

Gradu Amaierako Lana

Informatika Ingeniaritzako Gradua

Software Ingeniaritza

Sistema informatiko baten diseinua eta implementazioa SOS Arrazakeria gobernuz kanpoko erakunderako

Joane Mannion Agirre

Zuzendariak
Olatz Arbelaitz
Ana Arruarte

2021eko ekaina

Laburpena

Dokumentu honetan, SOS Arrazakeria gobernuz kanpoko erakundearentzat garatutako aplikazioari buruzko Gradu Amaierako Lanaren (GrAL) memoria bilduko da.

Memoria honetan proiektuaren garapenarekin zerikusia duen guztia azaltzen da, hau da, proiektuaren kudeaketa, erabili diren teknologiak, aukeratutako arkitektura, proiektuaren garapena egindako probak, proiektuaren kontrola eta jarraipena, proiektuaren ondorioak eta bibliografia.

SOS Arrazakeriarentzat sortu den web-aplikazioaren helburua, gobernuz kanpoko erakundeak dituen erabiltzaileen gestioa erraztea da. Erabiltzaile guztien informazioa bateratu, eta hauen kasuen jarraipen bat egitea.

Proiektu hau EHUko Informatika Ingeniaritzako Graduako ikasle batek egingo da SOS Arrazakeria gobernuz kanpoko erakunderako.

Gaien aurkibidea

Gaien aurkibidea	iii
Irudien aurkibidea	v
Taulen aurkibidea	vii
1 Sarrera	1
2 Proiektuaren kudeaketa-plana	3
2.1 Helburuak	3
2.1.1 Proiektuaren helburu zehatzen deskribapena	3
2.2 Plangintza	4
2.2.1 LDE	4
2.2.2 Atazen deskribapena	5
2.2.3 Atazen garapen-estimazioa	6
2.2.4 Gantt Diagrama	6
2.3 Informazio eta komunikazio sistemak	7
2.3.1 Informazio-sistema	7
2.3.2 Komunikazio-sistema	8
2.4 Arriskuak eta kalitatea	8
2.4.1 Arriskuen analisia	8
2.4.2 Kalitatearen kudeaketa	8
2.5 Interesatuak	9
3 Teknologiak eta metodologia	11
3.1 Proiektuan erabilitako teknologiak	11
3.1.1 Bezeroaren aldeko teknologiak	11
3.1.2 Zerbitzariaren aldeko teknologiak	12
3.1.3 Garapenerako teknologiak	13
3.2 SCRUM metodologia	13
4 Analisia eta diseinua	15
4.1 Arkitektura	15
4.1.1 MVC Arkitektura	15
4.1.2 MVC Arkitektura aplikazioan	15
4.2 Erabilpen-kasuak	16
4.2.1 Aktoreak	17
4.2.2 Sprintak	17

4.2.3	Erabilpen-kasuen diagrama	28
4.3	Gertaera-fluxuak eta sekuentzia-diagramak	28
4.4	Datu-basearen diseinua	37
4.4.1	Deskribapena	37
4.4.2	Diagrama	38
4.4.3	Taulak	39
4.5	Domeinuaren eredia	41
5	Soluzioaren garapena	43
5.1	Kodearen egitura	43
5.1.1	Bistak	43
5.1.2	Kontrolatzaileak eta modeloak	44
6	Testak	47
6.1	Test unitarioak eta integrazio-testak	48
6.2	E2E testak	52
7	Hedapena	55
7.1	Domeinua eta SSL ziurtagiria	55
8	Jarraipena eta Kontrola	57
8.1	Denboraren kudeaketa	57
8.2	Komunikazioa	57
9	Ondorioak	59
9.1	Proiektuaren ondorioak	59
9.2	Ondorio pertsonalak	59
9.3	Etorkizunerako hobekuntza eta proposamenak	60
Eranskina		61
Bileren aktak		61
Lehenengo bilera		61
Bigarren bilera		62
Hirugarren bilera		63
Laugarren bilera		64
Barne-langilearen fitxa		65
Kanpo-langilearen fitxa		66
Atzerritartasun formularioa		67
Bibliografia		73

Irudien aurkibidea

2.1	LDE Diagrama	4
2.2	Gnatt diagrama - Garapena	6
4.1	MVC arkitekturaren diagrama	16
4.2	Lehenengo Sprinta - Erabilpen-kasuak	18
4.3	Bigarren Sprinta - Erabilpen-kasuak	19
4.4	Hirugarren Sprinta - Erabilpen-kasuak	21
4.5	Laugarren Sprinta - Erabilpen-kasuak	23
4.6	Bosgarren Sprinta - Erabilpen-kasuak	25
4.7	Erabilpen-kasuen diagrama	29
4.8	Erabiltzailea gehitu interfazea	30
4.9	Erabiltzailea gehitu, sekuentzia-diagrama	31
4.10	Erabiltzaileak ikusi	32
4.11	Erabiltzaileak ikusi, sekuentzia-diagrama	33
4.12	Diskriminazio motako kasua gehitu interfazea	34
4.13	Kasua sortu, sekuentzia-diagrama	35
4.14	Langilea gehitu interfazea	36
4.15	Langilea gehitu, sekuentzia-diagrama	37
4.16	Entitate-erlazio eredu hedatua	38
4.17	Domeinuaren eredua	41
5.1	Aplikazioaren bistak	44
5.2	Konexio asinkronoak zerbitzariarekin	44
5.3	Modeloak	45
5.4	Kontrolatzaileak	45
5.5	Modeloa: Zita	46
5.6	Zita modeloaren kontrolatzaileko <i>DeleteCita</i> funtzioa	46
6.1	Egoitzaren testak	48
6.2	Langilearen testak	49
6.3	Erabiltzailearen testak	49
6.4	Kasuaren testak	50
6.5	Interbentzioaren testak	50
6.6	Zitaren testak	51
6.7	Estaldura	51
6.8	E2E testak - Administratzailea	52
6.9	E2E testak - Langilea	53
1	Barne-langilearen fitxa	65

2 Kanpo-langilearen fitxa 66

Taulen aurkibidea

2.1	Orduen estimazioa	7
4.1	Lehenengo Sprinta - Erabiltzailearen historiak	18
4.2	Lehenengo Sprinta - Ordu-estimazioa	19
4.3	Bigarren Sprinta - Ordu-estimazioa	19
4.4	Bigarren Sprinta - Erabiltzailearen historiak	20
4.5	Hirugarren Sprinta - Ordu-estimazioa	21
4.6	Hirugarren Sprinta - Erabiltzailearen historiak	22
4.7	Laugarren Sprinta - Ordu-estimazioa	23
4.8	Laugarren Sprinta - Erabiltzailearen historiak	24
4.9	Bosgarren Sprinta - Ordu-estimazioa	25
4.10	Bosgarren Sprinta - Erabiltzailearen historiak	26
4.11	Seigarren Sprinta - Ordu-estimazioa	27
4.12	Seigarren Sprinta - Erabiltzailearen historiak	27
8.1	Ordu errealak	58

Sarrera

Gipuzkoako SOS Arrazakeria Espainiako Estatuko SOS Arrazakeria Elkarten Federazioan federatuta dagoen erakunde autonomoa da. Bere helburu nagusia Gipuzkoako arrazakeriaren agerpenen aurka borrokatzea da [1]. Horretarako zenbait egoitza dituzte Gipuzkoan zehar banatuta, eta egoitza bakoitzean erabiltzaile ezberdinen kasuen jarraipena egiten dute. Orain arte prozesu horiek kudeatzeko, software ezberdinak erabili dira egoitza bakoitzean: Excel, Acces, datu-base bat sarean, Word...

Horren ondorioz, ez dute informazio guztia leku batean bateratuta. Gainera, urtean zehar zenbait informazio esportatu behar dute eta urteko datuen inguruko txosten ezberdinak sortu. Hori egiteko, kalkulu guztiak eskuz egin behar dituzte eta denbora asko galtzen da. Horrez gain, egoitzen artean informazioa elkarbanatzea asko zailtzen da, ez baitute informazioa formatu berean gordea.

Gradu Amaierako Lan honen helburua SOS Arrazakeriako erabiltzaileak kudeatzeko web-aplikazio bat diseinatu, garatu eta ezartzea da.

Gipuzkoako SOS Arrazakerian lan egiten duten guztientzat eskuragarri dagoena, baina langile bakoitzak bere arloarekin zerikusia duten datuak soilik eskuratzeko aukera izatea bermatzen duena.

Proiektuaren kudeaketa-plana

2.1 Helburuak

Gradu Amaierako Lan honen helburua web aplikazio bat diseinatu, garatu eta ezartzea da, Gipuzkoako SOS Arrazakeria gobernuz kanpoko erakundeak kudeatzen duen informazioa datu-base batean jaso eta informazio honen kudeaketa errazteko.

2.1.1 Proiektuaren helburu zehatzen deskribapena

- SOS Arrazakeriarekin lanean jardun duten bizilagun programako familia eta boluntarioen kontakturako informazioa kudeatzea. Baita lankidetzan aritzen diren elkarteen ere. Helburua informazio guztia batu eta gero irizpideen arabera aukeratzeko aukera izatea da, egingo den jardueraren arabera.

- SOS Arrazakerian dituzten erabiltzaileen informazioa biltzea. Erabiltzaile bakoitzak oinarrizko kontakturako informazioa edukiko du, horrez gain kasuak izan ditzake erabiltzaileak, SOSeko langileek kudeatuko dituztenak.

Kasu bat erabiltzaile bakoitzak edukiko duen fitxa moduko bat da, erabiltzaile bakoitzak kasu bat baina gehiago izan ditzake. Aplikazioak hiru kasu mota ditu: lan-aholkularitza etxeko langileei, atzerriartasuna eta diskriminazioa. Adibidez, etxeko langileen lan-aholkularitza kasu batean, erabiltzaile baten lan egoeraren inguruko datu guztiak gordeko dira kasuan, eta ondoren SOS Arrazakeriako langileek jarraipen bat egingo dute kasu horren inguruan.

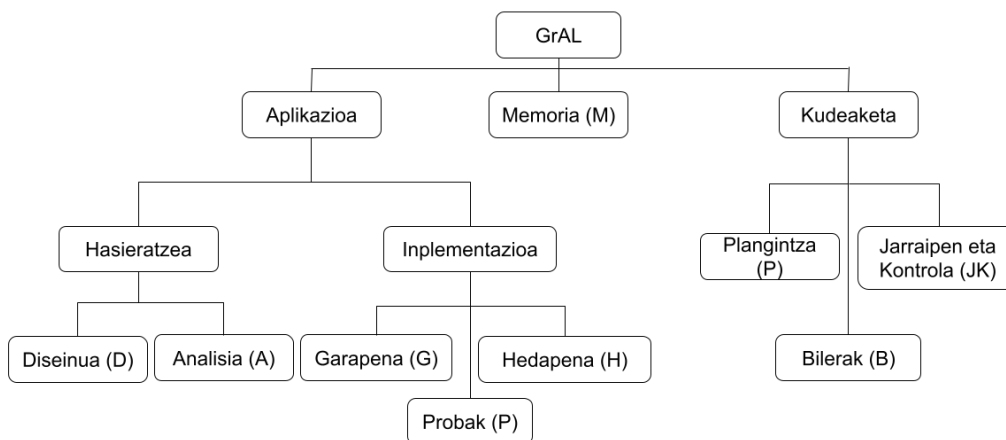
- Arreta bulegoetako kasuen jarraipena egitea. Bulegoetan hiru lan-esparru nagusi dituzte: atzerriartasuna, diskriminazioa eta etxeko langileentzako lan-aholkularitza. Lan eremu bakoitzean, honako informazio hau edukitzeko aukera behar da:
 - Arreta bulego eta lan-esparruetako ziten egutegia.
 - Kasuen erregistroa.
 - Kasu bakoitzaren dokumentazioa ikusteko aukera.
 - Kasuen jarraipena.

- Kasu bakoitzarentzat interbentzioak egiteko aukera izatea. Interbentzioak SOS Arrazakeriako langileek egingo dituzte eta interbentzio bakoitzean egindako ekintzak azalduko dira.
- Langile bakoitzak kudeatzen dituen kasuen zerrenda osoa ikusteko aukera izatea. Baita amaitu gabe dauden kasuen zerrenda ikusteko aukera ere.
- SOS Arrazakeriako langileek zenbait dokumentu sortu behar dituzte datu-baseko datuekin. Helburua izango da dokumentu hauek automatikoki sortzeko aukera izatea. Datu-basean dagoen informazioa pdf edo excel formatu zehatzetan deskargatzeko aukera eduki behar da.

2.2 Plangintza

2.2.1 LDE

Proiektua lan eta ataza ezberdinetan banatu da (Ikus proiektuari dagokion LDE diagrama 2.1 irudian). Ikus daiteken bezala, hiru atal nagusitan banatu da proiektua: Aplikazioa, Memoria eta Kudeaketa. Aplikazioaren atalean, bi azpiatal nagusi izango dira: Hasieratzea eta Inplementazioa.



2.1 Irudia: LDE Diagrama

Diseinua (D) lan-paketeak aplikazioaren diseinuarekin erlazionatutako atazak hartzen ditu barne.

Analisisa (A) lan-paketeak aplikazioaren garapenerako beharrezkoa den informazioa esku-ratzea izango du barne.

Garapena (G) lan-paketeak aplikazioa garatzeko behar diren atazak hartzen ditu barne.

Probak (P) lan-paketeak aplikazioaren funtzionamendua probatzeko testen atazak izango ditu.

Hedapena (H) lan-paketeak aplikazioa martxan jartzeko beharko diren atazak izango ditu.

Plangintza (P) lan-paketeak hasierako plangintza egiteko atazak eta behar badira, plangintza egunean mantentzeko behar daitezkeenak barne izango ditu.

Bilerak (B) lan-paketeak proiektua aurrera eramateko Gradu Amaierako Lanaren zuzendariarekin eta SOS Arrazakeriako kideekin egindako bilerak hartzen ditu barne.

Jarraipen eta Kontrola (JK) lan-paketeak proiektuaren garapen egokia bermatuko duten atazak edukiko ditu eta konkretuki, proiektuko dedikazioen jarraipena eta epeen eta espezifikazioen betetzea.

2.2.2 Atazen deskribapena

Analisia (A)

1. Sistemaren eskakizunak zehaztu.
2. Erabilpen-kasuak adostu.
3. Erabiliko diren teknologiak zehaztu.

Diseinua (D)

1. Aplikazioaren diseinua garatu.
2. Datu-basea diseinatu.

Garapena (G)

1. Datu-basearen implementazioa.
2. Web aplikazioaren garapena.

Probak (P)

1. Aplikazioaren testak implementatu.
2. Aplikazioaren testak exekutatu eta erantzunak aztertu.

Hedapena (H)

1. Aurreko datu-baseko informazioa datu-base berrira pasatu.
2. Zerbitzariaren instalazio eta konfigurazioa.
3. Aplikazioa erabiltzen hasi.

Memoria (M)

1. Memoriaren txostena idatzi.
2. Memoriaren azken zuzenketak.
3. Defentsa prestatu.

2.3. Informazio eta komunikazio sistemak

Ataza	Orduen estimazioa	Hasiera data	Amaiera data	Iraupen totala
Aplikazioa	250 ordu	10/12/2020	10/06/2021	182 egun
Hasieratzea	60 ordu	10/12/2020	15/02/2021	67 egun
A.1	10 ordu	10/12/2020	10/01/2021	31 egun
A.2	10 ordu	10/12/2020	10/01/2021	31 egun
A.3	10 ordu	10/01/2020	15/02/2021	31 egun
D.1	15 ordu	10/01/2021	15/02/2021	31 egun
D.2	15 ordu	15/01/2021	15/02/2021	31 egun
Inplementazioa	190 ordu	20/02/2021	10/06/2021	110 egun
G.1	10 ordu	25/02/2021	05/05/2021	69 egun
G.2	140 ordu	20/02/2021	20/05/2021	89 egun
P.1	10 ordu	10/05/2021	20/05/2021	10 egun
P.2	5 ordu	20/05/2021	24/05/2021	4 egun
H.1	20 ordu	25/05/2021	05/06/2021	11 egun
H.2	15 ordu	25/05/2021	08/06/2021	15 egun
H.3	5 ordu	01/06/2021	10/06/2021	9 egun
Memoria	70 ordu	01/02/2021	28/06/2021	147 egun
M.1	60 ordu	01/02/2021	18/06/2021	137 egun
M.2	2 ordu	18/06/2021	20/06/2021	2 egun
M.3	8 ordu	20/06/2021	28/06/2021	8 egun
Kudeaketa	30 ordu	11/11/2020	20/06/2021	220 egun
PL.1	10 ordu	10/12/2020	15/12/2020	5 egun
PL.2	6 ordu	10/12/2020	15/12/2020	5 egun
PL.3	2 ordu	-	-	-
B.1	10 ordu	11/11/2020	20/06/2021	220 egun
B.2	-	-	-	-
JK.1	2 ordu	11/11/2020	15/11/2020	4 egun
JK.2	-	-	-	-
JK.3	-	-	-	-
Guztira	350 ordu	2020/11/11	28/06/2021	229 egun

2.1 Taula: Orduen estimazioa

2.3 Informazio eta komunikazio sistemak

2.3.1 Informazio-sistema

Proiektua garatzeko behar den informazio gehiena hodeian mantentzea izango da helburua, honela bermatuko da informazioa ez galtzea. Horrez gain, ordenagailu pertsonalean ere egiten ari denaren segurtasun kopia bat izango da.

