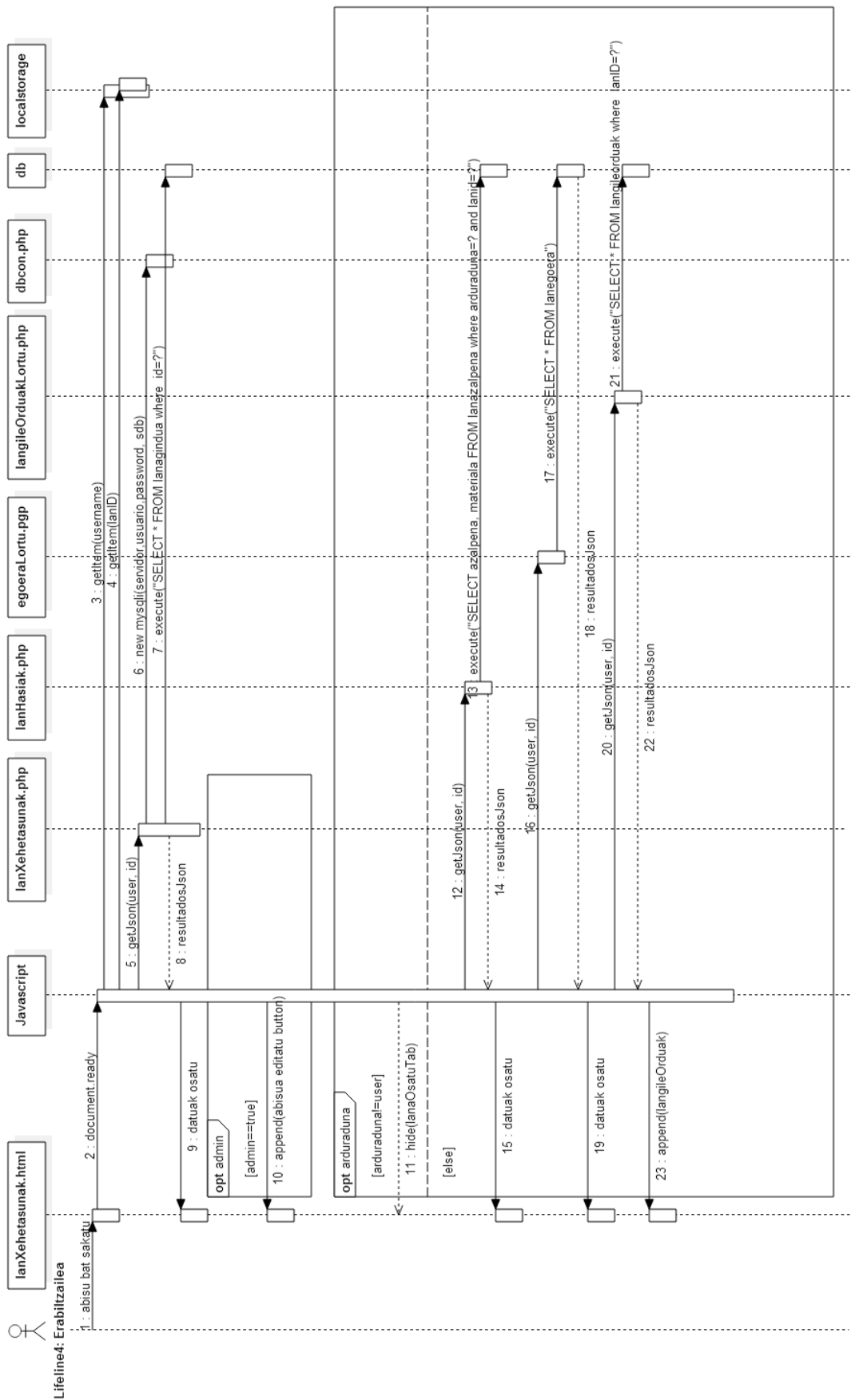


14 esleitutako abisuak sekuentzia diagrama.

4-2-2-6 Abisuaren xehetasunak ikusi

Aktoreak: Guztiak (Irakaslea, Mantentze Arduraduna eta Administratzailea)
Deskribapena: <i>Nire Abisuak</i> leihoan gaudela, abisuen zerrenda ikusiko dugu. Horrela, zerrendako abisu baten gainean sakatuz gero, <i>Lan xehetasunak</i> leihora bideratuko gaitu sistemak, abisu horren datu guztiak erakutsiz.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, <i>Nire Abisuak</i> leihoan, abisuen tolesgarriren bat irekita duela.
Gertaera fluxua: <ol style="list-style-type: none">1. Erabiltzaileak nahi duen abisua sakatuko du.2. Sistemak <i>lanXehetasunak(id)</i> funtzioari deituko dio, lanaren ID-a parametro moduan duelarik.3. Funtzioak <i>localstorage</i>-en abisuaren ID-a gordeko du eta <i>Lan xehetasunak</i> orrira bideratuko gaitu.4. <i>Lan xehetasunak</i> orrian gordetako ID-a jaso eta datu-basean honen bilaketa egingo du, datuak itzuliz.5. Abisuari dagozkion datuak bistaratuko zaizkigu.
Postbaldintza: A Abisuaren datuak ikusiko ditugu.

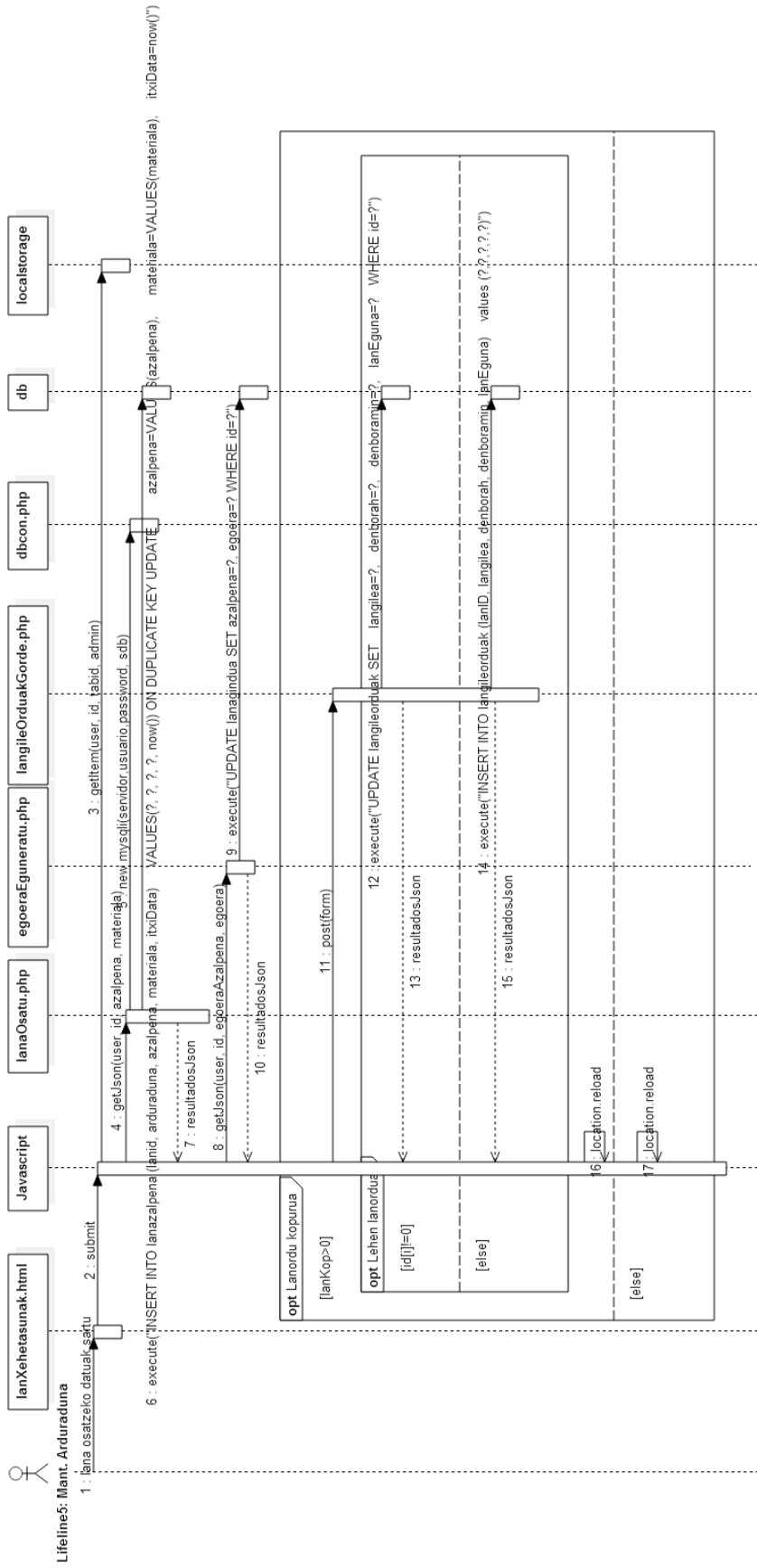
4.9 irudian ikusi dezakegu abisuaren xehetasunak ikusteko sekuentzia diagrama.



15 irudia. Abisuaren xehetasunak ikusti erabilpen kasuaren sekuentzia diagrama.

4.2.2.7 Lana osatu

Aktoreak: Mantentze Arduraduna
Deskribapena: Abisu baten xehetasunetan sartzen garenean bi erlaitz ikusiko ditugu. Lehen, defektuzkoa, abisuaren xehetasunak ikusteko izango da. Bestean sartuz gero ordea lana osatzeko aukera izango dugu, lanaren egoera, deskribapena, lana egiteko erabilitako materiala eta sartutako orduak gordetzeko aukerarekin.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, arduradun moduan, esleitutako lan baten <i>Abisuen xehetasunak</i> leihoan, <i>Abisua osatu</i> erlaitzean.
Gertaera fluxua: <ol style="list-style-type: none"> 1. Abisua osatzeko nahi ditugun datuak idatziko ditugu (lanaren egoera, deskribapena eta erabilitako materiala) 2. Gorde botoia sakatuko dugu. 3. Sistemak idatzitako datuak datu-basean gordeko ditu, <i>lanAzalpenak</i> taulan lanaren lanaren informazioa gordez. Honetaz gain, egoera aldatzen bada, eguneratu egingo du <i>lanAgindua</i> taulatik. 4. Datuak gorde eta leihoa birkargatuko da.
Ordezko fluxua: <ol style="list-style-type: none"> 1.b Langile berria klikatzen bada, sartutako lan-orduak gordetzeko aukera izango dugu. Bertan langilea, laneguna eta sartutako ordu kopurua zehaztuko ditugu. Botoi hau nahi adina aldiz klikatuko dugu, behar ditugun langile/lanegun-en arabera. 2.b Gorde botoia sakatuz gero, fluxu arrunteko pausoak jarraitzeaz gain, langile-orduak taulan langile/lanegun bakoitzari dagokion informazioa gordeko du sistemak, bakoitzarekin errenkada berri bat sortuz. Horretaz gain, lehenago ezarritako langile/lanegun bat aldatzen bada, datu-basean eguneratu egingo da. 2.c Sortu dugun langile/lanegunaren <i>Ezabatu</i> botoia sakatuz gero, langile/laneguna ezabatuko zaigu. Momentuan sortu badugu, ezabatu gainean sakatzea nahikoa izango da ezabatzeko. Bestela, lehenagotik datu-basean gordetako bada, baieztapen abisua aterako zaigu. 3.c Mezuari baiezkua emanaz gero, langile/lanegunaren ID-a datu basean bilatuko da, ondoren ezabatzeko. 4.c Lana osatu erlaitzean jarraituko dugu, dagoeneko langile/laneguna ezabatuta. 3.d Mezuari ezezkoa emanaz gero, lana osatu erlaitzean jarraituko dugu, ezabatu gabe.
Postbaldintza: <ol style="list-style-type: none"> A Egindako lanaren azalpenak gordeko ditugu.