Aplikazioaren garapena GitHub-en[2] gordeko da, honela ordenagailu pertsonalean aldaketa bat egiten denero, aldaketak GitHub-era ere igoko dira. Honekin segurtasun kopia bat edukitzea bermatzeaz gain, proiektuaren bertsio kontrola ere izango da. Hona-koa da proiektua GitHub-en ikusteko helbidea: <https://github.com/joanemannion1/sosracismo>

Memoriaren garapenerako Overleaf[3] hodeian oinarritutako \LaTeX editorea erabiliko

da. Dokumentu zientifikoak idazteko, editatzeko eta argitaratzeko erabiltzen da. Honela, memoriaren informazioa ez galtzea bermatuko da. Gainera nire memoria deskargatu eta GitHub-era igoko da periodikoki, segurtasun-kopia mantentzeko.

2.3.2 Komunikazio-sistema

Gradu Amaierako Laneko zuzendariekin komunikazioa mantentzeko unibertsitateko eposta erabiliko da. Bilera bat egin beharrez gero, Bb Collaborate bidez egingo da.

SOS Arrazakeriako kideekin komunikatzeko, unibertsitateko eposta, Whatsapp taldea eta Google Meet-eko bilera periodikoak izango ditugu.

2.4 Arriskuak eta kalitatea

2.4.1 Arriskuen analisia

Proiektu bat garatzerako orduan, beharrezkoa da proiektuan izan daitezkeen arazoak aurreikusi eta soluzio bat proposatzea. Horrela, arazo mota ezberdinak ekiditeko pausuak emango dira hasieratik. Egon daitezkeen arazo guztiak zerrendatu dira, eta arazo bakoitzari aurre egiteko soluzio bat azaldu da.

- **Datuen galera:** Edozein ezusteko izanda, proiektuaren informazioa galtzeko aukera dago beti. Arazo horri aurre egiteko, *Git* biltegi bat izango da, *GitHub* plataforman egongo dena internet bidez atzigarri. Horren bidez, datuen segurtasun-kopia periodiko bat izango da uneoro.
- **Datu-basearen diseinu okerra:** Aplikazioaren oinarrian behar duen datu-basea dago, eta datu-base honen diseinu eta inplementazio okerra eginez gero, aplikazio guztiaren funtzionamendua okerra izango da. Beraz, DBaren diseinua Gradu Amaierako Laneko zuzendariekin eta SOS Arrazakeriako kideekin adostuko da garapenarekin hasi aurretik.
- **Aplikazioaren testak huts egitea:** Proiektuaren garapenarekin bukatutakoan testak egin eta test hauek huts eginez gero, proiektuaren garapena asko luzatu daiteke. Hau ekiditeko, testak periodikoki egiten joango gara, aplikazioaren zati bat garatzen deno testak burutzen.
- **Estimazio desegokiak:** Denbora estimazio okerrak eginez gero proiektu hasieran, ondoren proiektu osoa atzeratzeko arriskua egongo da. Hau ekiditeko, proiektuaren kudeaketa plana kontu handiz egitea beharrezkoa izango da.
- **Hedapen arazoak:** Programaren garapena egin ostean, proiektua hedatu beharko da. Datu-basean beharrezko datuak sartu beharko dira, eta horrek denbora asko eraman dezake. Arrisku hori ekiditeko ezinbestekoa izango da SOS Arrazakeriako kideen laguntza, datu guztiak bildu eta datu-base berrira sartu ahal izateko.

2.4.2 Kalitatearen kudeaketa

Proiektu honen helburua, Gipuzkoako SOS Arrazakeriarentzat web aplikazio bat garatzea da. Helburua betetzeko, beharrezkoa izango da aplikazioa kalitate onekoa izatea. Horretarako,

hasieran ezarritako helburu guztiak betetzea beharrezkoa izango da. Baina horretaz gain, beste zenbait aspektu hartuko dira kontutan produktuaren kalitatea bermatzeko.

- **Kode ulergarri eta garbia:** Beharrezkoa da errorerik gabeko kode bat garatzea, hau bermatzeko zenbait proba egingo dira kodearen gainean. Gainera, kodeak ulergarria izan behar du, geroago beste garatzaile batek proiektuarekin jarraitu beharko balu, lana errazteko.
- **Interfazearen diseinua:** Garrantzitsua da garatuko den web aplikazioaren diseinua atsegina izatea. Gainera, erabiltzeko erraza eta ulergarria izan behar da, langileek aplikazioa erabiltzeko orduan arazorik ez izateko.
- **Funtzionalitateak:** Proiektua amaitzean, ezarritako erabilpen-kasu guztiak betetzen direla zehaztea beharrezkoa izango da. Funtzionalitate guztiak errore gabe funtzionatu behar dute, bezeroak eskaturiko helburuak beteaz.
- **Bezeroarekin harremana:** Kalitatezko proiektu bat bermatzeko, beharrezkoa izango da bezeroarekin harremana mantentzea proiektuaren garapen osoan zehar. Lortzen diren helburuak erakutsi, eta guztiarekin ados daudela ziurtatzeko.

2.5 Interesatuak

Proiektu honen interesatu nagusia lanaren egilea izango da, bera izango baita proiektua garatu, defendatu eta kalifikazioa jasoko duena. Gainera, honelako proiektu erreal bat aurrera eramaterako orduan ikasleak asko ikasi eta esperientzia lortzeko aukera izango du. Ikaslearen ardura izango da proiektua garatzeko egin beharreko lana zuzen burutzea.

Bigarren interesatuak Gipuzkoako SOS Arrazakeriako kideak izango dira. Bertako langileen ardura izango da proiektuaren helburuak finkatu eta hedapenarekin laguntzea.

Hirugarrenik, Gradu Amaierako Lanaren zuzendariak izango ditugu: Olatz Arbelaitz eta Ana Arruarte. Beraiek proiektua gainbegiratu eta garapenean zehar gidatzeko ardura izango dute.

Amaitzeko, bigarren mailako interesatu gisa, proiektuaren defentsa ebaluatuko duen epaimahaia izango da. Epaimahairen helburua proiektua kalifikatzea izango da.

Teknologiak eta metodologia

Atal honetan proiektua garatzeko erabili diren teknologiak azalduko dira. Erabilitako teknologiak aukeratzeko orduan, garrantzia eman zaio kode irekikoa izateari eta jadanik teknologia ezagutzeari. Horrela, ez da ikaste-prozesurik eman beharko.

3.1 Proiektuan erabilitako teknologiak

Proiektua aurrera eramateko tresna mota ezberdinak behar izan dira. Alde batetik, bezeroaren aldea sortzeko behar izan diren teknologiak daude. Teknologia hauek erabiltzaile interfaze erabilerraz eta argi bat egiteko helburua kontuan izanda hautatu dira. Bestaldetik, zerbitzariaren aldeko teknologiak hautatu dira. Teknologia hauek datu erabiltzaile interfazea, eta zerbitzaria komunikatzeko erabili dira, baita datu-baseko datuak kudeatzeko. Bukatzeko, garapenerako erabili ditudan teknologiak aipatuko dira. Hauek, proiektuaren garapena erraztu eta iturburu-kodearen kalitatea bermatzen lagundu dute.

3.1.1 Bezeroaren aldeko teknologiak

JS

JavaScript objektuei orientatutako, goi-mailako programazio-lengoaia da.

Lengoaia hau hautatu dut bezeroaren eta zerbitzariaren aldeko kodea lengoaia berean idatzita egotea garrantzitsua iruditu zaidalako. Baita *React* liburutegia lengoaia honekin sortua dagoelako, eta liburutegi honetan oinarritu dudalako nire proiektua.

HTML lengoaia erabiltzea zen beste aukera, baina gaur egun JSen oinarritutako web aplikazioak oso erabiliak dira. Beraz, liburutegi asko eskaintzen ditu JSek garapen prozesuarekin laguntzeko.

React

React [4] JavaScript-eko liburutegi bat da, kode irekikoa, erabiltzailearen interfazeak sortzeko diseinatua dagoena. Garatzaileei denbora guztian aldatzen diren datuak erabiltzen dituzten aplikazioak eraikitzen laguntzen saiatzen da React. Helburua erraza, adierazgarria

eta konbinatzen erraza izatea da. Honen erabilerak asko azkartuko du inplementazioaren lana, kodearen berrerabilpena asko errazten baitu liburutegi honek.

Angular [5] erabiltzea izan da beste aukera. Baina Angularren ikasteko prozesua luzeagoa da React-ekoa baino. Horrez gain, React-ek DOM (Dokumentuaren Objektu Eredua) birtual bat erabiliz egiten du lan, hau da, aldaketak eguneratzeko aukera ematen digu, ia HTML dokumentu osoa berridatzi gabe. Horrela eguneraketak askoz azkarrago egiten ditu eta errendimendu azkarra bermatzen du.

CSS

CSS (*Cascading Style Sheets* edo kaskada-estilo orriak lengoia erabiliko da kodearen atal batzuen diseinua zehaztu eta aldatzeko.

3.1.2 Zerbitzariaren aldeko teknologiak

JS

Lehenago aipatu dudana bezala, zerbitzariaren aldean ere *JavaScript* lengoia erabiltzea erabaki dut. Hau lehen aipatutako erabaki berdinenetan oinarrituta egin dut.

Node.js

Node.js[6] programatzeko ingurune bat da, interneteko aplikazio eskalagarriak sortzeko diseinatua dagoena, batez ere zerbitzariak. *Node.js* inguruneak kodea exekutatzeko Google-ren *V8 JavaScript* motorra erabiltzen du. Liburutegi batzuk ere erabiltzen ditu bere aplikazioek Web-zerbitzari moduan lan egin ahal dezaten.

Gertakizunetan oinarritutako arkitektura bat eta sarrera/irteera asinkronoa erabiltzen ditu, sistemaren denbora minimizatzeko eta eskalagarritasuna maximizatzeko. Gaur egun, programazio-ingurune oso ezaguna da web-garatzailen artean eta mundu osoko zenbait enpresa ezagunek erabiltzen dutena.

Express.js

Node.js aplikazioen azpiegitura minimoa eta malgua da *Express*[7], eta web-aplikazioetarako eta aplikazio mugikorretarako ezaugarri-multzo sendoa eskaintzen du. Kode irekikoa da, web-aplikazioak erraz sortzeko aukera ematen du eta hainbat ezaugarri eskaintzen ditu. Hala nola ibilbideak erabiltzea (helbideratzea), datu-baseekin integratzea, akatsak kudeatzea ...

MySQL

MySQL[8] datu-baseak kudeatzeko sistema bat da eta munduko kode irekiko datu-base ezagunentzat hartzen da. Hau hautatzeko arrazoia izan da, ondoen ezagutzen dudana datu-baseak kudeatzeko sistema dela. Hortaz, erraztasun handiz erabiltzeko aukera izango da.

3.1.3 Garapenerako teknologiak

Visual Studio Code

Microsoft-ek garatutako kode editore bat da VS Code [9]. Oso lagungarria da kodea idazterako orduan, luzapen ezberdin asko instalatzeko aukera eskaintzen du, kodea idaztea erraztuko dutenak.

Atom [10] erabiltzea izan da beste aukera. Atom eta Visual Studio Code oso kode editore antzekoak dira, baina VS Code erabiliagoa da gaur egun garatzaileen artean.

MarvelApp

Marvel[11] prototipoak egiteko tresna bat da, hau erabiliz aplikazioaren diseinu funtzional bat egiteko aukera izan dut. Behin hau eginda, SOS Arrazakeriako erabiltzaileei diseinua erakutsi, eta beraiekin diseinua adosteko aukera izan dut.

Git

Git[2] bertsio-kontrolerako garatu den softwarea da. Bertsio-kontrolak garatzen ari garen aplikazioaren aldaketen historia edukitzea ahalbidetzen digu. Horretarako software gehiago egon arren, git aukeratu dut kode irekikoa delako, gaur egun erabiliena delako eta unibertitatean zenbait proiektu garatzeko erabili dudalako.

SonarCloud

Sonar[12] iturburu-kodea ebaluatzeko plataforma bat da. Software librea da eta programa baten iturburu-kodearen kalitatea hobetzen lagun dezake. Besteak beste kodearen diseinua, kode zati bikoiztuak, iruzkinak eta erroreak aztertzen ditu. Hodeian dagoen Sonar bertsioa aukeratu dut, hau GitHub-eko proiektuarekin lotu dezakedalako, eta honela GitHuben eginiko aldaketa guztiak zuzenean aztertuko dira.

3.2 SCRUM metodologia

Proiektuaren garapena eta kudeaketa nola egingo den erabakitzeke orduan Software Ingeniaritzan jasotako ezagutzak aplikatu dira. Honela, dauden aukera ezberdinei buruzko azterketa bat egin da, GRaL honentzat metodo egokiena aukeratzeko helburuarekin. Azterketa hau egiteko, kontuan hartu da Gradu Amaierako Lan honek jarraitzen duen metodologia arina eta malgua izan behar dela, pertsona bakarrak kudeatuko baitu proiektua. Gainera, proiektuak moldagarria izan behar du, bezeroaren, SOS Arrazakeriaren, beharretara egokitzeko.

Ondorioz, ideia hauetan oinarrituz, eta kontuan edukita proiektuan parte hartzen duen taldea oso txikia dela, **SCRUM** [13] metodologia hautatu da.

SCRUM metodologia zuzeneko komunikazioa duten talde txikiei zuzenduta dago. Ingeniaritza konkurrentea erabiltzen du, ziklo edo fase sekuentzialen orde. Kalitate handiagoko softwarea lortzen laguntzen du, ekipoaren produktibitatea maximizatu eta proiektuaren arriskuak murrizten ditu. Horrez gain, bezeroak egin ditzakeen eskakizun edota aldaketetara egokitzeko malgutasuna ematen du metodologia honek.

SCRUM metodologia hiru rol nagusitan oinarritzen da:

- **SCRUM maisua (SCRUM master):** Proiektuan *SCRUM* metodologia aplikatzeaz arduratzen da. Ez da taldeko liderra, baina metodologiaren jarraipena egiten duen taldeko kidea da. Honetarako, taldekideen arteko bilera bat antolatzen du behar denean.
- **Produktuaren jabea:** Bezeroaren parte irudikatzen du, eta *SCRUM* taldea negozioaren ikuspegitik egoki lan egiten ari dela ziurtatzeaz arduratzen da. Horretarako, bezeroarekin eta *SCRUM* taldearekin elkarlanean arituko da. Proiektuaren baldintzak jaso eta ezagutzeaz eta produktu-pilako produktuak (*product backlog-a*) antolatu eta lehenesteaz arduratzen da.
- **SCRUM team:** *SCRUM* taldeak proiektuaren garapena gauzatuko du *sprint*-en bidez.

Proiektua *sprint*-en bidez gauzatuko da, *sprint* bakoitzak aste bat eta bi aste arteko iraupena izango du, nahiz eta estimatutako denbora aldatu egin daitekeen. Produktu-pila (*product backlog*) proiektua garatu bitartean egin nahi diren lan guztien zerrenda da. Produktuaren helburu edota baldintza nagusiak biltzen ditu, eta horiek erabiltzaile historia gisa adierazten dira. Erabiltzaile historia azken erabiltzailearen ikuspegitik idatzitako software-funtzio baten azalpen orokor eta informala da. Garrantzitsua da adieraztea ezen, *SCRUM* metodologia aplikatzen denean, ez dela beharrezkoa baldintza edota helburu guztiak proiektuaren hasieran definitzea; helburu horiek produktu-pilan daitezke proiektua garatzen ari den bitartean.

Sprint bati ekin aurretik, taldeak erabakiko du zer *item* osatu ahal izango diren *sprint* horretan, eta *item*-ak produktu-pilatik *sprint*-eko produktueta *sprint backlog* mugituko dira. Iterazioan zehar beharrezkoa deno bilera labur bat egiten da, *Daily Scrum* izenekoa, proiektuaren egoera aztertzeko eta hurrengo eguneko jarduerak planifikatzeko. Hala, taldeak ikuspen orokor bat lortzen du: zer egin den eta zer egin behar den.

Proiektuaren garapena pertsona bakar batek egingo duenez, hiru rolak esleituko zaizkio. Pertsona hori arduratu da bai *item*-ak produktu-pilara, bai *sprint* bidez funtzionalitateak garatzeaz. Halaber, jardunaldi batzuen hasieran denbora-tarte txiki bat hartuko da galdera batzuei erantzuteko. Hala, eguneko bilera simulatuko da, proiektuaren egoera aztertzeko, proiektuaren ideia argi bat izateko eta eman beharreko hurrengo urratsak ahalik eta zehatzenak izateko.

Proiektuaren garapen osoan zehar SOS Arrazakeriako kideekin harremana mantendu da, hala, produktuaren bezeroaren laguntzailearen rola izan dute kideek. Beraiekin harremanetan egonez feedbacka jaso eta akatsak zuzentzeko aukera izango da proiektua garapenean zehar.

Analisia eta diseinua

Atal honetan, proiektuaren analisia burutuko da. Analisia egiteko, SOS Arrazakeriak egiten duen lana eta gaur egun erabiltzen dituen metodoak aztertu dira, azterketa honetan garatuko den web-aplikazioaren helburu nagusiak zehaztu dira. Ondoren, aplikazioaren inguruko ideia nagusi bat izanik, erabaki zehatzagoak hartzeko bilerak egin dira, zehazki kudeatu behar duten informazioa aztertu eta nola antolatuko dugun erabakiz.

4.1 Arkitektura

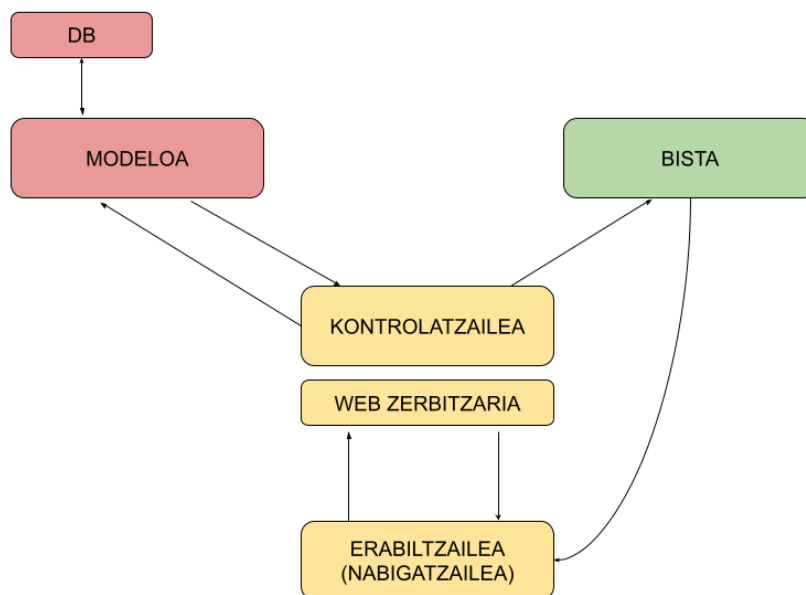
4.1.1 MVC Arkitektura

Proiektuaren garapenean MVC (*Model-View-Controller*) Arkitektura jarraituko da, hau *Modeloa-Bista-Kontrolatzailea* [14] arkitektura da eta softwarearen arkitektura-eredu bat da. Eredu honek, datuak eta, batez ere, negozio-logika bereizten ditu erabiltzailearen interfazetik. Horretarako, MVCK hiru osagai ezberdin eraikitzea proposatzen du: eredu, bista eta kontrolatzailea. Software-arkitekturaren eredu hau kodea berrerabiltzeko eta kontzeptuak bereizteko ideietan oinarritzen da. Ezaugarri horiek aplikazioak garatzeko eta ondoren mantentzeko lana erraztu nahi dute.

- **Modeloa:** Datuak kontsultatu, gehitu, eguneratu eta ezabatzeaz arduratzen da, normalean datu-base baten bitartez.
- **Bista:** Datuen errepresentazio bisualaz arduratzen den osagaia da, hau da, interfaze grafikoa.
- **Kontrolatzailea:** Erabiltzailearen aginduak jaso eta modeloari datuak eskatuko dizkio. Ondoren, informazio hau bistari pasatuko dio.

4.1.2 MVC Arkitektura aplikazioan

4.1 irudian, proiektuaren arkitekturaren diagrama erakusten da. Bistak JSX, JavaScript eta CSS erabiliz sortuko dira. Bista horiek erabiltzailearen aldean egongo dira. Kontrolatzaileak



4.1 Irudia: MVC arkitekturaren diagrama

JavaScript erabiliz garatuko dira eta hauek zerbitzarian egongo dira kokatuta, bistaren eskakizunak jaso eta modeloari egin beharreko eragiketak adieraziko dizkio.

Modeloak ere JavaScript erabiliz garatuko dira, eta zerbitzarian egongo dira. Modeloen ardura izango da, datuak lortu, gehitu, eguneratu eta ezabatzea. Lorturiko informazioa kontrolatzaileari pasa beharko dio.

Aplikazioaren datuak MySQL datu-base batean egongo dira gordeta. Datu-base hau ere zerbitzarian egongo da kokatua. Datuak datu-basetik bistara bidaltzeko, gehienetan, objektuak eta array-ak erabiliko dira.

4.2 Erabilpen-kasuak

Proiektuaren erabilpen kasuak ezartzeko, proiektuaren helburuak hartu dira oinarritzat. Bizilagun programako familien eta boluntarioen informazioaren kudeaketa baztertzea erabaki da. Hau, denbora faltagatik izan da, beste helburuei lehentasun handiagoa eman zaie. Baztertutako helburuak, etorkizunean egiteko helburu bezala ezarriko dira.

Proiektuaren erabilpen-kasuak azaltzeko, SCRUM metodologiako produktu-pilan (*Product backlog*) oinarrituko gara. Produktu-pila proiektua garatu bitartean egin nahi diren lan guztien zerrenda da. Pila erabiltzailearen historiekin [15] osatzen da, hau, erabiltzailearen ikuspegitik azaldutako helburua da. Softwarearen funtzio baten azalpen orokor eta informal bat.

Sprint bakoitzean produktu-pilako zenbait helburu beteko dira, atal honetan sprint bakoitzari eman zaizkion helburuak azalduko dira, horretarako erabiltzailearen historiak erakutsiko dira. Gainera, Sprint bakoitza egiteko aurreikusitako denbora azalduko da, eta sprint bakoitzeko helburu bakoitza betetzeko beharko den denbora ere.

Sprint bakoitza betetzeko jarritako helburuak, erabilpen-kasuak betetzean oinarrituta egongo dira. Beraz, sprint bakoitzaren azalpenarekin batera, beteko diren erabilpen-kasuak erakutsiko dira. Bukaeran, erabilpen-kasu guztien diagrama ere erakutsiko da.

4.2.1 Aktoreak

Proiektu honetan bi aktore mota bereiziko dira: administratzailea eta langilea. Hau web aplikazioaren segurtasuna bermatzeko izango da.

- **Langilea:** Langileak, erabiltzaile bat sortu, editatu eta ezabatzeko aukera izango du. Berak sortutako erabiltzaile bakoitzari, kasu bat gehitu ahalko dio, eta kasua editatu eta ezabatzeko aukerak ere izango ditu. Kasu bakoitzaren inguruan interbentzioak egiteko aukera ere izango du, eta interbentzioak ere, editatu eta ezabatzeko aukera izango du. Gainera, zitak sortzeko aukera izango du, baita beste langileen zitak ikusteko aukera.
- **Administratzailea:** Administratzaileak langileak egin dezakeen guztia egiteko aukera izango du. Horrez gain, langileen kudeaketa baimenduko zaio administratzaileari. Hau da, langileak sortu eta ezabatzeko aukera izango du, baita egoitzak sortu eta ezabatzeko aukera ere.

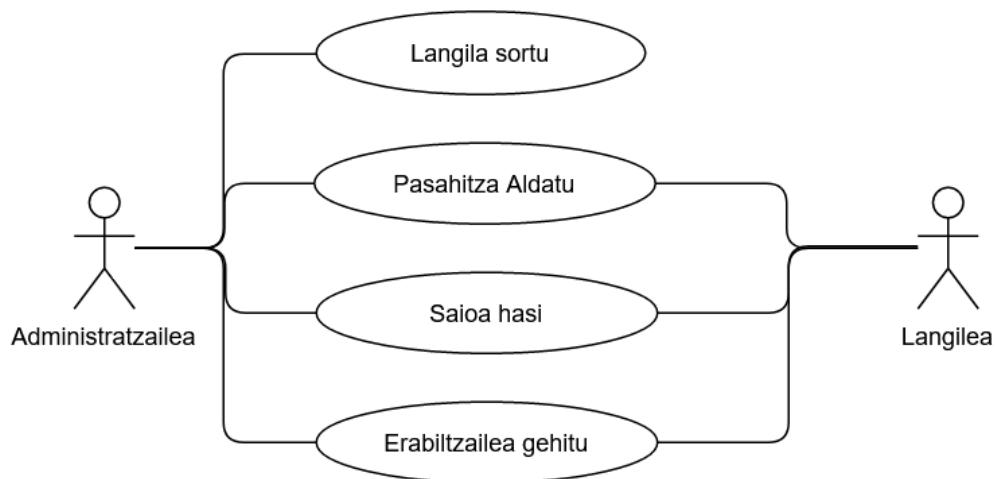
4.2.2 Sprintak

Lehenengo sprinta

Lehenengo Sprintean proiektuaren inplementazioa hasieratuko dugu, horrez gain, [4.1](#) taulan ikus daitezkeen lau helburu beteko dira: langilea sortu, pasahitza aldatu, saioa hasi eta erabiltzailea gehitu. Helburu horiek betetzeko aurreikusitako ordu kopurua [4.2](#) taulan erakusten da. Sprint honetan [4.2](#) irudian azalduko erabilpen-kasuak bete dira.

Bigarren sprinta

Bigarren Sprintean [4.4](#) taulan ikus daitezkeen lau helburu beteko dira: erabiltzaileak ikusi, erabiltzaileak iragazi, egutegia ikusi eta zitak gehitu. Helburu hauek betetzeko aurreikusitako ordu kopurua [4.3](#) taulan erakusten da. [4.3](#) irudian lortuko diren erabilpen-kasuak azalduko dira.



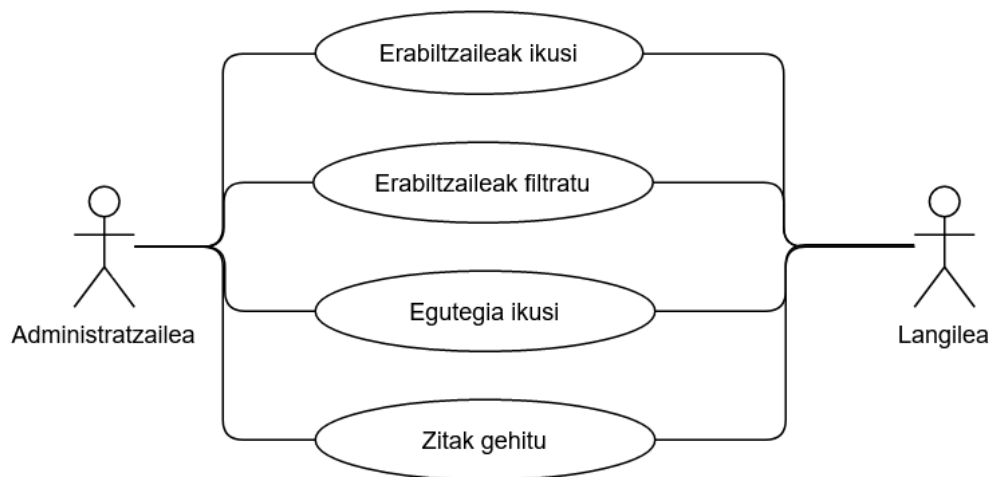
4.2 Irudia: Lehenengo Sprinta - Erabilpen-kasuak

Zenb.	Izena	Helburua	Baliozkotzea
1	Langilea sortu	Administratzaileak beste langile bat sortzeko aukera izatea.	Datu-basera gehituko dira, langilearen eposta, automatikoki sortutako pasahitza, izen-abizenak, egoitza eta kolore bat. Gainera, langileari eposta bidez bidaliko zaio bere pasahitza.
2	Pasahitza aldatu	Langileak pasahitza aldatzea.	Zerbitzariak langilearen uneko pasahitza konprobatu, eta berdina bada pasahitz berria gordeko du datu-basean.
3	Saioa hasi	Langileak saioa hastea.	Erabiltzailearen interfazean langileak eposta eta pasahitza sartu, eta zerbitzaritik konfirmazio mezu bat jasotzea.
4	Erabiltzailea gehitu	Langileak erabiltzaile berri baten datuak gehitzea.	Langileak erabiltzailearekin interfazean, formulario bidez, erabiltzaile berriaren datu guztiak sartuko ditu. Datu hauek zerbitzarira bidali eta datu-basean gordeko dira. Zerbitzariak konfirmazio mezu bat bidaliko du.

4.1 Taula: Lehenengo Sprinta - Erabiltzailearen historiak

Helburua	Hasiera-data	Bukaera-data	Ordu kopurua
Langilea sortu	2021/03/01	2021/03/05	8
Pasahitza aldatu	2021/03/05	2021/03/10	5
Saioa hasi	2021/03/10	2021/03/015	10
Erabiltzailea gehitu	2021/03/15	2021/03/17	5

4.2 Taula: Lehenengo Sprinta - Ordu-estimazioa



4.3 Irudia: Bigarren Sprinta - Erabilpen-kasuak

Helburua	Hasiera-data	Bukaera-data	Ordu kopurua
Erabiltzaileak ikusi	2021/03/17	2021/03/18	7
Erabiltzaileak iragazi	2021/03/18	2021/03/19	8
Erabiltzailea eguneratu	2021/03/19	2021/03/20	2
Erabiltzailea ezabatu	2021/03/20	2021/03/21	2
Egutegia ikusi	2021/03/21	2021/03/25	5
Zitak gehitu	2021/03/25	2021/03/25	5

4.3 Taula: Bigarren Sprinta - Ordu-estimazioa

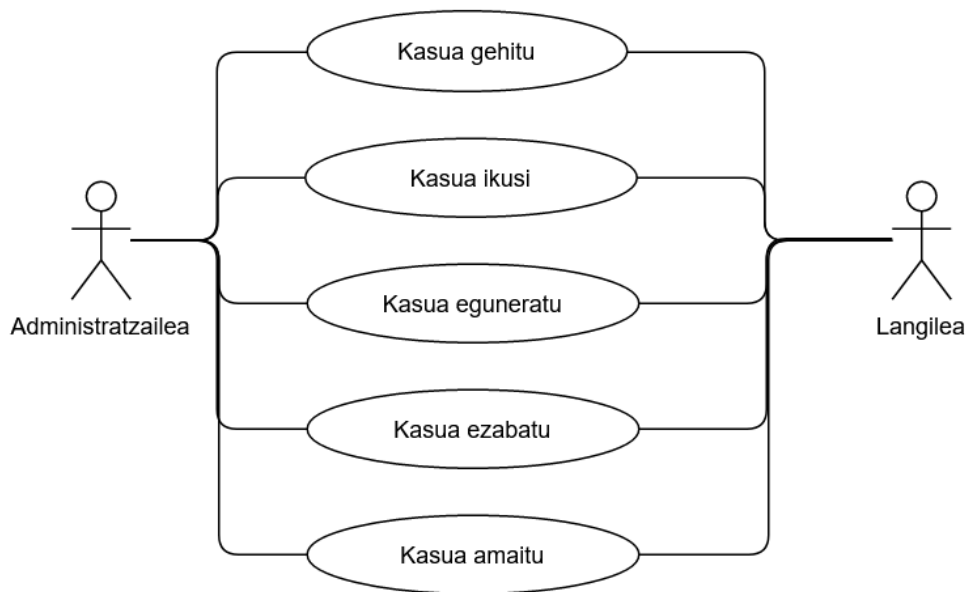
4. ANALISIA ETA DISEINUA

EHZ	Izena	Helburua	Baliozkotzea
5	Erabiltzaileak ikusi	Langileak gehitutako erabiltzaileen zerrenda ikustea.	Zerbitzariak datu-basetik langile zehatz batek gehitu dituen erabiltzaile guztiak lortuko ditu, ondoren erabiltzailearen interfazean erakutsiko dira.
6	Erabiltzaileak iragazi	Erabiltzaileen zerrenda iragazteko aukera izatea.	Datu-basetik jaso diren erabiltzaileak nazionalitate eta egoitzaren arabera iragazteko aukera eskainiko da.
7	Erabiltzailea eguneratu	Erabiltzaile baten informazioa eguneratzea.	Erabiltzaile zehatz baten informazioa erakutsi eta langileak informazio hau aldatzea. Aldaketa hauek zerbitzarira bidali eta datu-basea eguneratuko da.
8	Erabiltzailea ezabatu	Erabiltzaile baten informazioa ezabatzea.	Erabiltzaile zehatz bat hautatu eta hau ezabatzea, erabiltzailearen interfazeak alerta mezu bat erakutsiko du, langileak onartuz gero, erabiltzailea ezabatuko da datu-basetik. Baita erabiltzaile honi lotuak dauden kasuak ere.
9	Egutegia gehitu	Langileak egutegia ikustea.	Erabiltzailearen interfazean egutegia ikusteko aukera izatea.
10	Zitak gehitu	Langileak zita berri bat gehitzeko aukera izatea.	Langileak erabiltzailearen interfazean, formulario bidez, zita berri baten datu guztiak sartuko ditu. Datu horiek zerbitzarira bidali eta datu-basean gordeko dira. Zerbitzariak konfirmazio mezu bat bidaliko du.