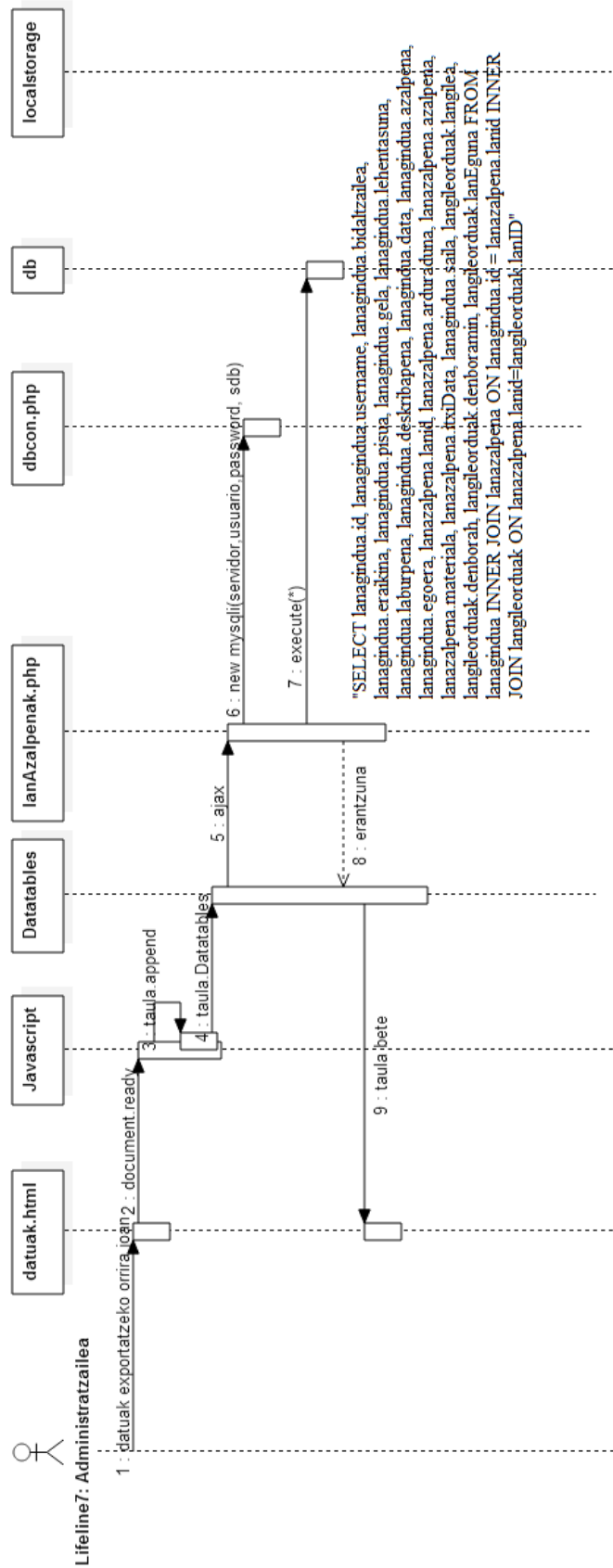
16 irudia. Lana osatu erabilpen kasuaren sekuentzia diagrama

4.2.2.8 Abisuak editatu

Aktoreak: Administratzailea
Deskribapena: Administratzaile moduan, abisu baten xehetasunak ikustera sartzean, behean <i>Abisua editatu</i> botoia edukiko dugu. Hau sakatuz gero, abisua bidaltzeko leihoaren antzekoa den bat azalduko zaigu. Bertan, abisuaren informazioa aldatzeko aukera izango dugu.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, <i>Lan xehetasunak</i> leihoko <i>Abisua editatu</i> botoia sakatzea .
Gertaera fluxua: <ol style="list-style-type: none">1. Erabiltzaileak intzidentziaren inguruko datuak aldatuko ditu (arazo mota, arduraduna, eraikina, solairua, gela, lehentasuna, laburpena eta deskribapena)2. Eguneratu botoia sakatzen du.3. Sistemak bidalketa jasotzen du eta abisua eguneratzen du.4. Erabiltzaileak abisua zuzen eguneratu dela mezua jasoko du eta menura birbideratuko da.
Ordezko fluxua: <ol style="list-style-type: none">3.b Datuen bat bete gabe badago, jakinarazpen mezua erakutsi eta eskaera ez da bidaliko.4.b Sistemak datuak bete eta berriz ere bidaltzeko aukera emango digu
Postbaldintza: <ol style="list-style-type: none">A Abisua zuzen eguneratu eta menura joango gara.B Leihoan bertan mantenduko gara, datuak zuzen betetzeko.

4.2.2.9 Datuak esportatu

Aktoreak: Administratzailea
Deskribapena: Administratzaile moduan, menuan, datuak esportatzeko aukera ematen digun botoia izango dugu (abisuen datuak, egindako lanaren azalpena eta egindako lan orduen informazioa izango dugu eskuragarri). Bertan, datuak taula batean azalduko zaizkigu, esportatzeko botoiak eta guzti. Horretaz gain, taula iragazteko hainbat aukera izango ditugu.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, administratzaile moduan eta datuak esportatzeko leihoan egotea.
Gertaera fluxua: <ol style="list-style-type: none">1. Sistemak datu basetik dagozkion datuak itzuliko dizkio, taula moduan.2. Administratzaileak datuak esportatzeko botoia sakatuko du, Export, CSV edo PDF formatua hautatuz.3. Sistemak eskaera jaso eta taula ordenagailura jaitsiko da, hautatutako formatuan.
Ordezko fluxua: <ol style="list-style-type: none">2.b Administratzaileak aukeran dituen iragazkiak erabiliko ditu (Data minimoa, data maximoa edo zutabekako iragazketa)3.b Administratzaileak datuak esportatzeko botoia sakatuko du, Excel edo CSV formatua hautatuz.4.b Sistemak eskaera jaso eta taula ordenagailura jaitsiko da, iragazitako datuekin soilik, hautatutako formatuan.
Postbaldintza: <ol style="list-style-type: none">A Datu-fitxategia jaitsiko dugu..



18 irudia. Datuak exportatu erabilpen kasuaren sekuentzia diagrama.

4.2.2.10 Lan prebentiboak sortu

Aktoreak: Administratzailea
Deskribapena: Aplikazioaren menuan dagoen <i>Lan prebentiboak sortu</i> aukeran sartuz, lan prebentiboak sortzeko aukera izango dugu. Bertan lanaren inguruko datuak beteko ditugu. Horretaz gain, eginbeharreko lana burutzeko hainbat datu sartzeko aukera izango dugu, bakoitzarekin lan prebentibo bat sortuz.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, administratzaile moduan, lan prebentiboak sortzeko leihoan.
Gertaera fluxua: Administratzaileak intzidentziaren inguruko datuak beteko ditu (arduraduna, eraikina, solairua, gela, lehentasuna, laburpena, deskribapena eta data). <i>Bidali</i> botoia sakatzen du. Sistemak bidalketa jaso eta lana gordeko du. Erabiltzaileak lana zuzen bidali dela mezu jasoko du eta menura birbideratuko da.
Fluxu alternatiboa: 3.b Daturen bat bete gabe badago, jakinarazpen mezua erakutsi eta eskaera ez da bidaliko. 4.b Sistemak datuak bete eta berriz ere bidaltzeko aukera emango digu.
2.c Administratzaileak <i>Data berria</i> botoia sakatuko du eta lana egin beharreko data berria txertatuko du. 2.c puntua nahi adina aldiz errepikatuko da. 3.c <i>Bidali</i> botoia sakatzen du. 4.c Sistemak bidalketa jasotzen du eta lanak gordetzen ditu, data bakoitzeko bat. 5.c Erabiltzaileak lana zuzen bidali dela mezu jasoko du eta menura birbideratuko da.
3.d Administratzaileak dataren ondoko <i>Ezabatu</i> botoia sakatuko du. 4.d Data berria ezabatuko da. 5.d Fluxua 2.c puntura itzuliko da.
Postbaldintza: A Abisua zuzen bidali eta menura joango gara. B Leihoan bertan mantenduko gara, datuak zuzen betetzeko.

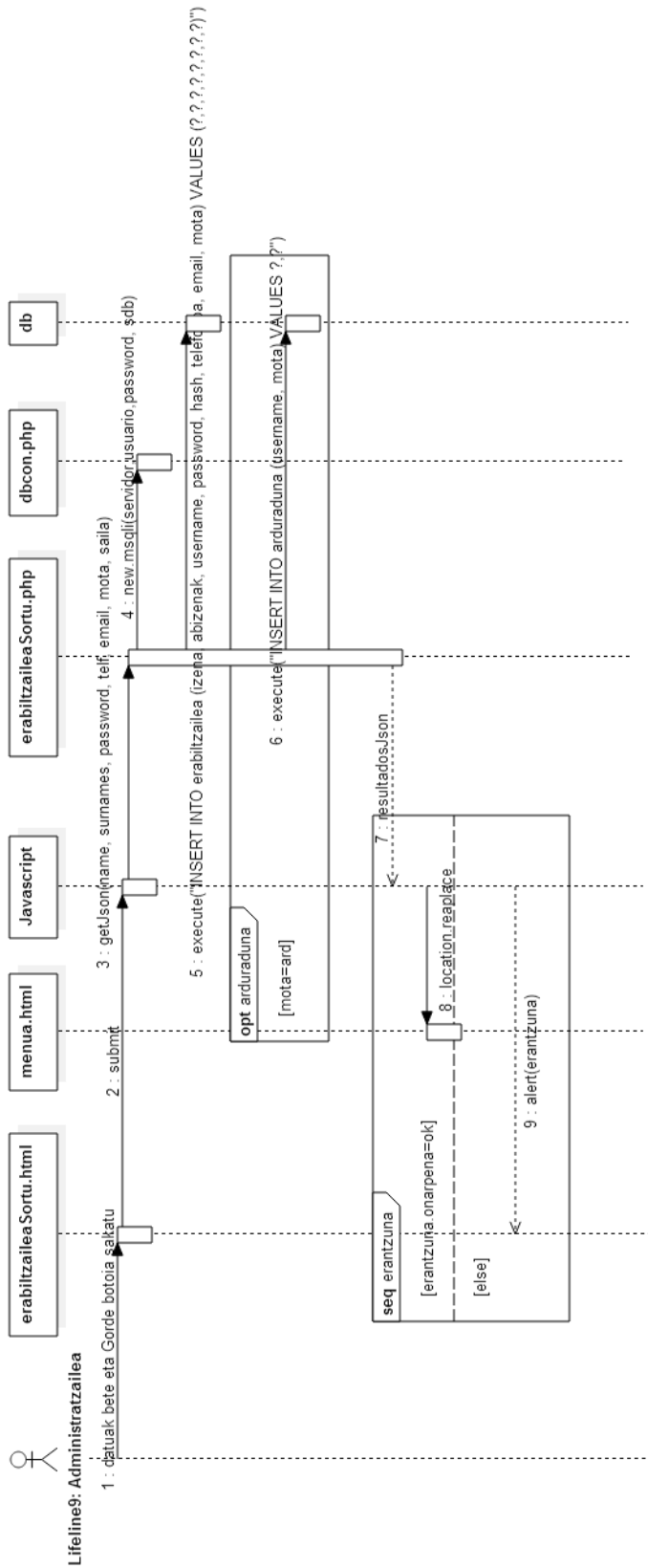
Lan prebentiboak sorturen sekuentzia diagrama abisua bidaliren oso antzekoa denez (ikus 4.7 irudia), txostenean ez sartzea erabaki dugu. Horrela, ezberdintasun bakarra data bakoitzeko abisu bat sortuko litzatekeela izango da, abisu bakarra sortu ordez.

4.2.2.11 Erabiltzailea sortu

Aktoreak: Administratzailea
Deskribapena: Aplikazioaren menuan dagoen <i>Erabiltzailea sortu</i> aukeran sartuz, erabiltzaileak sortzeko aukera izango dugu. Bertan erabiltzailearen inguruko datuak beteko ditugu.

Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, administratzaile moduan, erabiltzailea sortzeko leihoan.
Gertaera fluxua: Administratzaileak erabiltzailearen inguruko datuak beteko ditu (izena, abizenak, erabiltzaile izena, pasahitza, telefonoa, posta elektronikoa, mota eta, arduraduna bada, saila). <i>Gorde</i> botoia sakatzen du. Sistemak bidalketa jaso eta erabiltzailea gordeko du. Erabiltzailea menura birbideratuko da.
Fluxu alternatiboa: 3.b Daturen bat bete gabe badago, jakinarazpen mezua erakutsi eta eskaera ez da bidaliko. 4.b Sistemak datuak bete eta berriz ere bidaltzeko aukera emango digu.
Postbaldintza: A Erabiltzailea zuzen sortuko da eta menu nagusira itzuliko gara. B Leihoan bertan mantenduko gara, datuak zuzen betetzeko.

4.13 irudian *Erabiltzailea sorturen* sekuentzia diagrama ikusi dezakegu.



19 irudia. Erabiltzailea sortu erabilpen kasuaren sekuentzia diagrama.

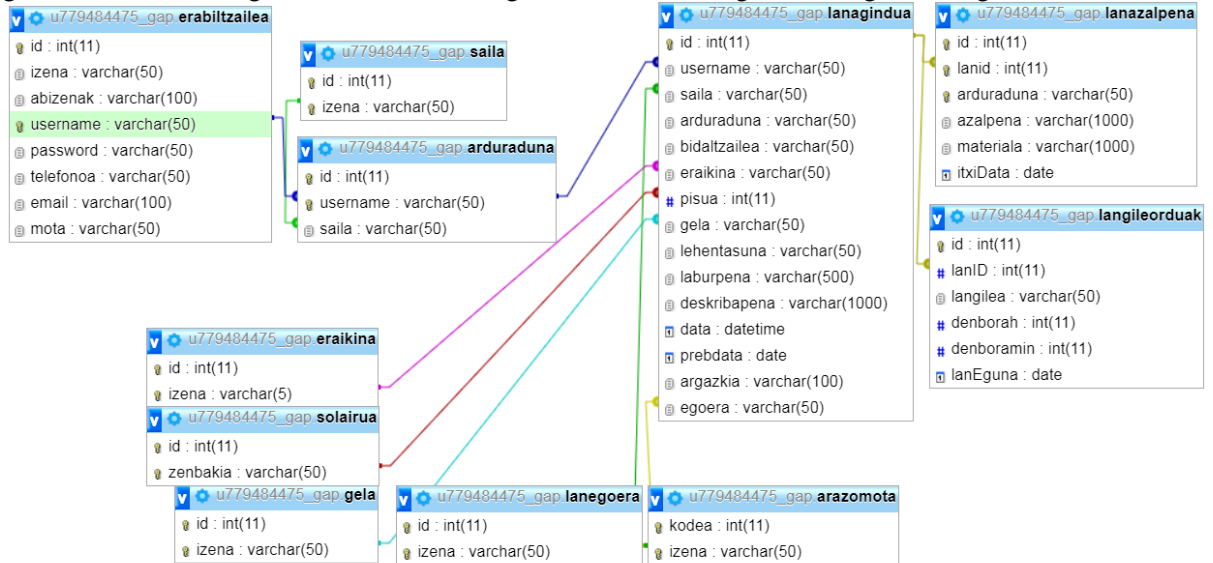
4.2.2.12 Erabiltzailea editatu edo ezabatu

Aktoreak: Administratzailea
Deskribapena: Aplikazioaren menuan dagoen <i>Erabiltzailea editatu edo ezabatu</i> aukeran sartuz, erabiltzaile zerrenda agertuko zaigu. Bertako batean sakatzean, erabiltzailearen datuak agertuko zaizkigu eta berau editatu edo ezabatzeko aukera izango dugu.
Aurrebaldintzak: Erabiltzailea aplikazioan saioa hasita egotea, administratzaile moduan, erabiltzailea editatzeko leihoan.
Gertaera fluxua: <ol style="list-style-type: none">1. Administratzaileak erabiltzaile bat sakatuko du zerrendan. Ondoren, eguneratu nahi duen datuak aldatu beharko du.2. <i>Gorde</i> botoia sakatuko du.3. Sistemak bidalketa jaso eta erabiltzailea gordeko du.4. Erabiltzaile zerrendara birbideratuko da.
Fluxu alternatiboa: <ol style="list-style-type: none">3.b Daturen bat bete gabe badago, jakinarazpen mezua erakutsi eta eskaera ez da bidaliko.4.b Sistemak datuak bete eta berriz ere bidaltzeko aukera emango digu.
<ol style="list-style-type: none">3.c Administratzaileak Ezabatu botoia sakatuko du.4.c Erabiltzailea datu basetik ezabatuko da.
Postbaldintza: <ol style="list-style-type: none">A Erabiltzailea zuzen eguneratu/ezabatu eta erabiltzaileen zerrendara itzuliko gara.B Leihoan bertan mantenduko gara, datuak zuzen betetzeko.

4.14 irudian *Erabiltzailea editaturen* sekuentzia diagrama ikusiko dugu.

4.2.3 Datu-basea

Lehenago esan dugun moduan, *MySQL* datu-base bat erabiliko dugu aplikazioko datuak gordetzeko. Sortu dugun datu-basearen diagrama 4.5 Irudian agertzen zaiguna izango da:



21. irudia. Datu basearen diagrama.

Datu-basean jasotzen diren taulak honako hauek dira:

Erabiltzailea: Aplikazioan sartzeko baimena izango duten erabiltzaile guztien informazioa gordeko da bertan. Gainera, erabiltzailea arduraduna bada, Arduraduna taularekin erlazioatuko da.

Arduraduna: Aplikazioko arduradunak biltzen dituen taula izango da. Bertan, erabiltzaile taularekin erlazioa osatuko da, erabiltzaile izenaren bidez. Horretaz gain, arduraduna zein sailetakoa den ere gordeko da.

Lan agindua: Aplikazioko taula nagusia dela esan liteke. Taula horretan gordeko dira bidalitako abisu guztiak. 4.5 irudian ikusi ditzazkegu abisuak izango dituen hainbat ezaugarri. Horietako bat argazkia da. Argazkia, ordea, ez da zuzenean argazkia gordeko den tokia izango. Izan ere, atributu honetan argazkiaren izena soilik gordeko da eta, argazkia bera, zerbitzariko fitxategi-sisteman gordeko da.

Lan azalpena: Lan aginduari zentzua ematen dion taula da. Bertan, jasotako intzidentzia konpondu osteko datuak gordeko dira. Hau da, behin lana eginda, lana konpontzeko eman diren pausoak azalduko dira eta, horretaz gain, erabilitako materiala, behar izan bada. Azken eguneraketaren data ere gordeko da.

Langile orduak: Lanaren inguruko azalpena eta materialaz gain, sartutako lan-orduak gordetzea ezinbestekoa izango da. Horretarako, beste taula hau sortu da. Bertan, egindako lanarekin lotura egingo da, ID-aren bidez eta sartutako lan orduak (ordu eta minutuak), lan eguna eta langilea gordeko dira. Egindako lan bakoitzak hainbat langile/lanordu/lanegun izan ditzake.

Eraikina/solairua/gela: Intzidentzia kokatzeko ezinbesteko taulak. Tolosaldeako gela bakoitza honako modu honetan dago identifikatuta: lehenik eraikinaren letra, jarraian solairua eta, azkenik, puntu batez bereizita, gela zenbakia. Adibidez: E0.9 edo F2.1.

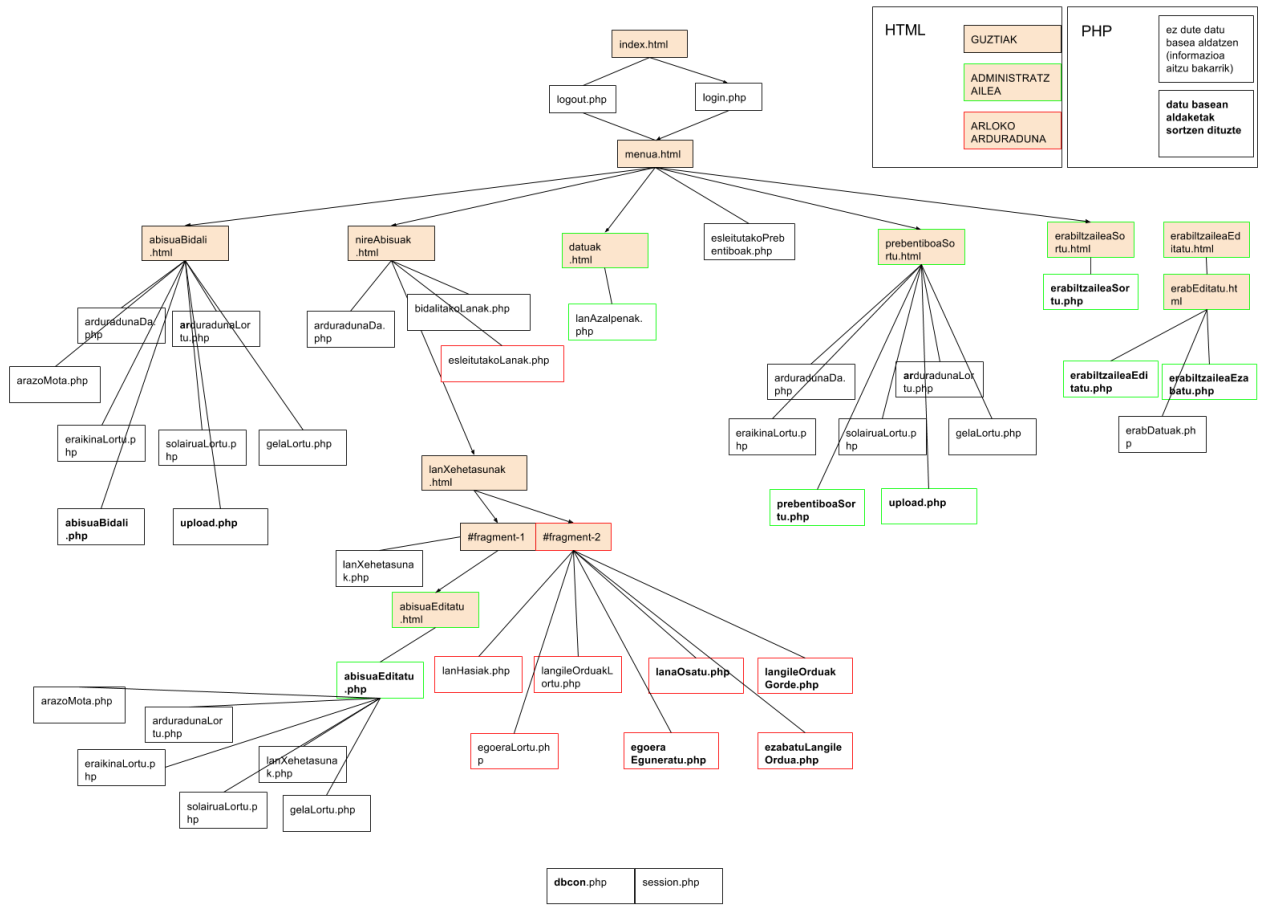
Modu horretan, kokapena hiru taulatan banatuta, modu erraz eta argi batean tokia identifikatuko dugu. Gainera, eraikin edo gelaren bat berritzen bada, aldaketak modu errazean egin ahalko ditugu.

Lan egoera: Izenak dioen moduan lanaren egoera gordeko da hemen, hau da, momentuan lana zein fasetan dagoen. Adibidez, abisua bidali berria bada, lanaren egoera berria izango da. Behin irakurrita badago, esleitua moduan ezarriko dugu. Azkenik, lana amaitzen dugunean, itxia egoera hautatu beharko dugu.