4.4 Taula: Bigarren Sprinta - Erabiltzailearen historiak

Hirugarren Sprinta

Hirugarren Sprintean 4.6 taulan ikus daitezkeen bost helburu beteko dira: kasua gehitu, kasuak ikusi, kasua eguneratu, kasua amaitu eta kasua ezabatu. Helburu horiek betetzeko aurreikusitako ordu kopurua 4.5 taulan erakusten da. 4.5 irudian, beteko diren erabilpen-kasuak erakusten dira.



4.4 Irudia: Hirugarren Sprinta - Erabilpen-kasuak

Helburua	Hasiera-data	Bukaera-data	Ordu kopurua
Kasua gehitu	2021/03/27	2021/03/30	7
Kasuak ikusi	2021/04/30	2021/04/01	3
Kasua eguneratu	2021/04/01	2021/04/03	2
Kasua amaitu	2021/04/03	2021/04/04	2
Kasua ezabatu	2021/04/04	2021/04/05	2
Excel-a exportatu	2021/03/04	2021/04/05	5

4.5 Taula: Hirugarren Sprinta - Ordu-estimazioa

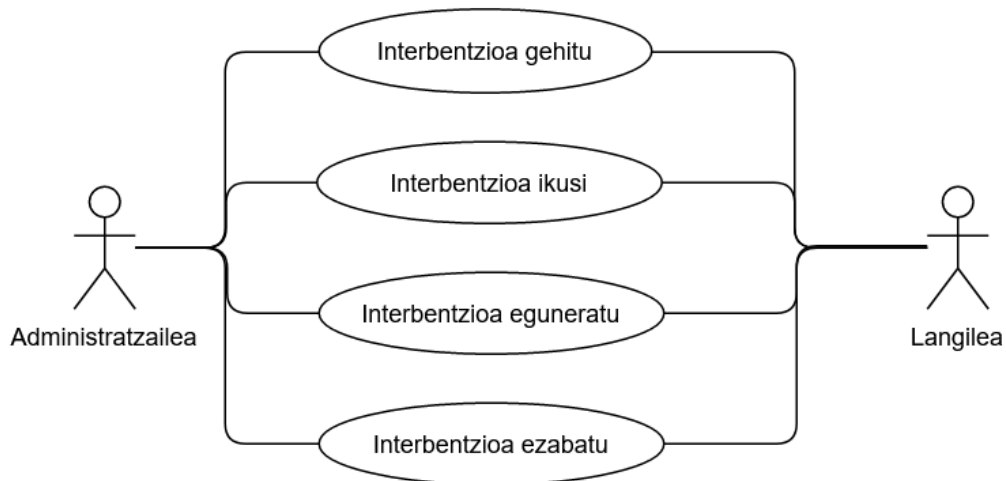
4. ANALISIA ETA DISEINUA

EHZ	Izena	Helburua	Baliozkotzea
11	Kasua gehitu	Langileak kasuak gehitzeko aukera izatea.	Langileak kasua gehitzeko hiru formularioetako bat bete (etxeko langilea, atzerritartasuna, diskriminazioa) eta datu hauek zerbitzarira bidali eta datu-basera gehitzea.
12	Kasuak ikusi	Langileak gehitutako kasuak ikusteko aukera izatea.	Zerbitzariak datu-basetik kasuaren informazioa jaso eta erabiltzailearen interfazean erakutsiko da.
13	Kasua eguneratu	Langileak kasu baten informazioa eguneratzeko aukera izatea.	Zerbitzariak datu-basetik kasuaren informazioa jaso eta erabiltzailearen interfazean erakutsiko da, langileak kasu hau aldatuko du eta egindako aldaketak datu-basean eguneratuko dira.
14	Kasua amaitu	Langileak kasu amaitutzat ematea.	Langileak kasu bat hautatu, eta amaitu egingo du. Erabiltzaile interfazeak mezu bat erakutsiko du, langilea ziur dagoen bermatzeko. Ziur egonez gero, datu-basean eguneratuko da kasuaren informazioa.
15	Kasua ezabatu	Langileak kasu bat ezabatzeko aukera izatea.	Kasu zehatz bat hautatu eta hau ezabatzea, erabiltzailearen interfazeak alerta mezu bat erakutsiko du, langileak onartuz gero, kasua ezabatuko da datu-basetik. Baita kasu honi lotuak dauden interbentzioak ere.
16	Excel-a exportatu	Langileak datuak excel dokumentu batean exportatzeko aukera izatea.	Data tarte zehatz bat hautatu eta data hauetan gehitu diren erabiltzaileen nazionalitateak eta beharrak excel taula batean deskargatzeka.

4.6 Taula: Hirugarren Sprinta - Erabiltzailearen historiak

Laugarren Sprinta

Laugarren Sprintean 4.8 taulan ikus daitezkeen lau helburu beteko dira: interbentzioa gehitu, interbentzioa ikusi, interbentzioa eguneratu eta interbentzioa ezabatu. Helburu hauek betetzeko aurreikusitako ordu kopurua 4.7 taulan erakusten da. 4.5 irudian, beteko diren erabilpen-kasuak erakusten dira.



4.5 Irudia: Laugarren Sprinta - Erabilpen-kasuak

Helburua	Hasiera-data	Bukaera-data	Ordu kopurua
Interbentzioa gehitu	2021/04/05	2021/04/07	3
Interbentzioak ikusi	2021/04/07	2021/04/08	2
Interbentzioa eguneratu	2021/04/08	2021/04/09	2
Interbentzioa ezabatu	2021/04/09	2021/04/10	2

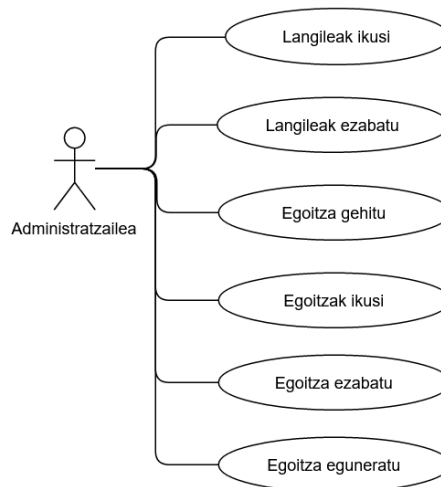
4.7 Taula: Laugarren Sprinta - Ordu-estimazioa

EHZ	Izena	Helburua	Baliozkotzea
17	Interbentzioa gehitu	Langileak interbentzioa gehitzea.	Langileak erabiltzailearen interfazean, formulario bidez, interbentzio berri baten datu guztiak sartuko ditu. Datu horiek zerbitzarira bidali eta datu-basean gordeko dira. Zerbitzariak konfirmazio mezu bat bidaliko du.
18	Interbentzioak ikusi	Langileak kasu baten interbentzioak ikusi ahal izatea.	Langileak erabiltzailearen interfazean, kasu bat hautatu, eta kasu honen interbentzioak ikustea. Honetarako, zerbitzariak datu-basetik lortu beharko ditu interbentzioak.
19	Interbentzioa eguneratu	Langileak interbentzio baten informazioa eguneratzeko aukera izatea.	Zerbitzariak datu-basetik interbentzioaren informazioa jaso eta erabiltzailearen interfazean erakutsiko da, langileak interbentzio hori aldatuko du eta egindako aldaketak datu-basean eguneratuko dira.
20	Interbentzioa ezabatu	Langileak interbentzio bat ezabatzeko aukera izatea.	Interbentzio zehatz bat hautatu eta hau ezabatzea, erabiltzailearen interfazeak alerta mezu bat erakutsiko du, langileak onartuz gero, interbentzioa ezabatuko da datu-basetik.

4.8 Taula: Laugarren Sprinta - Erabiltzailearen historiak

Bosgarren Sprinta

Bosgarren Sprintean 4.10 taulan ikus daitezkeen bost helburu beteko dira: nazionalitate bikoitza, erabiltzaile bat langile batek baino gehiagok, langile batek egoitza bat baino gehiago, langileak ikusi eta langileak ezabatu. Helburu hauek betetzeko aurreikusitako ordu kopurua 4.9 taulan erakusten da. Hurrengo irudian ikus daitezke Sprint honetan betetako erabilpen-kasuak.



4.6 Irudia: Bosgarren Sprinta - Erabilpen-kasuak

Helburua	Hasiera-data	Bukaera-data	Ordu kopurua
Nazionalitate bikoitza	2021/04/10	2021/04/07	5
Erabiltzaile batek langile bat baino gehiago	2021/04/07	2021/04/08	4
Langile batek egoitza bat baino gehiago	2021/04/08	2021/04/09	4
Langileak ikusi	2021/04/09	2021/04/10	2
Langileak ezabatu	2021/04/09	2021/04/10	2

4.9 Taula: Bosgarren Sprinta - Ordu-estimazioa

4. ANALISIA ETA DISEINUA

EHZ	Izena	Helburua	Baliozkotzea
21	Nazionalitate bikoitza	Langileak erabiltzaile bakoitzari nazionalitate bat edo gehiago emateko aukera izatea.	Erabiltzailearen interfazearen erabiltzailea gehitu atalean, nahi adina nazionalitate gehitzeko aukera emango da. Informazio hau datu-basera bidaltzean, nazionalitateak beste taula batean gordeko dira dagokion erabiltzailearen dokumentazio zenbakiarekin.
22	Erabiltzaile bat langile batek baino gehiagok	Langile batek, jadanik datu-basean existitzen den erabiltzaile bat gehitu nahi badu, inportatzeko aukera izatea.	Erabiltzailea gehitu formularioa betetzen ari denean langilea, jadanik existitzen den dokumentazio zenbaki bat sartuz gero, mezu bat agertuko da. Langileak, erabiltzailea inportatzeko aukera izango du, inportatuz gero, datu-basean gordeko da aipatutako langilearen eposta erabiltzailearen dokumentazio zenbakiarekin.
23	Langile batek egoitza bat baino gehiago	Langile bakoitzari egoitza bat edo gehiago esleitzeko aukera izatea.	Langile bat gehitu edo eguneratzean, nahi adina egoitza esleitzeko aukera izango da. Egoitza berri bat gehitzean, datu-basean gordeko da egoitzaren IDa eta langilearen eposta.
24	Langileak ikusi	Administratzaileak langile guztien zerrenda ikusteko aukera izatea.	Zerbitzariak datu-basetik langile guztien informazioa jaso eta erabiltzailearen interfazean erakutsiko da.
25	Langileak ezabatu	Administratzaileak langile bat ezabatzeko aukera izatea.	Langile zehatz bat hautatu eta hau ezabatzea, erabiltzailearen interfazeak alerta mezu bat erakutsiko du, administratzaileak onartuz gero, langilea ezabatuko da datu-basetik.

4.10 Taula: Bosgarren Sprinta - Erabiltzailearen historiak

Seigarren Sprinta

Seigarren sprintean 4.12 taulan ikus daitezkeen bost helburu beteko dira: menua osatu, kodea garbitu, E2E testak, test unitarioak eta integrazio-testak. Helburu hauek betetzeko aurreikusitako ordu kopurua 4.11 taulan erakusten da.

Helburua	Hasiera-data	Bukaera-data	Ordu kopurua
Menua osatu	2021/04/10	2021/04/07	5
Kodea garbitu	2021/04/07	2021/04/08	7
E2E Testak	2021/04/08	2021/04/09	10
Test unitarioak eta integrazio-testak	2021/04/09	2021/04/10	10

4.11 Taula: Seigarren Sprinta - Ordu-estimazioa

EHZ	Izena	Helburua	Baliozkotzea
26	Menua osatu	Langileak aplikazioko orrietan zehar menua erabiliz mugitzeko aukera izatea.	Erabiltzailearen interfazearen goialdean menu bat erakutsiko da, bertan langileak ikusi ditzakeen orrien zerrenda izango du, aplikazioan zehar nabigatzeko.
27	Kodea garbitu	SonarCloud-en sortzen diren erroreak ezabatu.	SonarCloud-en kodearen kalitatea neurtzen duen balioa ona izatea.
28	E2E testak	Ertzetik ertzerako testak egitea.	E2E-rako testak erabilpen-kasu guztiak zuzen egin daitezkeela bermatzea.
29	Test unitarioak eta integrazio-testak	Test unitarioak eta integrazio-testak egitea.	Test guztiak zuzenak izatea, eta estalduraren ehuneko gutxienez %75 izatea.

4.12 Taula: Seigarren Sprinta - Erabiltzailearen historiak

4.2.3 Erabilpen-kasuen diagrama

Aplikazioak izango dituen erabilpen-kasu guztiak, erabilpen-kasuen diagrama batean bildu dira. Diagraman aplikazioan izango diren funtzionalitateak erakutsiko dira, eta funtzionalitate bakoitza zein aktorek erabil dezakeen ere. 4.7 irudian ikus daitezke erabilpen-kasu guztiak irudi orokor batean.

Erabilpen-kasuen diagraman ikus daitekeen bezala, bi rol izango dira aplikazioan. Rol horiek, lehen aipatutako administratzailea eta langilea izango dira.

4.3 Gertaera-fluxuak eta sekuentzia-diagramak

Gertaera-fluxu bakoitzak erabilpen-kasu batek daraman prozesua azalduko du. Hurrengo atalean garrantzitsuak diren zenbait gertaera-fluxu azalduko dira. Gertaera-fluxuen ostean, sekuentzia-diagramak erakutsiko dira.

Erabiltzailea sortu

Aktorea:

Administratzailea eta langilea

Azalpena:

Langile bakoitzak nahi adina erabiltzaile sortzeko aukera izango du. Sortzen duen erabiltzaile bakoitzeko formulario bat beteko du, formularioko datu gehienak hautazkoak dira ez baitituzte beti datu berdinak behar erabiltzaile bakoitzeko.

Gertaera-fluxua:

- Sistemak erabiltzailea sortzeko pantaila erakutsiko dio langileari.
- Langileak erabiltzailearen inguruko datuak sartuko ditu.
- Langileak "Erabiltzailea gehitu" botoia sakatuko du.

Ordezko fluxua:

- Erabiltzailearen NA zenbakia datu-basean baldin badago dagoeneko, abisu bat eman eta erabiltzailea inportatzeko aukera emango da.
- Gutxienerako datuak sartu ez badira, errore bat emango da.
- Erabiltzailearen telefono-zenbakia datu-basean existitzen bada, abisu bat emango zaio langileari.

Interfazea:

Erabiltzailea sortzeko erabili den interfazea 4.8 irudian ikus daiteke.

4. ANALISIA ETA DISEINUA

hasita izatea.

Sekuentzia-diagraman ikus daitekeen bezala, orria kargatzearekin batera, GET motako HTTP eskari bat bidaliko da egoitzaren kontrolatzailerara. Kontrolatzaileak datu-basetik egoitzen zerrenda bat lortuko du, eta datu hori itzuliko da erabiltzailea gehitu interfazera. Horren ostean, langileak datu zehatz batzuk sartuko ditu formularioan. Datu horiek kontrolatzailerara bidaliko dira POST motako HTTP zenbait eskaeren bitartez.

Lehenengo eskaera telefono zenbakia existitzen den begiratzeko izango da, *CheckTelefono(telefono)* funtzioa erabiliko da honetarako, eta erabiltzaile kontrolatzaileak datu-basean bilatuko du ea badagoen erabiltzailerik zenbaki horrekin. Zenbakia existituko balitz *true* boolearra itzuliko du, bestela *false* itzuliko da. Boolear honen arabera alerta-mezu bat erakutsiko da.

Ondoren, erabiltzailea gehituko da datu-basera. Honetarako erabiltzailearen kontrolatzailerara datu guztiak bidaliko dira *CreateUser(userData)* funtzioaz baliatuz. Kontrolatzaileak erabiltzaile eta nazionalitate modeloak eskaintzen dituzten metodoak erabiliko ditu. Metodo hauen bitartez erabiltzailea gehituko du datu-basera, hau egin ostean, erabiltzaileak dituen nazionalitate guztiak gehituko ditu *Nazionalitate* taulara.

SOS Arrazakeria Erabiltzaileak ▾ Egutegia Kasuak ▾ Langileak ▾ 🌐 👤 Nire profila ▾

Erabiltzailea gehitu

Informazio pertsonala

Izena	Abizena	Abizena 2
<input type="text" value="Izena"/>	<input type="text" value="Abizena"/>	<input type="text" value="Abizena 2"/>
Dokumentazio mota	Dokumentazio zenbakia	Generoa
<input type="text" value="DNI"/>	<input type="text" value="11223344J"/>	<input type="text" value="Emakumea"/>
Egoitza	Nork atenditu zaitu	
<input type="text" value="Azpeltia"/>	<input type="text" value="Langile Izena2"/>	

Kontakturako datuak

Eposta	Telefona
<input type="text" value="example@example.com"/>	<input type="text" value="666111222"/>

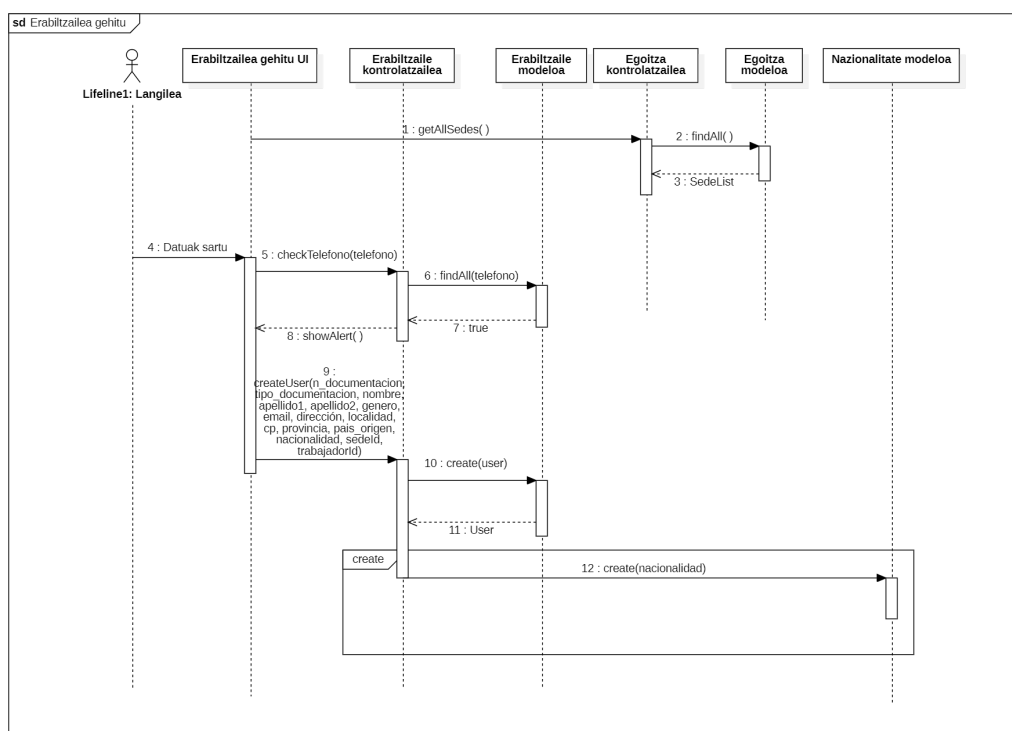
Helbidea

Helbidea	PK	Herria
<input type="text"/>	<input type="text"/>	<input type="text"/>
Probintzia	Jatorrizko herrialdea	Nazionalitatea
<input type="text" value="Gulpuzcoa"/>	<input type="text" value="España"/>	<input type="text" value="España"/>

Nazionalitatea gehitu

4.8 Irudia: Erabiltzailea gehitu interfazea

4.3. Gertaera-fluxuak eta sekuentzia-diagramak



4.9 Irudia: Erabiltzailea gehitu, sekuentzia-diagrama

Erabiltzaileak ikusi

Aktorea:

Administratzailea eta langilea

Azalpena:

Langile bakoitzak sortu dituen erabiltzaileak ikusteko aukera izango du. Hasieran sortu dituen erabiltzaile guztien zerrenda bat izango du, ondoren, iragazkiak erabiliz zerrenda hori iragazteko aukera izango du.

Gertaera-fluxua:

- Langile edo administratzaileak erabiltzaileak ikusteko leihoa hautatuko du.
- Sistemak erabiltzaile guztien zerrenda erakutsiko dio langileari.

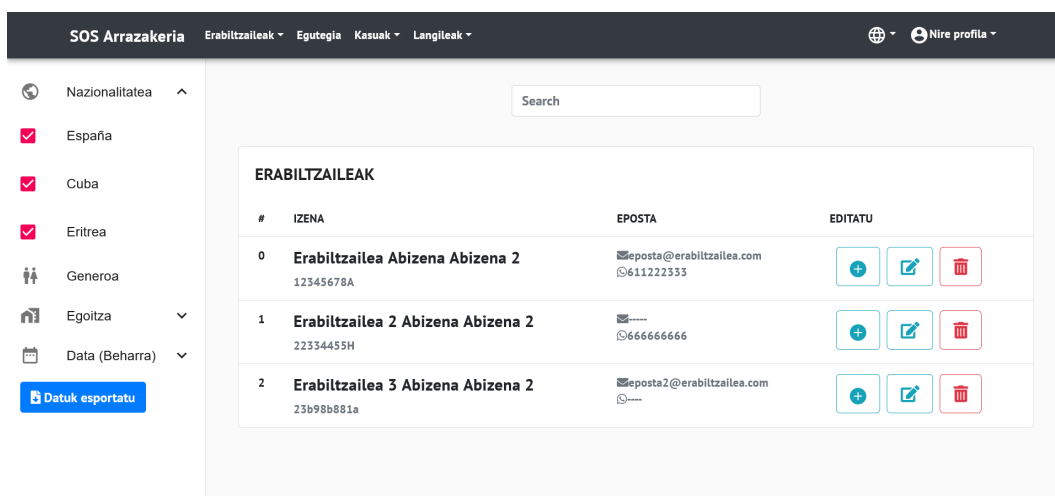
Interfazea:

Erabiltzaileak ikusteko erabili den interfazea 4.10 irudian ikus daiteke.

Sekuentzia-diagrama: 4.11 irudiak “Erabiltzaileak ikusi” kasuaren sekuentzia-diagrama erakusten du. Akzio hori betetzeko, beharrezkoa izango da langile edota administratzaileak kontu bat izan eta saioa hasita izatea.

Sekuentzia-diagraman ikus daitekeen bezala, orria kargatzearekin batera, GET motako bi HTTP eskari bidaliko dira. Lehenengoa Erabiltzaileen kontrolatzaileera joango da,

4. ANALISIA ETA DISEINUA



The screenshot shows the 'ERABILTZAILEAK' (Users) section of the 'SOS Arrazakeria' application. The interface includes a search bar and a list of users with the following data:

#	IZENA	EPOSTA	EDITATU
0	Erabiltzailea Abizena Abizena 2 12345678A	eposta@erabiltzailea.com 611222333	[+] [✎] [🗑️]
1	Erabiltzailea 2 Abizena Abizena 2 22334455H	----- 666666666	[+] [✎] [🗑️]
2	Erabiltzailea 3 Abizena Abizena 2 23b98b881a	eposta2@erabiltzailea.com -----	[+] [✎] [🗑️]

4.10 Irudia: Erabiltzaileak ikusi

getAllUsers(token) metodoa deituko da, eta langilearen identifikazio-tokena bidaliko da. Kontrolatzaileak erabiltzaile modeloak eskaintzen duen metodoa erabiliko du. Metodo horrek lehenengoz langilearen token-a zuzena den zehaztuko du. Token hori zuzena izanez gero, *verifyToken* funtzioak langilearen eposta itzuliko du. Epostaz baliatuz, datu-basean SELECT motako agindu bat exekutatu da, langilearen erabiltzaile guztiak itzuliko dituena.

Bigarren eskaria erabiltzaileen kasuak lortzeko izango da, eta funtzionamendua aurrekoaren berdina izango da. *getAllCasos(token)* metodoa deituko da, eta horrek kasuen modeloaz baliatuz token-a zuzena dela egiaztatuko du. Horrela izanez gero, langileak sartutako kasu guztiak itzuliko ditu.

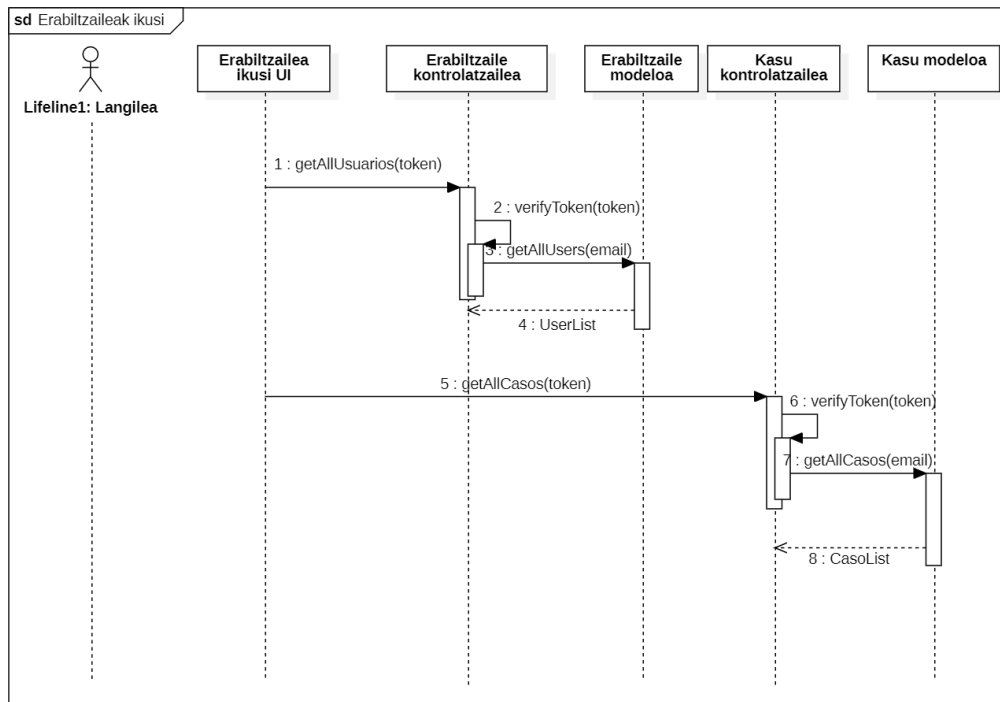
Kasua sortu

Aktorea: Administratzailea eta langilea

Azalpena: Langile bakoitzak, berak sortutako erabiltzaile bati kasu bat sortzeko aukera izango du. Sortzen duen kasua hiru motatakoa izan daiteke: etxeko langilea, atzerritarra ala diskriminazioa. Langileak hiru kasu horietako bat hautatu beharko du eta bertako datuak bete kasua sortu ahal izateko.

Gertaera-fluxua:

- Sistemak erabiltzaileen zerrenda erakutsiko dio langileari.
- Langileak erabiltzaile bat hautatu beharko du, berari kasu bat gehitzeko.
- Sistemak kasua sortzeko pantaila erakutsiko dio langileari.
- Langileak sartu nahi duen kasu mota hautatuko du: atzerritartasuna, diskriminazioa edo etxeko langilea.
- Sistemak kasuaren formularioa erakutsiko du.
- Langileak kasuaren inguruko datuak sartuko ditu.



4.11 Irudia: Erabiltzaileak ikusi, sekuentzia-diagrama

- Langileak “Kasua sortu” botoia sakatuko du.

Ordezko fluxua:

- Langileak gutxieneko datuak sartzen ez baditu, errore bat erakutsiko da.

Interfazea: Kasua sortzeko erabili den interfazea 4.12 irudian ikus daiteke.

Sekuentzia-diagrama: 4.13 irudiak “Kasua sortu” sekuentzia-diagrama erakusten du. Akzio hau aurrera eramateko, beharrezkoa izango da langile edo administratzaileak kontu bat izan eta saioa hasita izatea. Gainera beharrezkoa izango da gutxienez erabiltzaile bat existitzea, erabiltzaile bati lotua baitoa kasua.

Sekuentzia-diagraman ikus daitekeen bezala, orria kargatzearekin batera, GET motako HTTP eskari bat bidaliko da erabiltzailearen kontrolatzaileera. Kontrolatzaileak datu-basetik erabiltzaile zehatz bat lortuko du, eta datu hau itzuliko da kasua gehitu interfazera. Horren ostean, langileak datu zehatz batzuk sartuko ditu formularioan. Datu horiek kontrolatzaileera bidaliko dira zenbait POST motako HTTP eskaeren bitartez.

Langileak atzerritartasun formularioa bete badu, *CreateExtranjeria(necesidadList, proyectosList)* funtzioa erabiliko da. Kasuen kontrolatzaileak lehenengoz kasuen modeloko *create* metodoa erabiliko du, eta horrek sortu berri den kasuaren ID-a itzuliko du. Ondoren, ID hori erabiliz, atzerritartasun modeloak eskaintzen duen *create* metodoa erabiliz, *Atzerritartasun* taulan beharrezko datuak sartuko dira. Bukatzeko, proiektuen eta beharren zerrendetako balioak datu-basera sartuko dira begizta baten bitartez.

4. ANALISIA ETA DISEINUA

SOS Arrazakeria Erabiltzaileak Egutegia Kasuak Langileak 🌐 👤 Nire profila

Kasua gehitu : Erabiltzailea2 Abizena

Etxeko langilea↓

Diskriminazioa ↓

Dokumentu egoera Etxe egoera Ikasketa-maila

Nazionalitate espainiarra Etxeduna Ikasketarik gabe

Azal kolorea / Etnia / Ezaugarri fenotipikoak

Mota

Gatazka eta eraso arrazistak

Gatazka eta eraso arrazistak

Eremu publikoan

Adierazi beste batzuk:

Agente diskriminatzailea Gertaeren data Herria

Erakunde publikoa dd/mm/aaaa Donostia

Gertaeren kontakizuna

Agente diskriminatzailea Testigantza eman dezaketen pertsonak Jarraitu beharreko estrategia

Asumitu

Geure gain hartuko dugu kasua

Bide penala

Beste froga-elementu batzuk

Kasua gehitu

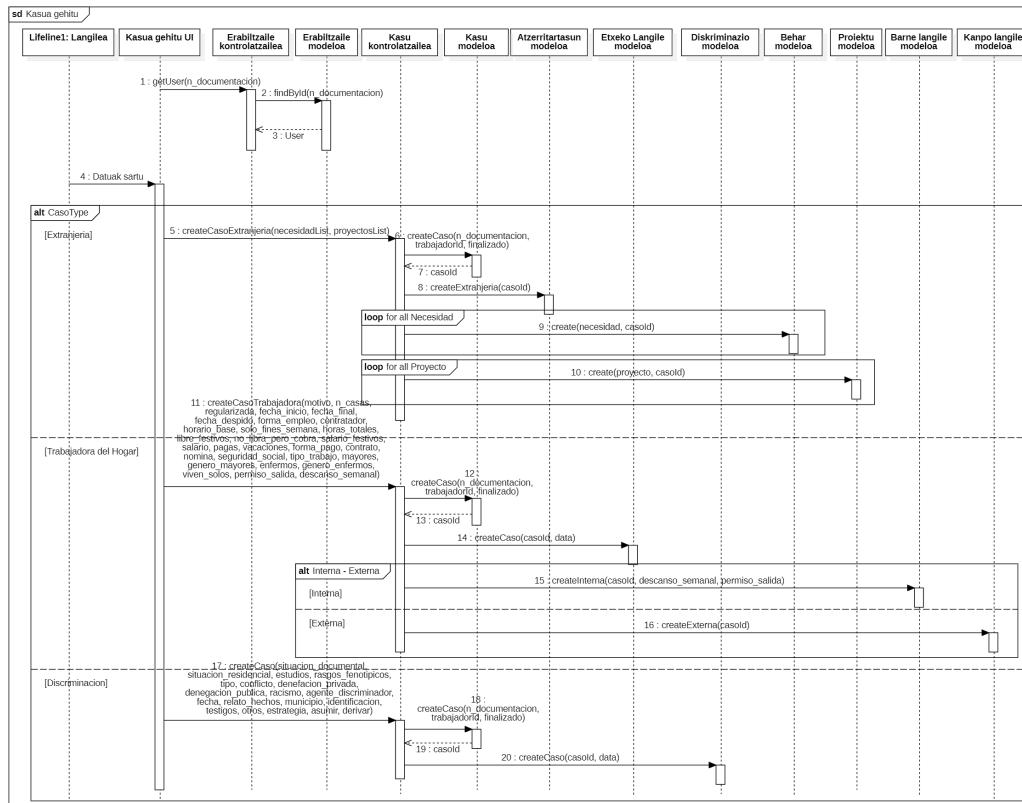
Atzeritartasuna ↓

4.12 Irudia: Diskriminazio motako kasua gehitu interfazea

Langileak ordea etxeko-langile formularioa bete badu, *CreateTrabajadoraHogar(data)* funtzioa deituko da. Honek atzeritartasuneko kasuaren pausu berak jarraituko ditu, behin kasuaren ID-a edukita, etxeko-langile modeloak eskaintzen duen *create* metodoa erabiliz, beharrezko datu guztiak datu-basean sartuko dira. Bukatzeko, hasierako datuetan *asteko deskantsua* eta *irteteko baimena* datuak definituta badaude, informazio hau *Barne Langile* taulan sartuko da, bestela, kasuaren ID-a *Kanpo Langile* taulan sartuko da.