Arazo mota: Intzidentzia bakoitza arazo mota batekoa izango da. Modu horretan, abisua bidaltzean arazo mota aukeratzeko dugunean, arduraduna hautatzeko aukeran arazoari dagokion saileko arduradunak bakarrik izango ditugu eskuragarri. Horrela, ezjakintasunagatik edo ustekabean arduradun ezegokiak jartzea ekidingo da

4.2.4 Klase-diagrama

4.14 irudian, aplikazioaren klase diagrama ikusi dezakegu. Bertan, aplikazioa osatzen duten orriak, bai html eta bai php orriak, ikusi ditzakegu. Html orriak erabiltzen dituen php orriak gezi bidez lortu ditugu. Horretaz gain, orriak koloreen arabera bereiztu ditugu: Karratu zuriak php orriak izango dira eta laranja berriz html orriak. Html orrien artean, ertz beltza dutenak aktore guztiek izango dituzte eskuragarri, ertz berdea dutenak berriz administratzaileak soilik eta gorria dutenak arduradunek. Bestalde, letra lodiz ageri diren php orriak datu basean aldaketak egingo dituzte, besteak berriz irakurketak bakarrik.



22 irudia. Aplikazioaren orrien egitura.

5

Software-garapena

Kapitulu honetan, garapenaren nondik norakoak azalduko ditugu.

Softwarea proiektuaren oinarria denez, denbora luzez aritu gintezke emandako pausoak azaltzen. Horregatik, aplikazioaren puntu esanguratsuenak aztertuko ditugu, softwarearen funtzionalitate nagusiak azalduz, honen funtzionamendua ulertzeko, modu labur batean. Gainera, azalpenari laguntzeko, interfazearen irudiak txertatu ditugu, emandako azalpenak aplikazioan nola ageri diren ikusteko. Azkenik, esanguratsuak diren kode-zatiak txertatu ditugu, softwarearen erroak zuzenean ikusi eta azaltzeko asmoz. Bestalde, kode osoa ikusteko, bibliografian utzi dugu github plataformarako helbidea[27].

Behin garatutako ezaugarriak azalduta, garapenean sortu zaizkigun arazoak eta hauei soluzioa emateko burututako pausoak azalduko ditugu.

5.1 Garatutako ezaugarriak

5.1.1 Hasiera orria/Saioa hasi

Lehenik eta behin, aplikazioaren hasiera den *index* orrialdea edukiko dugu. Aplikazioaren edozein ezaugarri erabiltzeko saioa hasi behar denez, hasierako leihoan Tolosaldea Institutuko logoaz gain, zuzenean saioa hasteko aukera txertatu dugu. Izan ere, erabiltzaileak saioa hasi arte ez du ezer ikusteko/erabiltzeko aukerarik izango eta, ondorioz, hasierako orria saioa hasteko erabiltzea pentsatu dugu.



Tolosaldea

Lanbide Heziketa Institutua

Erabiltzailea:

Pasahitza:


Saioa Hasi

23.irudia. Hasiera orria, saioa hasteko leihoa.

5.1 irudian ikusten den moduan, erabiltzailea eta pasahitza sartzeko eremuak izango ditugu eta, behin sartuta, saioa hasteko botoia sakatuko da.

5.1.2 Menua

Behin saioa hasita, aplikazioaren menu nagusia ikusiko dugu (ikus 5.2 irudia). Bertan, bi aukera izango ditugu, batetik, abisuak bidaltzeko aukera eta, bestetik, gure abisuak ikusteko aukera. Horretaz gain, saioa administratzaile moduan hasten badugu, beste lau aukera edukiko ditugu: *Datuak esportatu*, *Lan prebentiboak sortu*, *erabiltzailea sortu* eta *Erabiltzailea editatu edo ezabatu* aukera.

 Saioa itxi	Menua
Abisua Bidali	
Nire Abisuak	
Datuak ikusi	
Lan prebentiboak sortu	
Erabiltzailea sortu	
Erabiltzailea editatu edo ezabatu	

24.irudia. Aplikazioaren menu nagusia, administratzailearen ikuspegia.

5.1.3 Abisua bidali

Lehen aukera, abisua bidaltzekoa izango da (ikus 5.3 irudia). Bertan, erabiltzaileari gauzak erraztearren, bete beharreko datu ia guztiak aukeratzekoak izango dira. Horrela, soilik laburpena eta deskribapena idatzi beharko ditu abisua bidali nahi duen erabiltzaileak.

Atzera

Abisua bidali

Saioa itxi

Arazo mota

Arduraduna

Eraikina

solairua

Gela

Lehentasuna

aaramburu

Laburpena

Deskribapena...

Argazkia igo (Hautazkoa)

Bidali

25. irudia. Abisua bidaltzeko leihoa.

5.1.4 Abisuak ikusi

Abisuak ikusteko aukera sakatzen badugu, 5.4 irudiko leihoa ageriko zaigu. Bertan, bidalitako abisuak ikusteko aukera izango dugu eta, mantentze-arduradun edo administratzailea bagara, baita esleitutako abisuak ere. Abisu hauen laburpen txiki bat ikusteko aukera edukiko dugu bertan: abisua bidalitako data, abisuaren egoera, arloa, tokia eta laburpena, zehazki. Gainera, abisuaren egoeraren arabera goiburuko koloreak aldatu egingo dira: abisua berria bada, kolore gorria, behin irakurritakoan horia eta, ixten dugunean, berdea. Bestalde, lan prebentiboak beti kolore urdinez ageriko zaizkigu, egoera zein den axola gabe.

Nire Abisuak	
<p>Esleitutako Abisuak</p> <p>Filter items...</p> <p>2017-07-13 08:05:49 prebentiboa</p> <p>Prebentiboa-A2.pasilloa 123 Ur kontadorea begiratu.</p> <p>2017-06-28 10:44:00 esleitura</p> <p>Galdaragintzako Mant.-A-1.10 107 sagdg</p> <p>2017-06-28 09:33:57 itxia</p> <p>Mant. Orokorra-E0.pasilloa 105 Ordenagailua ez dabil.</p>	<p>Bidalitako Abisuak</p> <p>Filter items...</p> <p>2017-05-31 08:22:41 a esleitura</p> <p>Mant. Informatikoa-G2.20 81 Proba android</p> <p>2017-05-29 08:18:29 aaramburu berria</p> <p>Argia-A1.1 77 Proba android+hosting</p> <p>2017-05-18 09:57:16 a esleitura</p> <p>Argia-A1.1 71 dvsa</p>

26. irudia. Abisuak ikusteko leihoa.

5.1.5 Lanaren xehetasunak ikusi

Abisuetak baten gainean sakatzen dugunean, bere xehetasunak ikusteko leiho batetara bideratuko gaitu sistemak. Bertan, abisuaren inguruko datu guztiak izango ditugu eskuragarri. Informazio hau modu bereizgarri eta argian erakusteko, taula egitura erabiltzea erabaki da, 5.5 irudian ikusten den moduan.

Lan Xehetasunak			
Abisuaren xehetasunak		Abisua Osatu	
ID	Saila	Lehentasuna	Bidaltzailea
123	Prebentiboa	Normala	admin
Tokia		Bidaltze data	
A2.pasilloa		2017-07-13 08:05:49	
Laburpena		Deskribapena	
Ur kontadorea begiratu.		Ur kontadorea begiratu.	
Eginbeharreko eguna			
2017-09-03			
Argazkia			
Ez dauka argazkirik.			

27. irudia. Abisuaren xehetasunak.

5.1.6 Lana osatu

5.5 irudian, argi ibili bazara, bi erlaitz daudela konturatuko zinen. Bata, abisuaren xehetasunak ikusteko erlaitzari dagokio. Bestea, berriz, mantentze-arduradunak soilik ikusi ahal izango du, esleitutako abisuetan bakarrik gainera. Erlaitz horretan, lanaren egoera eguneratzeko aukera izango dugu (ikus 5.6 irudia), egindako lanaren inguruko deskribapena idaztea eta lan hau egiteko erabilitako materiala azaltzea. Honetaz gain, lana egin duten langileak sartzeko aukera izango dugu, *langile berria* botoia sakatuz. Bertan, langilea hautatuko da, lan eguna eta egunean sartutako lanorduak, ordu eta minututan banatuta. Horrela, egindako lanaren inguruko xehetasun guztiak gordeko ditugu, ondoren honen jarraipena egiteko.

Atzera Lan Xehetasunak Saioa itxi

Abisuaren xehetasunak Abisua Osatu

berria

Azalpena...

Erabilitako materiala...

Bidaltzailearentzako egoera azalpena...

Ezabatu

Langilea: Lan-eguna: Denbora (h): Denbora (min):

Langilea Egin klik data sartzeko... 25 30

Langile berria

Gorde

28. irudia. Lana osatzeko erlaitza.

5.1.7 Datuak esportatu

Leiho honetan abisuak eta burututako lanen inguruko datuak esportatzeko aukera edukiko dugu (ikus 5.7 irudia). Bertan, abisu bidaltzailea, abisuaren bidaltze data, eraikina, egindako lanaren azalpena, langileak, sartutako lanorduak... eta beste hainbat datu ikusteko aukera dago. Ikusi nahi ditugun datuak *Column visibility*-n hautatuko ditugu. Horrez gain, zutabe bakoitza filtratzeko aukera edukiko dugu eta baita data tarte jakin bateko abisu eta lanak bakarrik bilatzeko aukera ere. Nahi ditugun datuak filtratu ostean, taula esportatzeko botoia sakatu beharko dugu, CSV edo Excel botoiak hain zuzen.

Atzera Saltoa itxi

Data min Reset

Data max Reset

Column visibility Excel CSV

Search:

Lan id	Username	Eraikina	Laburpena	Egoera	Saila	Langilea	Denbora (h)	Denbora (min)	Lan Eguna
52	aaaramburu	A	Proba argazkia ijoyzeko berria	esletua	Orokorra	proba	1	30	2017-05-22
52	aaaramburu	A	Proba argazkia ijoyzeko berria	esletua	Orokorra	a	4	0	2017-05-30
53	a	A	Proba explorer	esletua	Argia	a	5	30	2017-05-30
53	a	A	Proba explorer	esletua	Argia	proba	0	18	2017-05-22
63	a	A	Proba	berria	Orokorra	a	0	30	2017-05-25
63	a	A	Proba	berria	Orokorra	a	1	15	2017-06-02
63	a	A	Proba	berria	Orokorra	a	0	30	2017-05-25
63	a	A	Proba	berria	Orokorra	a	50	30	2017-06-01
63	a	A	Proba	berria	Orokorra	a	50	0	2017-05-31
63	a	A	Proba	berria	Orokorra	proba	4	15	2017-06-02

Filtratu Lan id Filtratu Username Filtratu Saila Filtratu Langilea Filtratu Lan Eguna

Denbora (h) Denbora (min) Previous 1 2 3 4 5 ... 8 Next

Showing 1 to 10 of 74 entries

29 irudia. Datuak esportatzeko leihoa.

5.1.8 Lan prebentiboak sortu

Mantentze-kudeaketa bi arlo nagusitan banatzen da: mantentze-zuzentzailea eta prebentziozko mantentzea. Lehena, izenak dioen moduan, jada hauteman diren akatsak zuzentzeko lanari deritzo. Adibidez, komunean sortutako itoginak konpontzea edo gela batean pizten ez den bonbila aldatzea. Bigarrena berriz, aldizka egin beharreko lanari deritzo. Adibidez, urtean bitan ateen sarrailak koipeztatzea, edo, ikasturtea hasi aurretik, ordenagailuak formateatzea. Beraz, orain arte egin duguna, abisuak bidaltzeko aukera, mantentze-zuzentzailea arloan sartuko genuke. Hau nahikoa ez denez, mantentze prebentiborako abisuak sortzeko aukera garatu dugu. Hau abisu zuzentzaileak bidaltzeko leihoaren oso antzekoa da (ikus 5.8 irudia). Baina, kasu honetan, aukera gehigarri bat edukiko dugu, nahi adina data sartzeko aukera. Modu horretan, data bakoitzeko abisu prebentibo bat sartuko dugu. Horrela, lan prebentiboa egin beharreko egunean, esleitutako arduradunari mezu bat agertuko zaio aplikaziora sartzen denean, lana egin behar duela abisatuz (ikus 5.9 irudia). Bertan sakatuz gero, abisuaren xehetasunak ikusteko leihora bideratuko da.