Bukatzeko, langileak diskriminazioaren formularioa bete badu, aurreko kasuen antzera, *CreateDiscriminacion(data)* funtzioa erabiliko da. Honek kasuaren ID-a lortu, eta diskriminazio modeloaren *create* metodoa erabiliko du informazio guztia diskriminazioari dagokion taulan sartzeko.

4.3. Gertaera-fluxuak eta sekuentzia-diagramak



4.13 Irudia: Kasua sortu, sekuentzia-diagrama

Langilea sortu

Aktorea:

Administratzailea

Azalpena:

Administratzaileak langile berri bat sortzeko aukera izango du. Horretarako formulario bat bete beharko du langile berriaren datuekin.

Gertaera-fluxua:

- Sistemak langilea sortzeko pantaila erakutsiko dio administratzaileari.
- Administratzaileak langile berriaren datuak sartuko ditu.
- Administratzaileak “Langilea gehitu” botoia klikatuko du.
- Sistemak pasahitz bat sortuko du, eta langileari bidaliko dio epostaz.
- Sistemak langile berria gehituko du datu-basera.

Ordezko fluxua:

- Sartutako eposta datu-basean existitzen bada, errore bat erakutsiko da.

- Gutxienerako datuak sartzen ez badira, errore bat erakutsiko da.

Interfazea:

Langilea sortzeko erabili den interfazea 4.14 irudian ikus daiteke.

Langilea gehitu

Izen-abizenak

Eposta

Egoitza

×

+ Egoitza gehitu

Administratzailea da

Kolorea:

Langilea gehitu

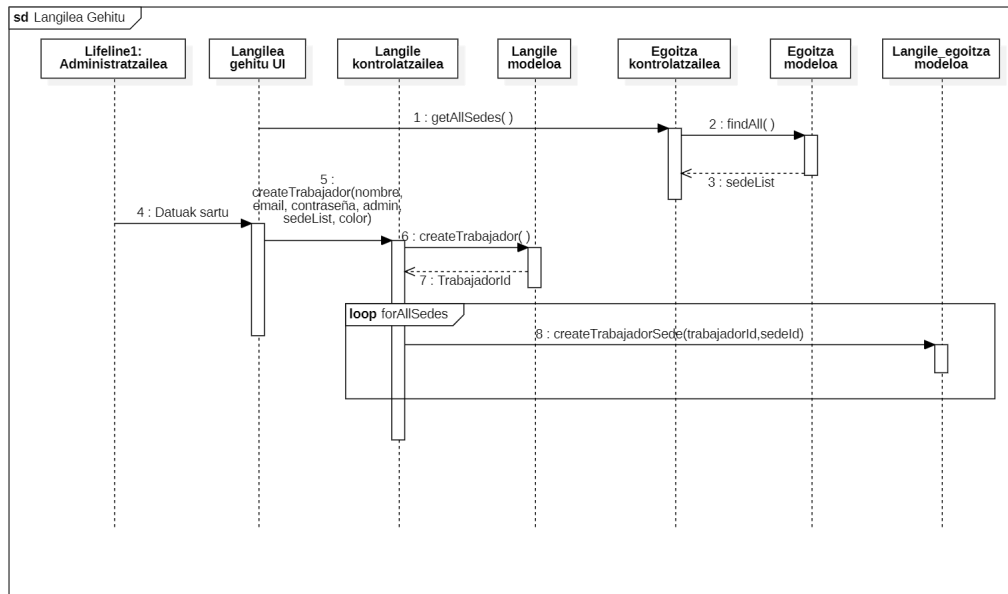
4.14 Irudia: Langilea gehitu interfazea

Sekuentzia-diagrama:

4.15 irudiak “Langilea gehitu” kasuaren sekuentzia-diagrama erakusten du. Akzio hau betetzeko, beharrezkoa izango da administratzaileak saioa hastea. Langileak ez du hau egiteko baimenik izango.

Sekuentzia-diagraman ikus daitekeen bezala, orria kargatzearekin batera, GET motako HTTP eskari bat bidaliko da egoitzaren kontrolatzailera. Kontrolatzaileak modeloari deia eginez, egoitzen zerrenda bat lortuko du eta datu hau itzuliko da langilea gehitu interfazera. Langileak formularioan eskatzen diren datuak sartu beharko ditu, eta datu hauek langilearen kontrolatzailera bidaliko dira POST motako HTTP eskaeraren bitartez.

Langilearen kontrolatzaileak, hasteko, langilearen modelora deia egingo du, datu-basean langile berriaren datu guztiak gehitzeko. Datuak ondo gehitu badira, langile berriaren datuak itzuliko dira, bestela errore mezu bat emango da. Langilea ondo gehitu bada, langilearen egoitza guztiak gehituko dira datu-basera. Horretarako, Langile_egoitza modelora bidaliko dira datuak eta honek datu-basean gehituko ditu. Ondo gehituz gero, arrakasta mezu bat itzuliko da, bestela, errore mezu bat izango da itzultzen dena.



4.15 Irudia: Langilea gehitu, sekuntzia-diagrama

4.4 Datu-basearen diseinua

4.4.1 Deskribapena

- Erabiltzaile bakoitza identifikatzeko bere NAa erabiliko da. Horrez gain, izen osoa (izena eta bi abizenaz osatua), telefonoa, generoa, emaila, helbidea eta nazionalitatea gordeko dira.
- Erabiltzaile bakoitzak kasu bat edo gehiago izango ditu. Kasu bakoitzak kasu zenbakia izango du identifikatzeko, horrez gain kasua itxi den ala ez gordeko da. Kasuetan hiru multzo nagusi bereiziko dira: etxeko langilea, atzerritarra, diskriminazioa. Kasu partikular bakoitza multzo bakarrean defini daiteke.
- Etxeko langileen kasuan atributu asko gordeko dira (atributuak 4.16 diagraman baino ez dira jasoko). Bi etxeko langile mota bereiziko dira: barne-langilea eta kanpo-langilea. Gainera, barne-langileentzat atributu hauek gordeko dira: astean zehar duen atsedena eta ateratzeko duen baimena.
- Atzerritarren kasuan, zein proiektura lotuko diren eta zein behar dituzten gordeko dira. Hauek atributu balio aniztunak dira.

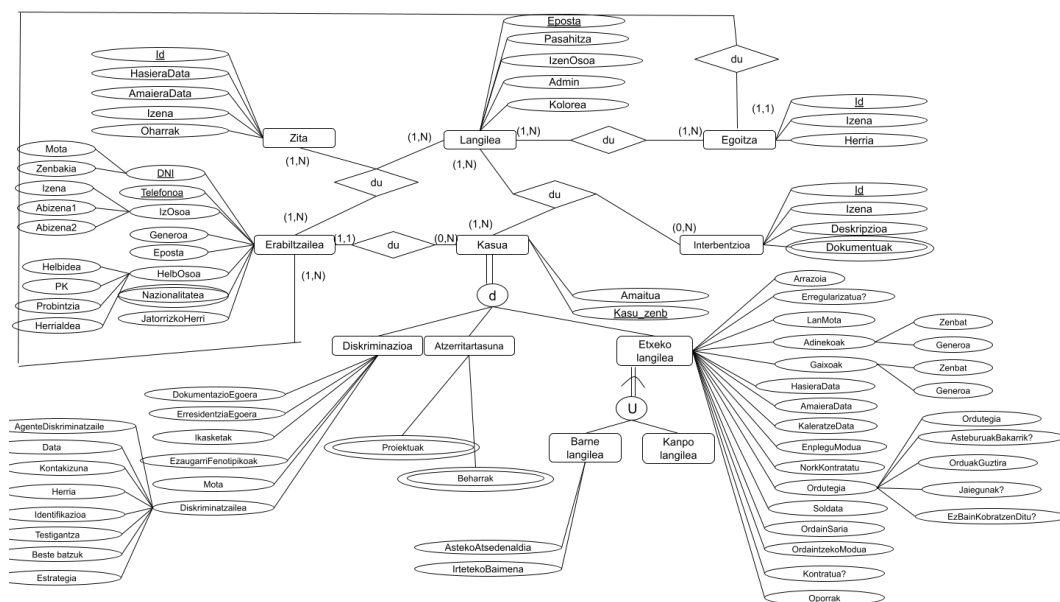
4. ANALISIA ETA DISEINUA

- Diskriminazioaren kasuan, atributu asko gordeko dira (atributu hauek 4.16 diagraman baino ez dira jasoko).
- Kasu bakoitzak interbentzio bat edo gehiago izango ditu. Interbentzio bakoitzak ID identifikatzaile bat edukiko du, horrez gain izena, deskripzioa, kasu-zenbakia, eta dokumentuak gordeko dira.
- Egoitzak IDarekin identifikatuko dira, eta egoitza bakoitza langile bat baina gehiagori lotuta egon daiteke. Egoitza bakoitzeko haren izena eta herria gordeko dira. Gainera, egoitza bakoitzak bertako erabiltzaileak ere edukiko ditu.
- Langile bakoitza bere emailez identifikatuko da. Horrez gain, pasahitza, izen osoa eta lanpostua, administratzaile den ala ez eta kolore bat ere gordeko dira. Langile bakoitzak bere zitak edukiko ditu. Langile bakoitzak egoitza bat edo gehiago izan ditzake.
- Zita bere IDarekin identifikatuko da. Horrez gain hasiera-data, bukaera-data, izen bat eta deskribapen bat gordeko dira. Zita bakoitza erabiltzaile bati eta langile bati lotua egongo da.
- Datu-basean gordetako datu guztientzat, noiz sortu eta azkenengo aldiz noiz eguneratu den gordeko da CreatedAt eta UpdatedAt atributueta.

4.4.2 Diagrama

Azpiatal honetan, proiektuko datu-baseak jasotzen dituen entitateak erakutsiko dira, baita entitate hauen arteko erlazioak ere eta entitate bakoitzak izango dituen atributuak ere.

Kasuen atributuak zehazteko, eranskinetan erakusten diren [Barne-langilearen fitxa](#), [Kanpo-langilearen fitxa](#), [Atzerritartasun formularioa](#) formularioak erabili dira.



4.16 Irudia: Entitate-erlazio eredu hedatua

4.4.3 Taulak

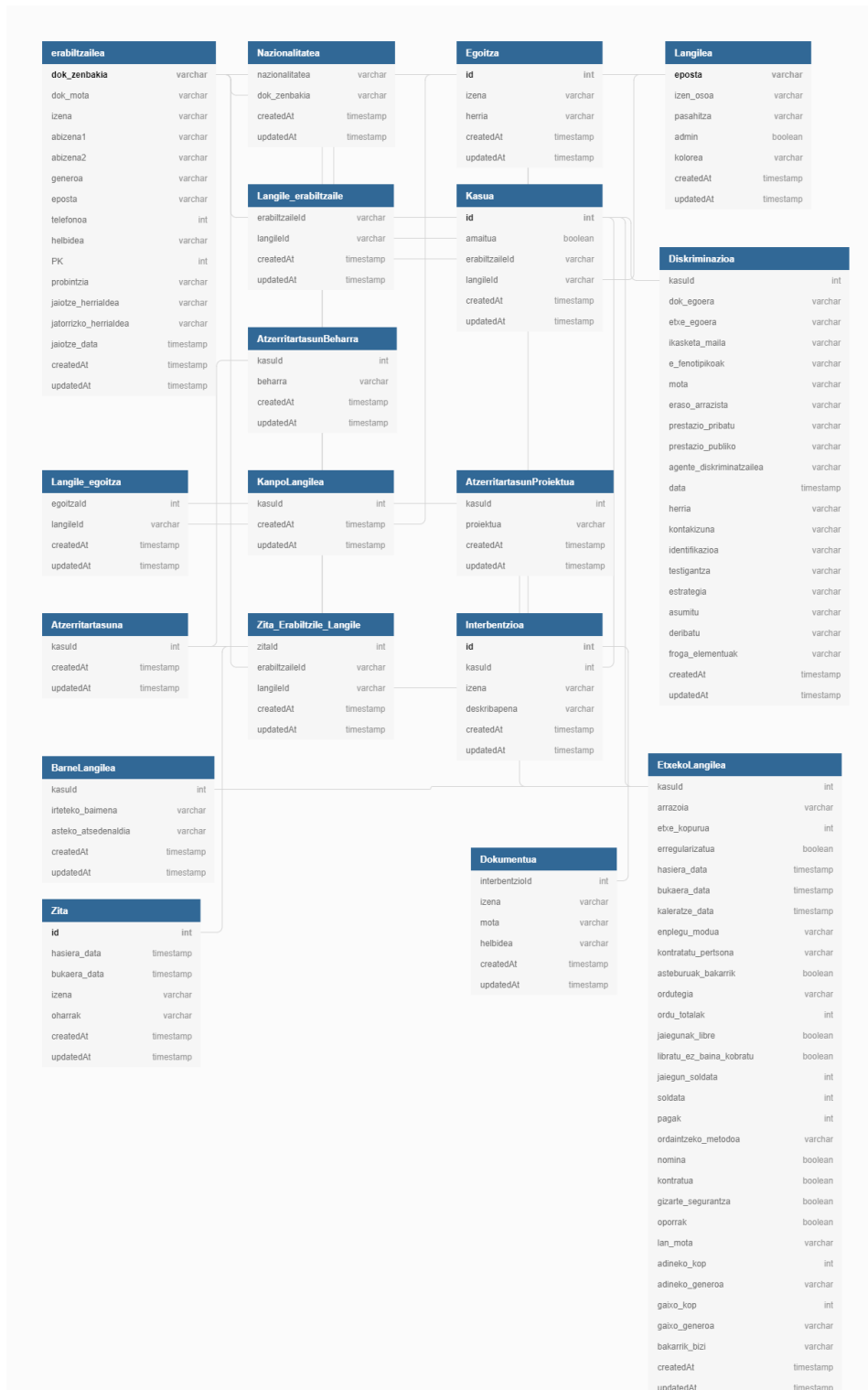
Atal honetan datu-basean sortu diren taulak azalduko dira, taula bakoitzak edukiko dituen atributuak eta gakoak erakutsiz. Aipatuko diren taula guztiek `createdAt` eta `updatedAt` atributuak izango dituzte, noiz sortu ziren eta azken eguneraketa datak gordeko dira bertan.

- **Egoitza:** Egoitza taulak honako atributuak izango ditu: IDa, izena eta herria. Identifikazio zenbakia (ID) automatikoki gehituko den zenbaki bat izango da.
- **Langilea:** Langilea bere epostaz identifikatuko da, horrez gain, izen osoa, eposta, pasahitza, administratzailea den eta kolore bat gordeko dira.
- **Langile_egoitza:** Taula honetan, langilea eta egoitzaren arteko erlazioa gordeko da. Hau da, egoitza guztietan dauden langile guztien zerrenda gordeko da. Izango dituen atributuak egoitzaren IDa eta langilearen eposta izango dira.
- **Erabiltzailea:** Erabiltzailea NA zenbakiaz identifikatuko da. Horrez gain, hurrengo atributuak gordeko dira: dokumentazio mota, izena, lehenengo abizena, bigarren abizena, generoa, eposta, telefono zenbakia, helbidea, herria, posta kodea, bizileku-herrialdea, jatorrizko herrialdea eta jaiotze data. Gainera, erabiltzaileari dagokion egoitza gordeko da.
- **Langile_erabiltzaile:** Taula honetan, langilea eta erabiltzailearen arteko erlazioa gordeko da. Hau da, langile bakoitzak dituen erabiltzaile guztien zerrenda gordeko da. Horretarako erabiltzailearen NA zenbakia eta langilearen eposta zenbakia gordeko dira. Erabiltzailea edo langilea ezabatuz gero, taula honetatik haien balioak ere ezabatuko dira.
- **Nazionalitatea:** Nazionalitatea taulan, erabiltzaileak dituen nazionalitate guztiak gordeko dira. Horretarako, nazionalitatea eta erabiltzailearen NA zenbakia gordeko dira. Erabiltzailea ezabatuz gero, hari dagozkion nazionalitateak ere ezabatuak izango dira.
- **Kasua:** Taula honetan, kasuaren identifikazio zenbakia gordeko da gako arrunt bezala. Hau automatikoki gehituko den zenbakia izango da. Gainera, kasua amaitu den ala ez adierazten duen balio boolearra gordeko da. Gako arrotzak, kasua gehitu duen langilearen eposta eta erabiltzailearen NA zenbakia dira.
- **Diskriminazioa:** Diskriminazio taulan, kasuaren IDa gordeko da gako arrotz gisa. Atributuak honako hauek izango dira: dokumentazio egoera, etxe egoera, ikasketamaila, ezaugarri fenotipikoak, diskriminazio mota, eraso arrazista, prestazio eta zerbitzu pribatuen sarbidea ukatzea, prestazio eta zerbitzu publikoen sarbidea ukatzea, agente diskriminatzailea, gertaeraren data, herria, gertaeraren kontakizuna, identifikazioa, testigantza eman dezaketen pertsonak, jarraitutako estrategia, onartu, deribatu eta beste froga elementu batzuk gordeko dira. Kasua ezabatuz gero, diskriminazio taulako gako arrotz bera duen balioa ere ezabatuko da.
- **Atzerritartasuna:** Taula honetan, kasuaren identifikazio zenbakia gordeko da. Kasua ezabatuz gero, atzerritartasun taulako gako arrotz bera duen balioa ere ezabatuko da.

- **AtzeritartasunBeharra:** Atzeritartasun kasuaren beharrak gordetzeko taulan, atzeritartasun kasuaren ID zenbakia gordeko da. Horrez gain, beharraren titulua ere gordeko da. Atzeritartasun kasua ezabatuz gero, hari dagokion beharra ere ezabatuko da.
- **AtzeritartasunProiektua:** Atzeritartasun kasuaren proiektuak gordetzeko taulan, atzeritartasun kasuaren ID zenbakia gordeko da. Horrez gain, proiektuaren titulua ere gordeko da. Atzeritartasun kasua ezabatuz gero, hari dagokion proiektua ere ezabatuko da.
- **Etxeko_Langilea:** Etxeko langileen taulan, dagokion kasuaren ID zenbakia gordeko da. Kasua ezabatuz gero, etxeko langile taulako balioa ere ezabatua izango da. Gordeko diren atributuak hurrengoak dira: kontsultaren arrazoia, etxe kopurua, erregularizatua, lanaren hasiera-data, lanaren amaiera data, kaleratze data, enplegua lortzeko modua, kontratatu zuen pertsona, asteburuetan bakarrik egiten duen lan, ordutegia, asteko ordu totalak, jaieguna libratzen dituen, libratu ez baina kobratzen dituen, zenbat kobratzen duen jaieguna, soldata, pagak, ordaintzeko metodoa, nomina duen, kontratua duen, gizarte-segurantza duen, oporrak dituen, lan mota, adinekoen kopurua eta generoa, gaixo kopurua eta generoa eta bukatzeko, bakarrik bizi diren.
- **BarneLangilea:** Barne langileak, etxeko langilearen kasuaren ID zenbakia gordeko du gako arrotz bezala. Etxeko langilearen kasua ezabatuz gero, taula honetako balioa ere ezabatuko da. Gordeko diren atributuak irteteko baimena eta asteko atsedenaldua izango dira.
- **KanpoLangilea:** Kanpo langileak, etxeko langilearen kasuaren ID zenbakia gordeko du gako arrotz bezala. Etxeko langilearen kasua ezabatuz gero, taula honetako balioa ere ezabatuko da.
- **Interbentzioa:** Interbentzio bakoitzeko identifikazio zenbaki bat gordeko da. Gako arrunt hau, automatikoki gehituko den zenbaki bat izango da. Dagokion kasuaren ID zenbakia gordeko da gako arrotz gisa, kasua ezabatuz gero, interbentzioa ere ezabatu egingo da. Interbentzio bakoitzeko gordeko diren atributuak izena eta deskribapena izango dira.
- **Dokumentuak:** Taula honetan, interbentzio bakoitzarekin igotzen diren dokumentuen informazioa gordeko da. Gako arrotza interbentzioaren ID zenbakia da, eta interbentzioa ezabatzean, dokumentuak ere ezabatuko dira. Horrez gain, dokumentuaren izena, dokumentu mota eta gordeta dagoen karpeta helbidea gordeko dira.
- **Zita:** Zita bakoitzak gako arrunt bat izango du: ID zenbakia. Hau automatikoki gehituko den zenbaki bat izango da. Horrez gain, zenbait atributu gordeko dira: hasiera-data, bukaera-data, izena eta oharrak.
- **Zita_erabiltzaile_langile:** Taula honetan, zita, erabiltzailea eta langilearen arteko erlazioa gordeko da. Horretarako, hiru gako arrotz gordeko dira: langilearen eposta, erabiltzailearen NA zenbakia eta zitaren identifikazio zenbakia.

4.5 Domeinuaren eredia

Aurreko atalean azaldutako taulen domeinu-eredua erakusten da 4.17 irudian.



4.17 Irudia: Domeinuaren eredia

Soluzioaren garapena

Aurreko atalean azaldu bezala, aplikazioaren funtzionamenduak MVC patroia jarraituko du. Atal honetan, aipatutako patroia inplementazioa erakutsiko da. Hiru zatitan banatuko da: modeloa, bista eta kontrolatzailea. Bista erabiltzailearen aldean egongo da, modeloa eta kontrolatzailea ordea zerbitzarian alojatuko dira. Kodearen egitura hobeto ulertzeko, zenbait kode zati erakutsiko dira.

5.1 Kodearen egitura

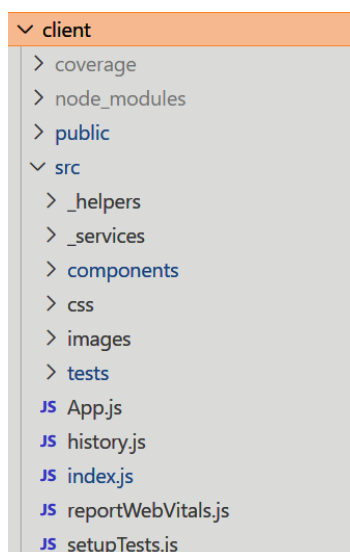
5.1.1 Bistak

Bistak JavaScript, CSS eta JSX erabiliz idatzi dira. JSX (JavaScript Syntax Extension), React-eko elementuak sortzeko erabiltzen den lengoia da, JavaScripten sintaxiaren luzapen bat. HTML-ren itxura antzekoa du. Bistak erabiltzailearen aldean egongo dira, eta erabiltzailearen interfazea sortzeaz arduratzen dira.

Aplikazioak interfaze eleaniztuna eskainiko du, euskaraz eta gaztelaniaz. Honetarako, *i18next* [16] liburutegia erabili da, erabiltzaileak aukeratutako hizkuntza detektatu eta itzulpenak kargatzeaz arduratuko dena.

JSX erabiliz bistaren oinarria sortu da, hau da, nola dagoen antolatua eta duen testua. CSS eta Bootstrap erabiliz, itxura eman diogu aipatutako oinarriari. Horrela, aplikazioak itxura erakargarria izatea lortu dugu besteak beste, elementuen tamainak, koloreak eta tipografiak aldatuz. Amaitzeko, JavaScript erabiliz, dinamismoa gehitu diogu aplikazioari (alertak, mugimenduak, trantsizioak ...). Horrez gain, zerbitzariarekin konexio asinkronoa ezarri da. Konexio asinkronoak, erabiltzailearen aldeak eta zerbitzariaren aldeak datuak elkarbanatzeko aukera ematen du. Gainera, ez da beharrezkoa orri guztia kargatzea datu hauek bidali edo jasotzen direnero, datuak erakutsiko dituen zatia bakarrik eguneratuko da.

5.2 irudian ikus daitekeen bezala, *axios* [17] JavaScript-eko liburutegia erabili da zerbitzariarekin konexioak ezartzeko. Liburutegi honek errazten du zerbitzariari deiak egin eta erraz prozesatzeko emaitzak jasotzea. Aipatutako irudian ikus daitekeen funtzioa, erabiltzaile bat sortzeko funtzioa izango da. Horrek formularioan sartutako datuak jaso eta zerbitzariaren aldera bidaliko ditu. Jasotako emaitza zuzena bada, *open* aldagaiari *true*



5.1 Irudia: Aplikazioaren bistak

```

const createUser = (data,e) => {
  data = {
    ...data,
    nacionalidad: nacionalidades
  }
  axios.post('http://localhost:8080/usuario/create', { data }).then(res => {
    setOpen(true);
  }).catch(error => {
    setError(true)
  })
  e.preventDefault();
  e.target.reset();
}

```

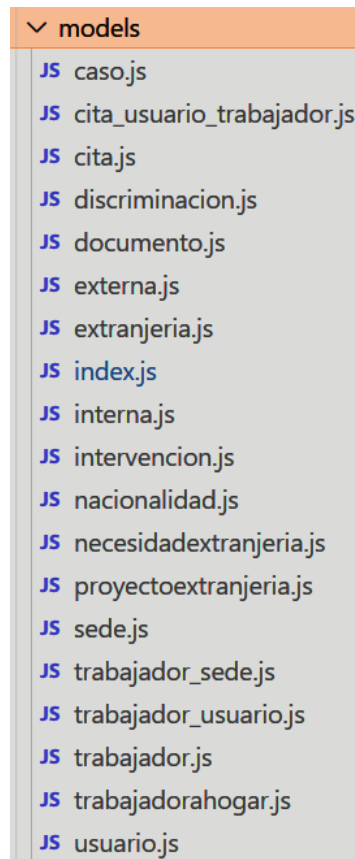
5.2 Irudia: Konexio asinkronoak zerbitzariarekin

balioa emango dio, bestela *error* aldagaiari *true* balioa emango dio. Aldagai horien arabera, errore mezu bat edo arrakasta mezu bat erakutsiko da. Ondoren, *e.target.reset()* deituz, formularioko input-ak garbituko dira, berriro hutsa egoteko beste erabiltzaile bat gehitu ahal izateko.

5.1.2 Kontrolatzaileak eta modeloak

Zerbitzariaren aldean egongo dira MVC arkitekturako kontrolatzailea eta modeloak osatzen dituzten fitxategiak. 5.3 eta 5.4 irudietan, proiektuaren modeloak eta kontrolatzaileak erakusten dira. Lehenago aipatu bezala, kontrolatzaileak erabiltzailearen ordenak jaso, eta datuak eskatuko dizkio modeloari. Datu horiek, ondoren, bistara bidaliko dira. Modeloak ordea, datuak eguneratu, kontsultatu eta gehituko ditun datu-base baten bidez.

Kontrolatzailea eta modelo garatzeko, Sequelize [18] erabili da. Sequelize, Node.js-rako ORM (Objektu-Erlazional Mapaketa) da, hau, objektuetara bideratutako programazio-lengoaia baten klaseen eta datu-base erlazional baten taulen artean datuak bihurtzeko



5.3 Irudia: Modeloak



5.4 Irudia: Kontrolatzaileak

programazio-teknika bat da. Sequelize erabiliz sortu dira aplikazio honetako modeloak eta kontrolatzaileak.

Modelo bat, datu-baseko taula bat irudikatzen duen abstrakzio bat da. Sequelizen, modeloa klasearen eratorria izango da. Modeloak, Sequelize-ri irudikatzen duen entitatearen zenbait datu emango dizkio, besteak beste, datu-baseko taularen izena, dituen zutabeak eta zutabe horien datu-motak.

5.5 irudian ikus daiteke Zita modeloa. Irudian agertzen den bezala, zita-ren zutabeak hasieratu ditugu bertan. Lau zutabe izango dira: hasiera data (DATA motakoa), amaiera data (DATA motakoa), izena (STRING motakoa) eta notak (STRING motakoa).

```

module.exports = (sequelize, DataTypes) => {
  class Cita extends Model {
    static associate(models) {
    }
  }
};
Cita.init({
  fechaInicio: DataTypes.DATE,
  fechaFin: DataTypes.DATE,
  nombre: DataTypes.STRING,
  notas: DataTypes.STRING
}, {
  freezeTableName: true,
  sequelize,
  modelName: 'Cita',
});
return Cita;
};

```

5.5 Irudia: Modeloa: Zita

Kontrolatzaileak, modeloak eskaintzen dituen funtzioak erabiliko ditu datu-basera datuak gehitu, kontsultatu, eguneratu edota ezabatzeko. Behin modeloak eskaintako funtzioa erabilita, honen emaitza aztertu eta mezu bat itzuliko du bistara. Mezu horrek testu-kate bat edota datu array edo objektu bat eduki dezake.

5.6 irudian, zita modeloaren kontrolatzaileko *DeleteCita* funtzioa erakusten da. Funtzio honen helburua, zita baten IDa jaso eta zita hau datu-basetik ezabatzea izango da. Horretarako, lehenengo zitaren IDa gordeko du *citaId* aldagaian. Ondoren, *Cita.destroy()* metodoa erabiliko du zita ezabatzeko. Metodo honek 1 balioa itzultzen badu, zita ondo ezabatu delako da. Kasu horretan, arrakasta mezu bat bidaliko du. Bestela, errore-mezu bat bidaliko du, zita ezin izan dela ezabatu azalduz.

```

exports.deleteCita = (request, result) => {
  const citaId = request.params.id;
  Cita.destroy({
    where: {
      id: citaId
    }
  }).then(num => {
    if (num === 1) {
      result.send({
        message: "La cita ha sido eliminado correctamente."
      });
    } else {
      result.status(500).send({
        message: `No se ha podido eliminar la cita con id=${citaId}!`
      });
    }
  }).catch(err => {
    result.status(500).send({
      message: err.message || `Ha habido algun error eliminando la cita con id=${citaId}!`
    });
  });
};
};

```

5.6 Irudia: Zita modeloaren kontrolatzaileko *DeleteCita* funtzioa

Testak

Gaur egun, oso garrantzitsua da inplementatutako software baten inguruko testak egitea. Horrela, egiazta dezakegu ea sortutako softwareak hasieran ezarritako betekizunak betetzen dituen. Horrez gain, sortutako produktuak errorerik ez duela ere bermatuko da.

Test automatizatuak giza-elkarrekintza simulatzen dute, eta modu azkar eta eragin-korrean ziurtatzen dute kodean aldaketaren bat egiten den bakoitzean dena behar bezala dabilela. Horrela bermatuko dugu kodea funtzionala dela eta aurreikusitakoaren arabera funtzionatzen duela. Zure kodean akatsak atzemateko modu bat da, aplikazioarekin hedapenarekin hasi aurretik.

Test bat diseinatzerako orduan, testean espero diren emaitzak idatziko dira. Horrela testa exekutatzean, emaitza berdinak jasoz gero testa arrakastatsua izan dela jakingo dugu.

Proiektu honetarako, hiru motatako testak erabiliko ditugu:

- **Test unitarioak:** kode zati zehatzek (batez ere funtzioek) ondo funtzionatzen dutela egiaztatzeke erabiltzen dira proba unitarioak. Adibidez, datu-basean eskaera bat eginez gero, eskaerak emaitza zuzenak itzultzen dituela egiaztatzeke erabiliko dira.
- **Integrazio-testak:** aplikazioaren unitate edo ezaugarri ezberdinak konbinatuta probatzeko testak dira.
- **E2E testak:** ezaugarri-sorta guztiak erabiltzailearen ikuspegitik funtzionatzen duela egiaztatzeke erabiltzen dira. Nabigatzaile bat erabiltzen da test hauek egiteko. Kaxa beltzeko testak dira, hau da, aplikazioaren barne-egitura ez dute kontuan izango. Aplikazioaren erabiltzaile-interfazean zenbait urrats emango dira, erabiltzaile batek nabigatzaile batean eskuz egingo lukeena simulatuz.