← Atzera **Abisu prebentiboa sortu** Saioa itxi

Prebentiboa aaramburu

F -1 1 Normala

admin Tornua garbitu eta koipeztatu.

Tornua garbitu eta koipeztatu behar da.

2017/01/01

2017/04/01

2017/08/01

2017/12/01

Data berria

30. irudia. Abisu prebentiboak sortzeko leihoa.



31 irudia. Abisu prebentiboen jakinarazpena.

5.1.9 Erabiltzaileak sortu

Datu-basean sartu beharrik gabe, erabiltzailea sortzeko aukera ere eskaini dugu aplikazioan. Erabiltzaileen kopurua oso handia ez denez, administratzaileak sortzea erabaki dugu. Era honetan, gainera, edozeinek kontu bat sortzea galarazten da, erabiltzaileen jarraipen kontrola erraztuz. Erabiltzailea sortzeko leihoa nahiko sinplea dela esan genezake (ikus 5.9 irudia). Zenbait datu sartzeko eremuak ditugu (izena, abizena, erabiltzaile-izena, pasahitza, telefonoa eta posta elektronikoa) eta baita erabiltzaile mota hautatzeko aukera botoiak ere. Erabiltzaile mota arduraduna bada, dagokion saila ere aukeratu beharko du.

← Atzera
Erabiltzailea sortu
⏻ Saioa itxi

Izena:

Abizenak:

Erabiltzaile izena:

Pasahitza:

Telefonoa:

Email:

Erabiltzaile mota:

Irakaslea

Arduraduna

Administratzailea

Gorde

32. irudia. Erabiltzaile berria sortzeko leihoa.

5.1.10 Erabiltzaileak editatu/ezabatu

Erabiltzaileak sortzeaz gain, ikastaro berrirako interesgarria izan daiteke erabiltzaileak editatu edota ezabatzea. Horrela, menu nagusiko *Erabiltzailea editatu* sakatuz gero, erabiltzaile zerrenda bat agertuko zaio administratzaileari, erabiltzaile izena eta mota erakusten dituena (ikus 5.11 irudia). Bertan, izenaren arabera bilaketa egiteko aukera izango da. Zerrendako erabiltzaile bat hautatuz gero, *Erabiltzailea sorturen* oso antzekoa den leihoa agertuko da, baina, kasu honetan, datuak beteta dituena. Pasahitzaren atala, noski, ez da agertuko. Horrela, hutsik utzi ezker, pasahitza ez da aldatuko, baina berria sartzen badugu, eguneratu egingo da. Gainera, beste botoi bat ere eskuragarri edukiko dugu, ezabatzeko botoia hain zuzen. Sakatuz gero, erabiltzailea ezabatu eta berriz ere erabiltzaileen zerrendara itzuliko gara.

← Atzera
Erabiltzailea hautatu
⏻ Saioa itxi

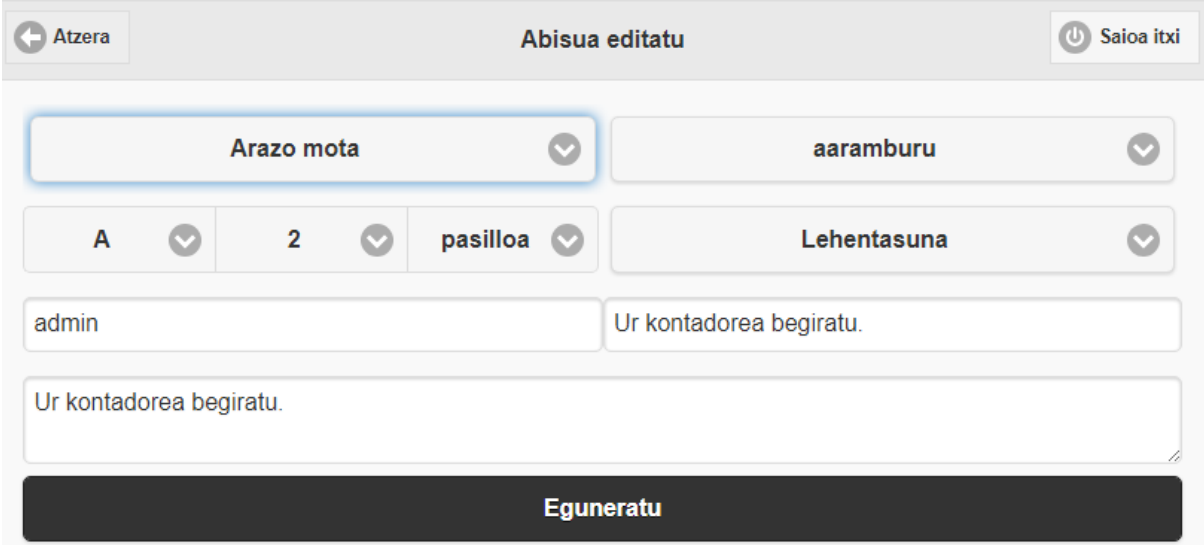
✕

informatika - Irakaslea	➤
adminis - Aministratzailea	➤
galdaragintza - Irakaslea	➤
alexaritz - Arduraduna	➤

33 irudia. Erabiltzaile zerrenda.

5.1.11 Abisuak editatu

Noiz edo noiz gerta liteke bidalitako abisu batean daturen bat oker sartzea, ezjakintasunagatik edo nahastu egin garelako. Horregatik, administratzaileari aukera eman diogu edozein abisu editatzeko. Abisu baten xehetasunetara administratzaile moduan sartzen bagara, beheko aldean abisua editatzeko aukera ematen digun botoia edukiko dugu (ikus 5.12 irudia). Hau sakatuz gero, abisua bidaltzeko leihorearen oso antzekoa den beste bat agertuko zaigu. Kasu honetan ordea, aukerak abisuaren datuekin beteta egongo dira. Nahi dugun datua aldatuko dugu eta eguneratu botoia sakatuko dugu. Bestalde, ez dugu argazkia eguneratzeko aukera ezarri, ez baitzaigu iruditu argazki okerra bidaltzeko aukera dagoenik, bidaltzeko unean aurreikusteko aukera edukita.



34.irudia. Abisua editatzeko leihoa.

5.2 Arazoak

5.2.1 Segurtasun arazoak

Software-aplikazioa enpresa batentzako denez, segurtasunari garrantzi berezia eman diogu. Horrela, erabilitako teknologien inguruko informazioa eta segurtasuna bermatzeko eman beharreko pausoak bilatu ditugu interneten. Orduan, zenbait segurtasun arazo genituela konturatu gara eta, berandu izan aurretik, hauek ekiditeko soluzioak bilatu ditugu. Ondoren, aurkitutako arazoak eta hauei emandako soluzioak azalduko ditugu.

5.2.1.1 SQL Injection[28]

Hasiera batean, MySQL datu-basea atzitzeko honen oinarriko sententziak erabili ditugu. Esan bezala, interneten hainbat bilaketa egin ostean konturatu gara modu hau ez dela segurua eta SQL Injection delako erasoak jasateko arriskua dagoela.

SQL Injection erasoak kode arrotza txertatzean datua, software-akats bat aprobetxatuta. Akatsa SQL kodea sortzen edo erabiltzen duten aldagaiak modu desegokian aztertu eta filtratzeagatik sortzen da. Helburua, SQL arrotz hori exekutatzea da, datu-baseko datuak ezabatu, aldatu edo lortzeko. Ondoren, zenbait adibide azalduko ditugu, zertan datzan ulertzeko[29]:

""="" beti da egia

Erabiltzaile eta pasahitza sartzeko bi eremu ditugu. Ondoren, SQL sententzia bat exekutatzen da bertako datuekin, dagokion erabiltzailearen informazioa jasotzeko.

Username:

Password:

Example

```
uName = getRequestString("username");
uPass = getRequestString("userpassword");

sql = 'SELECT * FROM Users WHERE Name =' + uName + ' AND Pass
      =' + uPass + ''
```

Result

```
SELECT * FROM Users WHERE Name ="John Doe" AND Pass ="myPass"
```

35 irudia. Kode ahula.

Baina, eremuetan benetako erabiltzailea sartu ordez, ondorengo " or ""="" txertatzen badugu (ikus 5.11 irudia), sententzia hori beti egia izango da eta, ondorioz, erabiltzaile guztien informazioa jasoko dugu, berez informazio hori erabiltzaile arruntaren esku ez dagoelarik.

User Name:

Password:

The code at the server will create a valid SQL statement like this:

Result

```
SELECT * FROM Users WHERE Name ='' or ''='' AND Pass ='' or ''=''
```

36 irudia. Sortuko den SQL sententzia.

Aldi bereko SQL sententziak

Adibide honetan eremu bakarra izango dugu, erabiltzaile bat bilatzeko id-a hain zuzen. Horrela, erabiltzaile-id egoki bat sartzen badugu, id horri dagokion erabiltzailearen informazioa berreskuratuko dugu. Baina, id-az gain, ; *DROP TABLE Suppliers* sartzen badugu, datu-baseko SELECT sententzia exekutazeaz gain, Suppliers taula ezabatzeko sententzia ere exekutatu da. Modu honetan, ezagutzen diren taulen izenak erabiliz, edozeinek taula hauek ezabatu ditzake, izugarritzko kalteak sortuz.

Example

```
txtUserId = getRequestString("UserId");  
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

And the following input:

User id:

Result

```
SELECT * FROM Users WHERE UserId = 105;
```

37 irudia. Espero den sententzia zuzena.

Aldi bereko SQL sententzietan oinarritutako SQL Injection erasoa

User id: `105; DROP TABLE Suppliers`

The valid SQL statement would look like this:

Result

```
SELECT * FROM Users WHERE UserId = 105; DROP TABLE Suppliers;
```

38 irudia. Aldi bereko sententzia, kode arrotza exekutatuko diguna.

Beraz, hau ekiditea ezinbestekoa izango zaigu, izugarritzko kalteak sortu baititzake. Ekiditeko, *Prepared Statement*[30]delakoak erabiltzea erabaki dugu. *Prepared Statement*-a antzekoak edo berdinak diren SQL sententziak behin eta berriz eraginkortasun handiz exekutatzeko ahalbidetzen duen ezaugarria da. Honela funtzionatzen dute sententzia horiek:

Prestatu: SQL sententzia eredu bat sortzen da, eta datu-basera bidaltzen da. Balioak ez dira zehazten, horren ordez ? ikur bat bidaltzen da.

Datu-baseak itzuli, konpilatu eta kontsulta optimizazioa burutzen du SQL ereduan, azkenik emaitza gordetzeko, exekutatu gabe.

Exekutatu: Ondoren, aplikazioak parametroen balioak lotzen ditu eta sententzia exekututzen da. Aplikazioak behin eta berriz exekutatu dezake sententzia, balio ezberdinekin.

Modu horretan, SQL injection-en kontra babesten da, sintaxia analizatzeko denbora murrizten da, kontsultaren prestaketa behin bakarrik egiten baita (nahiz eta sententzia hainbat aldiz exekutatu) eta zerbitzarirako banda zabalera txikiagotzen da, parametroak soilik bidaltzen direlako eta ez kontsulta osoa.

5.1.1.2 Localstorage/Session

Bigarren segurtasun-arazoa *localStorage*[31]delakoarekin sortu zaigu. Izan ere, erabiltzaile batek modu errazean bertako balioak alda ditzake. Horrela, hasiera batean *localStorage* delakoan gordetzen genuen, adibidez, erabiltzailea administratzailea zen edo ez eta, horren arabera, funtzionalitate batzuk edo besteak bistaratzen ziren. Baina, modu horretan, administratzailea ez zen erabiltzaile batek *localStorage* manipulatzeko administratzailea zela esan zezaken. Horregatik, zerbitzari aldeko *phpk* eskaintzen digun *\$_SESSION* aldagaia erabiltzea erabaki dugu eta, *localstoragen* gordetzen dugun informazio oro, arrotz moduan tratatzea.

5.1.1.3 Argazkien igoera

Beste segurtasun arazo bat argazkiak igotzeko aukerarekin sortu daiteke. Izan ere, erabiltzaile batek irudia igo ordez, script bat igo lezake zerbitzarira, irudiaren formatua simulatuz, ondoren hau exekutatu eta datuak lortu, aldatu edo ezabatzeko. Horregatik, igotako fitxategia benetan irudia dela ziurtatu dugu, zerbitzari aldeko *phpk*-k eskaintzen dizkigun ezaugarriak erabiliz, *upload.php* fitxategian[32][33]:

```
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);  
#jpg, jpeg , png edo git irudien formatuak bakarrik onartu  
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType !=  
"jpeg"  
&& $imageFileType != "gif" ) {
```



```
header('Content-Type: application/json');
echo json_encode(['erantzuna' => "Sorry, only JPG, JPEG, PNG & GIF
files are allowed.", 'ok' => "errorea"]);
$uploadOk = 0;
}
#uploadOk 0 den, hau da, errorea dagoen, konprobatu
```

39 kodea. Irudien formatua zuzena dela bermatzeko kodea.

Pathinfo funtzioa erabiliz, igotako fitxategiaren formatua lortzen dugu. Formatua okerra bada, aldagai batean gordeko dugu ez igotzeko abisua eta, ondorioz, argazkia ez da zerbitzarira igoko. Bestela, guk onartutako irudia bada, irudiaren kalitatea %25 murriztuko dugu, modu berean tamaina txikitzeko. Horrela, begi bistakoa ez den kalitate galera izango dugu eta, modu berean, zerbitzarira igotzeko denbora eta bertako memoria aurreztuko dugu, trinkoketa lortuz[34].

6

Probak eta hobekuntzak

Atal honetan, softwarea garatzen joan garen heinean egin ditugun proba nagusiak azalduko ditugu, aplikazioak zuzen funtzionatzen duela bermatzea ezinbestekoa zaigulako. Horretarako, softwareari hainbat eta hainbat proba egin dizkiogu, egoera eta funtzionalitate guztiak probatzeko, hainbat balio ezberdin emanaz. Horretaz gain, web-aplikazioa denez, sistema eragile eta nabigatzaile ezberdinetan probatu dugu eta, bai ordenagailu eta baita mugikorretarako ere garatu denez, dimentsio ezberdinetako pantailak dituzten gailuak probatu ditugu. Modu horretan, egoera eta baldintza ezberdinetan izan duen portaera aztertu dugu eta, beharrezkoa izan denean, zuzenketak egin ditugu. Ondoren, jasandako arazo nagusiak azaldu ditugu.

6.1 Lazy Loading[35]

Abisuak erakusteko, bi tolesgarri erabili ditugu. Horietan, bidali eta esleitutako abisuak ikusten ditugu, zerrenda moduan. Zerbitzaria alferrik ez kargatzearen, *lazy loading* delako teknika erabili dugu. Teknika hau, definizioak dioen moduan, elementuen karga atzeratzean datza, beharrezkoa izan arte. Hau da, gure kasuan, ez ditugu abisu guztiak zerrendan bistaratuko, pixkanaka kargatzen joango gara, ez baitugu momentuoro abisu guztiak ikusteko beharra izango. Hori lortzeko, erabiltzaileak tolesgarria beraino korritu duela kontrolatu beharko dugu, ondoren *ajax* eskaera baten bidez hurrengo elementuak kargatzeko.[36]

Baina, horrek, uste gabeko arazoa sortu zigun. Lehen kargan, bost elementu bakarrik kargatzen genituen. Guk probatutako gailu guztietan, zerrenda horrek pantailak erakusten zuena baino gehiago okupatzen zuen. Beraz, gure algoritmoak ondo funtzionatzen zuen, *scroll*-a egin behar baikenuen. Baina, pantaila oso handi batean proba eginda, konturatu ginen bost elementuko zerrenda pantailan bertan kabitzen zela, ondorioz, algoritmoa baliogabe uzten zuela. Izan ere, zerrenda pantailan bertan kabitzen zenez, ez zigun uzten leihoa korritzen eta, ondorioz, ez zitzaion beharrezko funtzioari deitzen, elementu berriak ezin kargatu utziz. Horregatik, hasieran erakutsiko ditugun elementu kopurua handitzea erabaki da, 15 elementura igoz. Modu horretan, zerrenda edozein pantailatan ez dela kabituko ziurtatuko dugu.

Beste arazo bat ere eduki dugu. Lehen kargan, askotan, abisuak desordenatuta ateratzen zitzaizkigun, arrazoirik gabe. Proba batzuk egin ostean, konturatu ginen orria kargatzerakoan *scroll*-arekin gauza arraroak egiten zituela, funtzioari deitu eta *ajax* eskaerak eginez, behar ez zirenean. Konpontzeko, funtzioa orria erabat kargatu eta ikusgarri dagoenean soilik aktibatzea erabaki dugu, *ajax* eskaera guztiak bukatutakoan.

6.2 Datables[37]

Datuak esportatzeko plugin honekin ere arazoak izan ditugu. Taulan datuak erakustea nahiko modu errazean lortu dugu. Funtzionalitateak gehitzen hasi garenean ordea, arazoak izan ditugu APIarekin. Zutabe bakoitzeko filtro aukera sortzeko adibidez, zutabe bakoitzeko eremua gehitzeko, APIari deitu behar genion. Deia egiteko, 6.1 irudian agertzen zaigun moduan, hiru modu eskaintzen dizkigu plugin-ak. Baina, hiru deiak probatu eta probatu ibili ginen eta ez genuen APIari zuzen deitzea lortu. Azkenean, hainbat ordu galdu ostean, konturatu ginen lehenago deitzen geniola APIari, taula hasieratzean hain zuzen. Horrela, taularen hasieraketa aldagai batean gorde eta ondoren aldagai hori erabili dugu deiak egiteko (ikus 6.2 irudia).

```
- $( selector ).DataTable();
```

```
- $( selector ).dataTable().api();
```

```
- new $.fn.dataTable.Api( selector );
```

40. irudia. Datatables pluginaren APIari deia egiteko modu ezberdinak..

```
t=$('#taula').DataTable({
    #kodea
});

t.columns().every( function () {
    #kodea
});
```

41. irudia. APIari egindako deia.

7

Jarraipena eta kontrola

Kapitulu honetan, proiektuan egindako lanaren jarraipenaren inguruko datuak azalduko ditugu. Hasieran egindako plangintzarekiko jasandako desbideratzeak aztertuko ditugu eta horien arrazoia azaltzen saiatuko gara.

7.1 Proiektuaren garapena

Nahiz eta plangintza erreal eta zuzena egiten saiatu, zenbait ataletan desbideratzeak eduki ditugu, bai atazak uste baino lehenago bukatu ditugulako, baita (eta hauek dira gehien interesatzen zaizkigunak) uste baino denbora gehiago pasa dugulako ere. Hala ere, esan beharra dago, ezinezkoa dela tamaina horretako proiektu batean plangintza % 100 zuzena egitea. 7.1 taulan ikusten den moduan, ataza bakoitzari aurreikusitako orduak eta benetan sartutako orduak aztertuko ditugu.

Ataza	Estimazioa	Erreal	Desbiderapena	
GAP	320:00	372:00	+52:00	% 16,2
Kudeaketa	50:00	47:00	-3:00	% 6
Plangintza	20:00	20:00	0:00	% 0
Jarraipena eta kontrola	15:00	17:00	+2:00	% 13,3
Bilerak	15:00	10:00	-5:00	% 33,3
Produktua	200:00	245:00	+45:00	% 22,5
Analisia	50:00	45:00	-5:00	% 10
Garapena	150:00	200:00	+50:00	% 33,3
Dokumentazioa	70:00	80:00	+10:00	% 14,3

Memoria	60:00	70:00	+10:00	% 16,7
Aurkezpena	10:00	10:00	0:00	%0

42 taula. Proiektuan estimatutako ordu eta benetako orduen arteko desbideraketak.

Ikusten dugunez, desbideratze nabarmenena garapenean izan dugu. Aurreikusitako 150 ordu izan ordez, azkenean 200 ordu inguru sartu behar izan ditugu. Beraz, % 33,3ko desbideratzea izan dugu, kontuan hartzekoa. Honen arrazoi nagusia, eskakizun-bilketetan izandako aldaketa izan da. Izan ere, abisu prebentiboen arloa ez zen hasiera batean oinarritzko eskakizun moduan zehaztu baina, jada implementazioa ia bukatuta genuela, aplikazioaren arlo garrantzitsua zela abisatu zitzaigun eta balio erantsi handia emango ziola aplikazioari. Horregatik, nahiz eta denboraz juxtu xamar ibili, implementatzea erabaki genuen.

Horrelako ustekabeak gertatu zitezkeela aurreikusi genuen eta, horregatik, ordu gehiago eskaini genizkion garapenari, baina, hala ere, esleitutako orduak motz geratu zaizkigu.

Bestela, orokorrean, nahiko estimazio zuzena egin dugula esan dezakegu. Azkenean, proiektu osoko estimazioan %16,2ko ordu desbideraketa izan dugu (nahiz eta batzuetan ordu gehiegi eta bestetan gutxiegi estimatu ditugun, benetako desbideraketa handiagoa izanez). Baina, horrek, ez du proiektuko mugarrietan aldaketa handirik suposatu, proiektu amaieraren aurreikuspena eta entrega dataren artean nahikoa denbora izan baitugu lanak amaitzeko.

7.2 Kalitatea

Proiektuan jasandako orduen desbideratzeaz gain, garatutako proiektuaren kalitatearen kontrola ere egin dugu. Plangintzan proiektuaren kalitatea bermatzeko ezarritako faktoreak banan bana aztertu ditugu, bai aplikazioak, bai memoriak, betetzen dituela frogatzeko.

7.2.1 Aplikazioaren kalitatea

Aplikazioaren kalitatea bermatzeko, hiru faktore nagusi zehaztu genituen:

- **Arintasuna:** Garatutako aplikazioaren funtzionamendua arina izatea lortu dugu. Bai gu, bai Tolosaldeako Institutuko irakasleak, ados gaude aplikazioan gabiltzala, ez dagoela erabiltzailearen esperientzia kaltetzen duen denbora karga luzerik. Denbora karga luzeena, abisua argazki eta guzti bidaltzeko unean daukagu, normala den bezala. Hala ere, gehienez 5MB-eko tamainako argazkiak igoko direnez, eta gaur egungo interneteko abiadura nahiko handiak direnez, argazkia igotzea ere nahiko azkarra da. Hori guztia, noski, internetarekin edo zerbitzariarekin arazorik ez daukagula suposatuta.
- **Interfazea erabilerraza:** Abiaduraz gain, interfazea intuitiboa eta erabiltzeko erraza izatea nahi genuen. Hau probatzeko, berriz ere aplikazioaren bezeroak, hau da, irakasleak, erabili dira. Ia azalpenik eman gabe izan dira gai aplikazioaren funtzionamenduak erabiltzeko. Beraz, esan genezake, faktore honek zehaztutako kalitate maila bete dugula.

- **Kode-garbitasuna:** Azken faktore hau gure kabuz baloratu dugu. Hala ere, uste dugu lortutako garbitasuna nahiko handia dela. Izan ere, kode osoa zuzen tabulatuta dago, dagozkion iruzkinak idatzita eta fitxategietan banatuta. Horretaz gain, bai aldagaiak, bai fitxategien izenak ere, ahal diren esanguratsuenak izatea saiatu gara. Beraz, txosteneko azalpenak (sekuentzia diagramak, gertaera fluxuak...) eta kodea edukita, kodea ulertzeko arazorik ez litzatekeela egongo uste dugu.

7.2.2 Memoriaren kalitatea

Luzera: Memoriaren tamainari dagokionez, 13000 hitz inguruko txostena egin dugu azkenean. Bertan, egindako plangintza, garapenean emandako xehetasunak eta proiektuaren jarraipen eta kontrola dokumentatu ditugu, eta amaieran ateratako ondorioak bildu ditugu. Beraz, esan genezake, bai tamainan, eta bai edukian, aurrez zehaztutako kalitate-maila betetzen duela memoriak.

Zuzentasuna: Memoriaren zuzentasunari dagokionez, zehaztutako pausoak bete ditugu. Memoria osoa XUXEN 5[38] zuzentzailetik pasa da. Honetaz gain, bai tutoreari eta baita familiakoei ere pasa zaio txostena, xuxenek egindako zuzenketa ortografiko eta gramatikalaz gain, beste zuzenketa egiteko. Terminologiari dagokionez, Mozilla nabigatzaileko Euskalbar[39] gehigarria erabili da, esan bezala, Elhuyar Hiztegia eta Euskalterm terminologia-banku publikoa erabiltzeko.

7.3 Arriskuak

Aurreikusi genituen arriskuetatik zenbait jasan ditugu. Ondoren, aurreikusi eta jasan ditugun arriskuak aztertu ditugu eta aurrera eramateko hartutako erabaki eta soluzioak azaldu.

- **Plangintza okerra egitea:** Orokorrean, plangintza nahiko zuzena egin dugula esan genezake. Hala ere, arazo nagusi bat eduki dugu. Arriskuek sortutako atzerapenak aurreikusi genituen plangintza, eta ordu batzuk gehiago esleitu genizkion garapenaren atalari, badaezpada. Baina, ondoren azalduko dugun moduan, hau ez da nahikoa izan, motz geratu baikara. Horrek, azkenean, proiektuaren hainbat atal atzeratzea ekarri digu. Beraz, hurrengo proiektuetarako, arriskuak ebazteko ordu gehiago esleitzea komeniko zaigu.
- **Plangintza ez jarraitzea:** Atzerapenak kenduta, proiektuaren plangintza jarraitu dugula esan genezake. Honetan, goizetan Tolosaldea Institutura joan beharrak lagundu egin digu, lan-ordutegia jarraitu dugulako.
- **Eskakizunen aldaketa:** Aplikazioak bete beharreko eskakizunen bilketan ere aldaketak izan dira. Lehenago azaldu bezala, mantentze prebentiborako funtzionalitateak izan dira desbideratze nagusia eragin digutenak. Horrelako ustekabeak gertatu zitezkeela aurreikusi genuen, eta horregatik ordu gehiago eskaini genizkion garapenari, baina, hala ere, esleitutako orduak motz geratu zaizkigu. Hala ere, egindako plangintzan proiektuaren bukaera proiektuaren entrega baino bi hilabete lehenago ezarri zenez, denbora izan dugu entrega egin aurretik abisu-prebentiboak inplementatzeko.

- **Aplikazioaren garapenean arazoak:** Plangintza osatzerakoan, garapenean atzerapenak edukiko genituela hartu genuen kontuan. Hala ere, ezarritako ordu kopurua ez da nahikoa izan eta, horrek, proiektu osoan atzerapenak eragin dizkigu. Beraz, hurrengo baterako implementazioan sortu daitezkeen atzerapenekin kontu handiagoa izan beharko dugu, batez ere, kasu honetan bezala, teknologia berriak erabiltzen baditugu.
- **Aplikazioaren bateragarritasuna:** Aplikazioa ezarritako gailu eta nabigatzaileetan funtzional izatea lortu da.
- **Datuen galera:** Datuekin ez dugu inolako arazorik eduki. Badaezpada, ezarritako babes-kopia guztiak egin dira, egunaren amaieran. Kasu batean, github-en aurreko bertsio bat jaitsi behar izan zen, ondoren azalduko dugun elementu baten eguneratzeak sortutako arazoa identifikatzeko. Baina, horretaz gain, ez dugu babes-kopiak erabiltzeko beharrik eduki.
- **Erabilitako software-elementuen eguneratzeak:** Aurreikusi genuen gerta zitekeela erabilitako software-elementuren bat eguneratzea, eta hala gertatu da. Datak sartzeko egutegia eskaintzen digun *datebox* pluginak, egun batetik bestera, funtzionatzeari utzi zion. Horrela, aplikazioaren aurreko bertsioa aztertu ondoren, konturatu ginen liburutegia eguneratu egin zela. Kasu honetan, garapen-fasean gertatu zaigu hau eta CSS eta Javascript helbideak eguneratzea soilik izan da. Beraz, ez digu atzerapen garrantzitsurik sortu, baina, esan liteke, etorkizunean adi egoteko abisua eman digula.

8

Ondorioak

Atal honetan, proiektuan zehar atera ditugun ondorioak azalduko ditugu, lortutako emaitzak aztertuz eta baloratuz. Gainera, etorkizunerako aurreikusitako aukerak ere azalduko ditugu.

8.1 Lortutako emaitza

Mantentze-kudeaketarako aplikazioa garatzea lortu dugu, hasieran eskatutako eginkizunak beteta. Gure ustez, lan txukuna egin da, aplikazioak zuzen funtzionatzen baitu, intuitiboa baita eta eskatutako funtzionalitateak baititu. Hala ere, bukaerako benetako iritzia bezeroena izango da, beraiek baitira aplikazioa egunero erabiliko dutenak.

Horrela, bezeroen iritzia jasotzeko, aurkezpen txiki bat egin zaie, aplikazioa aurkeztu asmoz. Aurkezpenean, gainera, saioa hasteko kontu bat eman zaie, bakoitzak bere mugikorrean aplikazioa probatu dezan. Horrela, abisuak bidali dituzte, argazki eta guzti (jolasean, elkarren argazkiak bidaliz), abisuak ikusi, lanaren inguruko datuak osatu... Horretaz gain, guk, ordenagailua erabili dugu administratzailearen funtzionalitateak erakusteko, nahiz eta mugikorrean eskuragarri izan, ordenagailuan maneiatzeko pentsatuta baitaude. Datuak esportatzea da horren adibide, ez baita oso praktikoa izango, datuak Excel fitxategi batera esportatu, eta mugikorrean datu horiek maneiatzea. Azalpena behin amaituta, bezeroen iritzia oso positiboa izan da. Gailu ezberdinetan funtzionatzea, abisua mugikorretik argazki eta guzti bidaltzeko aukera, abisuen egoera adierazteko koloreak erabiltzea... Horiek guztiak mugikorretik atzitzeko aukera edukitzea eskertu dute, modu sinple batean gainera, alferrikako aukerak ekidinez. Gainera, eta nahiz eta asko landu ez izan, interfazearen sinpletasuna ere eskertu dute. Beraz, bezeroen iritzi positiboa ikusita, lan txukuna egin dugula esan dezakegu.

8.2 Ebaluazio pertsonala

Proiektu honen bidez, zenbait teknologia berri erabiltzen ikasteaz gain, neurri handiko proiektuak kudeatzen ikasi dut eta, baita benetako bezeroekin elkarrekintza nolakoa den ikusi ere. Proiektua garatzen hasi aurreko momentuetan, errespetua ematen zidan enpresa batean lan egiteak, baina, behin lanean hasita, giroa oso ona izan da. Horrek ere, noski, lagundu dit proiektuko lana modu arinago batean ematen. Izan ere, nahiz eta proiektuko lana, hau da, software bat garatzea, gustatu, momentu jakin batzuetan motibazio maila txikitu egin zait. Adibidez, atal baten diseinua garbi ez nuenean, inplementazioren batean errorea ezin

konpondu nenbilenean... Horrelakoetan, ataza hori utzi eta beste zerbaitekin hasteak lagundu egin dit, burua argitu eta ondoren beste ikuspegi batekin itzultzeko. Beraz, esan beharra dago proiektuan zehar, orokorrean, motibatuta eta oso eroso sentitu naizela. Ondoren, proiektuan zehar ateratako ondorio batzuk azaldu ditut:

- Internet oso baliagarria da: Oso ohikoa eta begi-bistakoa izango da, baina internet oso baliagarria da software bat garatzeko unean. Garapenean sortu zaizkidan edozein arazo eta dudak ebazteko internet erabili dut. Galdera zuzen egin ezker, kasu gehienetan soluzioa edo erantzuna erraz aurkitzen da. Batzuetan, nahiz eta erantzuna zehatz-mehatz behar dudana ez izan, erantzun horretatik abiatu eta nire arazoari soluzioa eman diot. Horrela, etengabe erabili dut internet laguntzaile gisa eta, benetan, oso baliagarria suertatu zait.
- Benetako bezeroak: Baliagarria izan zaidan beste gauza garrantzitsua, benetako bezeroekin izandako esperientzia izan da. Egia da, graduan zehar, hainbat proiektu egin ditugula bezeroen eskakizunak simulatuz, baina, benetako bezeroekin bildu naizenera arte, ez naiz konturatu hauei eskaini beharrezko arretaz. Izan ere, kasu honetan, askotan gertatzen den moduan, bezeroek software-garapenaren inguruan zuten ezagutza maila oso baxua zen. Horregatik, ezarritako denboran egin zitezkeen eta alde batera utzi beharreko funtzionalitateak, horren zergatiak, zailtasunak... modu sinple batean azaltzea ezinbestekoa izan da. Gainera, nik neuk proposamenak egitea ere garrantzitsua dela ohartu naiz, bezeroei momentuan bururatzen ez zaizkien edo lehenetsiak direla pentsatzen duten eskakizun eta funtzionalitateak identifikatzeko.
- Plangintza: Hau ere begi-bistakoa da, baina oso garrantzitsua izango zaigu proiektuaren garapenaren plangintza bat sortzea, are gehiago, tamaina handikoa bada. Plangintzak, eman beharreko urratsak finkatuko dizkigu, bete beharreko helburuak ezarri. Horrela, garapenean hasten garenean, eskema bat finkaturik edukiko dugu eta, jarraituz gero, lana erraztu eta arinduko digu. Bestalde, orduak aurreikusiz, ataza bakoitzean pasatako orduak konparatu ditzakegu, atzerapenak aurreikusi eta ekiditeko.
- Dokumentazioa: Plangintzarekin batera, egindako lana dokumentatzea ere garrantzitsua izan da. Tamaina handiko proiektuetan, ezinezkoa izango zaigu egindako lanaren kontrola izatea, dokumentazioa azkenerako uzten badugu. Beraz, dokumentazioa egunean eramatea beharrezkoa izango da, proiektuaren jarraipen egokia izateko. Izan ere, ez dugu lehenago hain tamaina handiko proiekturik garatu eta, ondorioz, horiek dokumentatzeko memoriak ere ez dira oso luzeak izan. Baina, kasu honetan, garapenean zehar egindako lana handia izan da eta beraz, memoria ere modu berean luzatu da. Ondorioz, memoriari ere ohikoa baino denbora gehiago eskaini diot. Hasieratik memoriaren eskema bat finkatzeak lagundu egin dit beharrezko dokumentazioa biltzen eta, ondoren, azaltzen.
- Bertsio-kontrola: Software inplementazioa egitean, edozein aldaketa txikik aplikazio osoa baliogabetu dezake. Horregatik, ezustekoak ekiditeko, bertsio-kontrola erabiltzea gomendagarria uste dugu. Horrek, gainera, hainbat garatzailearen artean egindako garapena modu errazean bateratzea eskaintzen du. Hau gutxi balitz, babes-kopia moduan ere funtziona lezake kodeak, arazoren bat badaukagu, aplikazioaren kodea berreskuratzeko aukera eskaintzen digulako.

Orokorrean, esan nezake, oso baliagarria izan zaidala gradu amaierako lan hau garatzea, are gehiago, enpresa batean lan eginda. Modu honetan, lan-mundutik gertuago egongo den esperientzia neureganatu dut eta, nahiz eta hau azalekoa izan, lan-munduan sartzen

garenerako oinarri bat egiten lagundu didala esan dezaket. Horretaz gain, noski, teknologia berriak erabiltzen ikasi dut eta, garrantzitsuagoa dena, proiektu bat neure kabuz aurrera eramatea.

8.3 Etorkizunerako aukerak

Sortu dugun aplikazioarekin oso gustura gaude, baina, halere, zenbait hobekuntza egin daitezkeela uste dugu, batez ere irismenetik kanpo geratu zaizkigun ezaugarriak.

Horretarako aukera sortu zaigu gainera. Izan ere, proposamen bat jaso dugu. Eusko Jaurlaritzak sustatutako UrratsBat[40]izeneko programaz ari gara. Programa horren helburua, berez, LHko ikasleei enpresa sortzen laguntzea da, baina, kontsulta egin ostean, guretzako ere baliagarria izan daitekeela ikusi dugu. Horren bidez, diru laguntza bat jasotzeaz gain, Tolosaldea ikastetxean laguntza teknikoa eta garapenerako lantokia ere izango genuke. Beraz, laguntza hori aprobeztatu dezakegu softwarea gehiago garatzeko (nahiz eta oraindik ezer zehaztuta ez eduki). Aukera horri heltzen badiogu, ondorengo ezaugarriak gehitzeko aukera ikusten dugu:

- Auditoria: Aukerari helduz gero, Ikastetxean gaudela aprobeztatu dezakegu bertan egiten diren auditorietara aplikazioa aurkezteko. Bertan, aplikazioak zenbait segurtasun eta kalitate maila betetzen dituen hala ez aztertuko da, eta ondoren gure lana izango da jasotako arazo horiei konponbidea ematea.
- Posta elektronikoen lotura: Oso interesgarria izango litzateke aplikazioaren abisuek posta elektronikoen lotura. Horrela, ez litzateke aplikazioan sartu beharrik izango, zuzenean postan abisuen xehetasunak ikusteko aukera edukiko baikenuke.
- Hizkuntzak: Aplikazioa hizkuntza ezberdinetan erabiltzeko aukerak, balio erantsi handia emango lioke. Horregatik, interesgarria izango litzateke aplikazioa Euskaraz gain, Ingelesean eta Gaztelaz erabiltzeko aukera izatea.
- Aplikazio natiboa/hibridoak: Egia da aplikazioa hainbat gailutan funtzional izatea lortu dugula. Hala ere, pauso bat gora egingo genuke kalitate mailan, web aplikazioa atzitu beharrean, smartphone eta tabletentzako aplikazio propioak sortuko bagenu. Horrela, aplikazio horiek eskaintzen dizkiguten abantailak izango genituzke, hala nola, *push* notifikazioak, aplikazioa konexiorik gabe erabiltzeko aukera, gailuen *API*a atzitzeko aukera... Horretaz gain, errendimendua ere hobetu egingo litzateke.
- Itxura: Aplikazioaren itxura moldatzea interesgarria izango litzateke, izan ere, aplikazioa erabilgarria izatea bezain garrantzitsua da erabiltzailearentzako erakargarria izan den interfazea sortzea. Beti ere sinpletasun eta garbitasuna mantenduko dugu, baina aplikazioari identitate propioa emango lioke.

Beraz, aukera oso interesgarria suertatu zaizu proiektuko lanarekin aurrera jarraitzeko.

Bibliografia

- [1] “Informatika fakultatea. Gradu amaierako lanari buruzko arautegia”
http://www.ehu.es/documents/340468/2334257/GAP_araudia.pdf
- [2] “Push notifikazioak”
https://en.wikipedia.org/wiki/Push_Notification
- [3] “Github”
<https://github.com/>
- [4] “2016/17 IKASTURTEA-GRADU AMAIERAKO PROIEKTUAK-Defentsa eta kalifikazio prozedura eta egutegia”
http://www.ehu.es/documents/340468/2334228/Calendario_calendario+de+defensa_PFG_+eus.pdf
- [5] “ADDIn GRAL bat osatzeko gida”
http://www.ehu.es/documents/340468/2334264/ADDI_gida.pdf
- [6] “Responsive web-diseinua”
https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable
- [7] “Gradu Amaierako Lana. Software Ingeniaritza espezialitatea”
<http://www.ehu.es/documents/340468/2334257/SI+espezialitatea>
- [8] “Mozilla Firefox”
<https://www.mozilla.org/es-ES/>
- [9] “Google Chrome”
<https://www.google.es/chrome/browser/desktop/index.html>
- [10] “Internet Explorer”
<https://www.microsoft.com/es-es/download/internet-explorer.aspx>
- [11] “Safari”
<https://www.apple.com/es/safari/>
- [12] “StatCounter, Nabigatzaileen merkatu-banaketa”
<http://gs.statcounter.com/browser-market-share#monthly-201706-201706-bar>
- [13] “HTML5”
<https://www.w3.org/TR/html5/>
- [14] “PHP”
<http://php.net/manual/es/intro-what-is.php>
- [15] “Javascript”
<https://www.javascript.com/>
- [16] “CSS”
<https://www.w3.org/Style/CSS/Overview.en.html>
- [17] “jQuery Mobile”
<https://jquerymobile.com/>
- [18] “MySQL”
<https://www.mysql.com/>
- [19] “XAMPP”
<https://www.apachefriends.org/es/index.html>
- [20] “Hostinger”
<https://www.hostinger.es/>
- [21] “Notepad++”
<https://notepad-plus-plus.org/>
- [22] “Git”
<https://git-scm.com/>

- [23] “Brainstorming”
https://es.wikipedia.org/wiki/Lluvia_de_ideas
- [24] “Mantis Bug Tracker”
<http://www.mantisbt.org/>
- [25] “Veedor, mantentze-kudeaketa eta produkzioaren kontrolerako plataforma”
<http://veedor.es/caracteristicas/>
- [26] “PHP, \$_Session aldagaia”
<http://php.net/manual/en/reserved.variables.session.php>
- [27] “Github plataforma, aplikazioaren kodea”
<https://github.com/Alexaritz/GAP>
- [28] “Wikipedia, SQL injection”
https://en.wikipedia.org/wiki/SQL_injection
- [29] “W3Schools, SQL injection”
https://www.w3schools.com/sql/sql_injection.asp
- [30] “Mysql, Prepared statement”
<https://dev.mysql.com/doc/refman/5.7/en/sql-syntax-prepared-statements.html>
- [31] “W3Schools, HTML5 localStorage”
https://www.w3schools.com/html/html5_webstorage.asp
- [32] “Ajax image upload, PHP and JQuery”
<https://www.formget.com/ajax-image-upload-php/>
- [33] “W3Schools, PHP5 file upload”
https://www.w3schools.com/php/php_file_upload.asp
- [34] “Compress image file size while uploading”
<http://blog.clonesinfo.com/how-to-reduce-compress-image-file-size-uploading-using-php-code/>
- [35] “Wikipedia, Lazy loading”
https://en.wikipedia.org/wiki/Lazy_loading
- [36] “Jquery Mobile Trick, Infinite scrolling, do it yourself!”
<https://jqmtricks.wordpress.com/2014/07/15/infinite-scrolling/>
- [37] “Datatables, table plug-in for jQuery”
<https://datatables.net/>
- [38] “Xuxen 5 zuzentzaile ortografikoa”
<http://xuxen.eus/eu/xuxen5>
- [39] “Euskalbar, hainbat hizkuntzetarako itzultzaile eta hiztegia”
<http://euskalbar.eu/>
- [40] “Urratsbat programa”
<http://ekingune.tknika.net/urratsbat>

A ERANSKINA

Aplikazioa martxan jartzeko, *php* eta *mysql*rekin lan egiteko gai den zerbitzaria beharko dugu. Behin zerbitzaria daukagula, ondorengo pausuak eman beharko ditugu web-aplikazioa martxan jartzeko.

Lehenik eta behin, aplikazioa bera jaitsi beharko dugu. Horretarako, <https://github.com/Alexaritz/GAP> helbidera joan beharko dugu. Bertan, aplikazioaren kodeaz gain, datu basea ere jaitsiko dugu.

Behin kodea eta datu basea ditugula, kodean aldaketa txiki batzuk egin beharko ditugu. Aplikazioaren PHP karpeta sartu eta `dbcon.php` izeneko fitxategia irekiko dugu, testu editore batekin. Fitxategi hau datu basearekin konexioa ezartzeko beharrezkoa da. Bertan, A1 irudiak ikusi dezakegun kodea ikusiko dugu:

```
<?php
header("Content-Type: text/html;charset=utf-8");
$server = "ZERBITZARIAREN HELBIDEA";
$usuario = "DATU BASEKO ERABILTZAILA IZENA";
$password = "PASAHTZA";
$db = "DATU BASEAREN IZENA";

$mysqli = new mysqli ($server,$usuario,$password, $db);
if ($mysqli->connect_error) {
    printf("Connection failed: " . $mysqli->connect_error);
}
$tilde = $mysqli->query("SET NAMES 'utf8'");
?>
```

A 1 irudia. dbcon.php fitxategia.

Fitxategian, letra lodiz agertzen zaizkigun datuak bete beharko ditugu, gure zerbitzariaren arabera.

Hau eginda, html orriak irekiko ditugu. Hauetan, *javascript* zatiaren hasieran dagoen zerbitzariaren helbidea aldatu beharko dugu, gure zerbitzariaren helbidea jarritz.

```
var server="ZERBITZARIA"
```

Horrela, aplikazioa zerbitzarira igotzeko prest izango dugu. Fitxategiak irudiko egituran igo beharko ditugu, edo, bestela, html orrietan karpeten helbidea aldatu.

📁 CSS	01/06/2017 17:59	Carpeta de archivos	
📁 IMAGES	10/07/2017 16:28	Carpeta de archivos	
📁 less	01/06/2017 17:59	Carpeta de archivos	
📁 PHP	31/08/2017 18:29	Carpeta de archivos	
🌐 abisuaBidali	10/07/2017 16:28	Chrome HTML Do...	12 KB
🌐 abisuaEditatu	10/07/2017 16:28	Chrome HTML Do...	11 KB
🌐 amaitu	01/06/2017 17:59	Chrome HTML Do...	1 KB
🌐 bidalitakoAbisuak	01/06/2017 17:59	Chrome HTML Do...	4 KB
🌐 datuak	01/09/2017 16:09	Chrome HTML Do...	9 KB
🌐 erabEditatu	01/09/2017 21:53	Chrome HTML Do...	6 KB
🌐 erabiltzaileaEditatu	01/09/2017 19:22	Chrome HTML Do...	3 KB
🌐 erabiltzaileaSortu	03/09/2017 15:01	Chrome HTML Do...	4 KB
🌐 index	10/07/2017 16:28	Chrome HTML Do...	3 KB
🌐 lanXehetasunak	28/08/2017 12:49	Chrome HTML Do...	21 KB
📄 mantentzedb.sql	01/06/2017 17:59	Archivo SQL	22 KB
🌐 menua	09/08/2017 10:44	Chrome HTML Do...	4 KB
🌐 nireAbisuak	26/07/2017 10:40	Chrome HTML Do...	10 KB
🌐 prebentiboaSortu	10/07/2017 16:28	Chrome HTML Do...	13 KB

A 2 irudia. Aplikazioaren egitura zerbitzarian.

Behin aplikazioa zerbitzarian daukagula, datu basea instalatu beharko dugu.

Horretarako, gure zerbitzarira datu basea inportatuko dugu, kodearekin batera jaitzitako datu basea hain zuzen. *Phpmysqladmin* erabiltzen badugu, A3 irudian ikusi dezakegun leihoa aurkituko dugu. Gainkaldean agertzen zaigun *Importar* aukera sakatu, datu basea hautatu, eta *Continuar* botoia sakatuko dugu. Guztia zuzen joan bada, datu basea inportatu egingo eta aplikazioa martxan jartzeko moduan egongo gara.

A 3 irudia. *Phpmysqladmin* softwarean datu basea inportatzeko leihoa.

Aurreko pausoak zuzen egin baditugu, dagoeneko aplikazioan sartzeko aukera izango dugu. Horretarako, datu basean administratzaile kontu bat gorde dugu, aplikazioaren kudeaketa egiteko. Kontuaren datuak ondorengoak dira:

Erabiltzaile izena: admin

Pasahitza: admin

Beraz, kontu horrekin aplikazioan sartu eta behar ditugun erabiltzaileak sortzea besterik ez zaigu geratuko eta, hori eginda, aplikazioa martxan izango dugu.