Azterketa bat egin ostean bi tresna erabiltzea erabaki da testak egiteko: Jest eta Puppeteer.

Jest [19] Facebook-en test framework bat da, kode irekikoa. Jest test unitarioak eta integrazio-testak egiteko erabili dugu. E2E test-ak egiteko, Puppeteer[20]-ekin konbinatu dugu. Jest-ek ez du konfiguraziorik behar erabiltzeko, gainera erraza eta azkarra da.

Puppeteer [20] Node.js liburutegi bat da, Googlek sortu duena. Chrome nabigatzaile bat *headless* (interfaze grafikoa erabili gabe) kontrolatzeko erabiltzen da. Chrome erabiltzean, edozein web-aplikazioren gainean E2E testak egiteko erabil daiteke, erabilitako zerbitzariaren aldeko teknologia edozein dela ere. Modu honetan, Puppeteer bidez, erabiltzaile batek egin ditzakeen erabiltzaile-interfazearen gaineko ekintza batzuk simulatzen dira, eta Jest-ek egiaztatzen du ekintza horien bidez lortutako emaitza espero zena dela.

Selenium [21] erabiltzea ere pentsatu da, hau, E2E motako testak egiteko beste liburutegi ezagun bat da. Baina Puppeteer-ek zenbait abantaila ditu Selenium-en aurrean. Besteak beste, instalazio-prozesua errazagoa da Puppeteer-en, azkarragoak dira probak eta Chromek eskaintzen dituen orrialdeko kargatze- eta errenderizazio-denborak neurtzeko aukera ematen du.

6.1 Test unitarioak eta integrazio-testak

Test unitarioak, lehenago aipatu bezala, kode-zati zehatzak testatzeko erabiltzen da. Integrazio-testak ordea, unitate edo ezaugarri ezberdinak konbinatuta testatzeko erabiltzen dira. Proiektuan test mota hauek erabili dira datu-baseari egindako dei guztien funtzionamendua egokia dela aztertzeko. Egindako testak kontrolatzaileka banatu ditugu, eta kontrolatzaile bakoitzean dauden funtzio guztiak aztertu ditugu. Garatutako proben helburua %75eko estaldura lortzea izan da.

Kontrolatzaile bakoitzeko pare bat test azalduko dira, horien exekuzioa nolakoa izan den erakusteko.

Egoitza

Egoitzaren kontrolatzailearen probak egiteko, 9 test egin dira 6.1 irudian ikusi daitekeen bezala. Test horiek datu-baseko deien funtzionamendua bermatzen dute, eta horien bidez, 6.7 irudian ikus daitekeen bezala, egoitzaren kontrolatzailearen %86ko estaldura lortu da.

Egoitzaren kontrolatzailearen inguruko “Egoitza bat sortu” testa azalduko da orain.

```
PASS tests/routes.test.js
Egoitzaren testak
  ✓ Egoitza bat sortu (101 ms)
  ✓ Errore bat ematen du informazioa bidaltzen ez bada (7 ms)
  ✓ Egoitza guztiak lortzen ditu (17 ms)
  ✓ Egoitza zehatz bat eguneratzen ditu (29 ms)
  ✓ Errorea ematen du existitzen ez den egoitza bat eguneratzeak (12 ms)
  ✓ Egoitza zehatz bat lortzen ditu (12 ms)
  ✓ Errorea ematen du existitzen ez den ID baten egoitza lortzeak (10 ms)
  ✓ Errorea ematen du existitzen ez den ID baten egoitza ezabatzeak (10 ms)
  ✓ Egoitza ezabatzen du (28 ms)
```

6.1 Irudia: Egoitzaren testak

Langilea

Langilearen kontrolatzailearen probak egiteko, 6.2 irudian ikus daitezkeen 15 probak egin dira. Test horiek eginez, 6.7 irudian ikus daitekeen bezala, langilearen kontrolatzailearen

%88ko estaldura lortu da.

```

PASS tests/routes.test.js
Langilearen testak
  ✓ Langilea sortu (433 ms)
  ✓ Langilea sortu : errorea itzultzen du ez bada informaziorik bidaltzen (8 ms)
  ✓ Ezin da sortu langile bat DBan existitzen den eposta batekin (331 ms)
  ✓ Langilea eguneratu (49 ms)
  ✓ Errorea ematen du existitzen ez den langile bat eguneratzen saiatzeak (24 ms)
  ✓ Langile guztiak lortzen ditu (33 ms)
  ✓ Langile zehatz bat lortzen du eposta erabiliz (13 ms)
  ✓ Errorea ematen du existitzen ez den langile bat lortzen saiatzeak (10 ms)
  ✓ Langilearen pasahitza eguneratzen da (638 ms)
  ✓ Errorea ematen du langilearen pasahitza eguneratzeak, langilea existitzen ez bada (16 ms)
  ✓ Errorea ematen du langilearen pasahitza eguneratzeak, egungo pasahitza okerra bada (357 ms)
  ✓ Ezin da ezabatu existitzen ez den langile bat (9 ms)
  ✓ Errorea ematen du eposta ezezagun batekin saioa hasteak (11 ms)
  ✓ Errorea ematen du okerreko pasahitzarekin saioa hasteak (317 ms)
  ✓ Saioa ondo hasten da (312 ms)

```

6.2 Irudia: Langilearen testak

Erabiltzailea

Erabiltzailearen kontrolatzailearen probak egiteko, 25 test egin dira, test horiek 6.3 irudian ikus daitezke. Test horiek eginez, 6.7 irudian ikus daitekeen bezala, langilearen kontrolatzailearen %82ko estaldura lortu da.

```

PASS tests/routes.test.js
Erabiltzailearen testak
  ✓ Erabiltzailea sortu (138 ms)
  ✓ 2. erabiltzaile bat sortu (71 ms)
  ✓ Erabiltzailea sortu : Errorea ematen du ez bada informaziorik bidaltzen (12 ms)
  ✓ Erabiltzailea sortu: Errorea ematen du existitzen den dokumentazio zenbaki bat sartzen saiatuz gero (19 ms)
  ✓ Saioa hasi langile batek (482 ms)
  ✓ Dokumentazio zenbaki zehatz bat duen erabiltzailea lortu (17 ms)
  ✓ Errorea ematen du existitzen ez den erabiltzaile bat lortzeak (10 ms)
  ✓ Langile baten erabiltzaile guztiak lortu (18 ms)
  ✓ Langilearen token-a okerra bada, errorea ematen du bere erabiltzaile guztiak lortzen saiatzeak (6 ms)
  ✓ Erabiltzailea eguneratu (80 ms)
  ✓ Erabiltzailea eguneratu: errorea ematen du ez bada existitzen (24 ms)
  ✓ Egiaztatu ea telefono zenbakia existitzen den (22 ms)
  ✓ Telefono zenbakia existitzen ez bada errorea itzuliko du (19 ms)
  ✓ Egiaztatu ea erabiltzailea existitzen den (18 ms)
  ✓ Erabiltzailea existitzen ez bada, errorea itzuliko du (15 ms)
  ✓ Lortu langile batek sartu dituen nazionalitate guztiak (16 ms)
  ✓ Langilearen token-a okerra bada, ezingo dira sartu dituen nazionalitateak lortu (17 ms)
  ✓ Lortu datu-basean dauden nazionalitate guztiak (13 ms)
  ✓ Lortu langile batek sartu dituen sede guztiak (13 ms)
  ✓ Langilearen token-a okerra bada, ezingo dira sartu dituen sedeak lortu (6 ms)
  ✓ Lortu nazionalitate bakoitzeko erabiltzaile kopurua (14 ms)
  ✓ Lortu behar (atzeritartasuneko kasuak) bakoitzeko erabiltzaile kopurua (12 ms)
  ✓ Gehitu langile bat erabiltzaile bati (56 ms)
  ✓ Ezabatu erabiltzaile bat dokumentazio zenbakia erabiliz (40 ms)
  ✓ Ezin da existitzen ez den erabiltzaile bat ezabatu (12 ms)

```

6.3 Irudia: Erabiltzailearen testak

Kasua

Kasuaaren kontrolatzailearen probak egiteko, 28 test egin dira, test horiek 6.4 irudian ikus daitezke. Test horiek eginez, 6.7 irudian ikus daitekeen bezala, langilearen kontrolatzailearen %77ko estaldura lortu da.

6. TESTAK

```
PASS tests/routes.test.js
Kasuen testak
  ✓ Langile baten saioa hasi (567 ms)
  ✓ Diskriminazio kasua sortu (81 ms)
  ✓ Ezin da kasua sortu erabiltzailea edo langilea existitzen ez bada (70 ms)
  ✓ Diskriminazio kasua eguneratu (41 ms)
  ✓ Diskriminazio kasua amaitu (29 ms)
  ✓ Atzerritartasun kasua sortu (31 ms)
  ✓ Atzerritartasun kasua eguneratu (22 ms)
  ✓ Atzerritartasun kasua amaitu (161 ms)
  ✓ Etxeko langile kasua sortu (75 ms)
  ✓ Etxeko langile kasua eguneratu (39 ms)
  ✓ Etxeko langile kasua amaitu (26 ms)
  ✓ Langile baten kasu guztiak lortu (23 ms)
  ✓ Ezin dira langile baten kasu guztiak lortu okerreko tokenarekin (10 ms)
  ✓ Amaitu gabeko kasu guztiak lortu (12 ms)
  ✓ Ezin dira langile baten amaitu gabeko kasu guztiak lortu okerreko tokenarekin (7 ms)
  ✓ Kasu guztiak lortu (19 ms)
  ✓ Kasu bat lortu IDa erabiliz (17 ms)
  ✓ Kasu espezifikoa lortu IDa erabiliz: Etxeko langilea (27 ms)
  ✓ Kasu espezifikoa lortu IDa erabiliz: Diskriminazioa (14 ms)
  ✓ Kasu espezifikoa lortu IDa erabiliz: Atzerritartasuna (12 ms)
  ✓ Daten arabera, behar guztiak lortu (14 ms)
  ✓ Kasu baten beharrak lortu IDa erabiliz (14 ms)
  ✓ Kasu baten priektuak lortu IDa erabiliz (17 ms)
  ✓ Kasu mota lortu IDa erabiliz (21 ms)
  ✓ Ezin da existitzen ez den kasu bat ezabatu (14 ms)
  ✓ Kasua ezabatu: Diskriminazioa (31 ms)
  ✓ Kasua ezabatu: Etxeko langilea (28 ms)
  ✓ Kasua ezabatu: Atzerritartasuna (26 ms)
```

6.4 Irudia: Kasuaren testak

Interbentzioa

Interbentzioaren kontrolatzailea testatzeko 7 test egin dira. Testak 6.5 irudian ikus daitezke. Test horiekin,%67ko estaldura lortu da. Kasu honetan ez da estalduraren helburua bete, hori Jest-en dokumentuak bidaltzeko aukerarik ez dagoelako da. Hau da, interbentzioek erantsitako dokumentuak izan ditzakete. Dokumentu horiek erabiltzaileak ordenagailuan hautatuko ditu, eta ondoren zerbitzarira bidaliko dira. Jest-ek ez du hau simulatzeko aukera ematen, beraz dokumentuei lotutako kode zatiak ezin izan dira probatu.

```
PASS tests/routes.test.js
Interbentzioen testak
  ✓ Interbentzioa sortu (144 ms)
  ✓ Ezin da interbentzio bat sortu, kasua existitzen ez bada (29 ms)
  ✓ Interbentzioa eguneratu (37 ms)
  ✓ Kasu baten interbentzio guztiak lortu (16 ms)
  ✓ Interbentzio zehatz bat lortu IDa erabiliz (11 ms)
  ✓ Interbentzio baten dokumentu bat deskargatu. (30 ms)
  ✓ Interbentzioa ezabatu IDaz baliatuz (31 ms)
```

6.5 Irudia: Interbentzioaren testak

Zita

Zitaren kontrolatzailearen testak egiteko 7 test implementatu dira. Test horiek, 6.6 irudian ikus daitezke. Kontrolatzaile honentzat testen bidez lortutako estaldura %89koa izan da, 6.7 irudian ikus daitekeen bezala.

```

PASS tests/routes.test.js
Zitaren testak
  ✓ Zita sortu (342 ms)
  ✓ Zita sortu : errorea ematen du informazioa bidaltzen ez badugu (35 ms)
  ✓ Erabiltzaile gabeko zita sortu (60 ms)
  ✓ Ezin da zita sortu langile gabe (80 ms)
  ✓ Zita guztiak lortu (31 ms)
  ✓ Ezabatu zita IDa erabiliz (68 ms)
  ✓ Ezin da zita ezabatu IDa existitzen ez bada (28 ms)

```

6.6 Irudia: Zitaren testak

Estaldura

6.7 irudian erakusten den bezala, atal honetan zerbitzariaren aldeak izan duen estaldura ikus daiteke. Argazkiko taulan ikusten den bezala, fitxategien estaldura totala %85ekoa izan da. Gainera, modelo guztiak %100eko estaldura dutela bermatzen da.

File	% Lines
All files	85.27
server	71.43
server.js	71.43
server/config	100
auth.config.js	100
db.config.js	100
server/controllers	80.55
caso.js	77.78
cita.js	89.29
intervencion.js	66.67
sede.js	86.84
trabajador.js	89.19
usuario.js	82.41
server/models	100
caso.js	100
cita.js	100
cita_usuario_trabajador.js	100
discriminacion.js	100
documento.js	100
externa.js	100
extranjeria.js	100
index.js	100
interna.js	100
intervencion.js	100
nacionalidad.js	100
necesidadextranjeria.js	100
proyectoextranjeria.js	100
sede.js	100
trabajador.js	100
trabajador_sede.js	100
trabajador_usuario.js	100
trabajadorahogar.js	100
usuario.js	100
server/routes	90.54

6.7 Irudia: Estaldura

6.2 E2E testak

Lehenago aipatu bezala, E2E edo ertzetik ertzerako testak, ezaugarri-sorta guztiak erabiltzailearen ikuspegitik funtzionatzen duela egiaztatzeko erabiltzen dira. Test hauek Chromium [22] nabigatzailea erabiliz exekutatu dira.

Mota honetako testetan ezarri den helburua erabilpen-kasu guztiak testatzea izango da. Ondorioz, administratzaile eta langile rolak kontuan edukiz, test batzuk bi aldiz implementatuko dira, langile ezberdinarekin.

Administratzailea

6.8 irudian administratzailearen rolean egindako ertzetik-ertzerako testak erakusten dira. Irudian ikus daitekeen bezala, egoitzak eta langileak kudea ditzake administratzaileak.

```
PASS src/tests/routes.test.js
Login
  ✓ users can login (4475 ms)
Langilea
  ✓ Langilea sortu (7206 ms)
  ✓ Langilea eguneratu (1447 ms)
  ✓ Langilea ezbatu (1014 ms)
  ✓ Langile guztiak ikusi (464 ms)
Erabiltzailea
  ✓ Erabiltzailea sortu (5779 ms)
  ✓ Erabiltzailea eguneratu (1813 ms)
  ✓ Erabiltzaileak ikusi (482 ms)
Kasua
  ✓ Kasua sortu (7582 ms)
  ✓ Kasua eguneratu (1328 ms)
  ✓ Kasua amaitu (1407 ms)
  ✓ Langilearen kasu guztiak ikusi (503 ms)
  ✓ Kasu guztiak ikusi (513 ms)
Egoitza
  ✓ Egoitza sortu (5741 ms)
  ✓ Egoitza eguneratu (888 ms)
  ✓ Egoitza ezabatu (651 ms)
  ✓ Erabiltzailea ezabatu (1019 ms)
  ✓ Egoitza guztiak ikusi (478 ms)
Interbentzioa
  ✓ Interbentzioa sortu (5596 ms)
  ✓ Interbentzioa eguneratu (1216 ms)
  ✓ Interbentzioa ezabatu (820 ms)
  ✓ Interbentzio guztiak ikusi (1014 ms)
  ✓ Kasua ezabatu (1004 ms)
```

6.8 Irudia: E2E testak - Administratzailea

Langilea

6.9 irudian langilearen rolean egindako ertzetik-ertzerako testak erakusten dira.

```
PASS src/tests/langilea.test.js
Erabiltzailea
  ✓ Erabiltzailea sortu (6608 ms)
  ✓ Erabiltzailea eguneratu (518 ms)
  ✓ Erabiltzaileak ikusi (12 ms)
Kasua
  ✓ Kasua sortu (9667 ms)
  ✓ Kasua eguneratu (686 ms)
  ✓ Kasua amaitu (1738 ms)
  ✓ Langilearen kasu guztiak ikusi (22 ms)
Interbentzioa
  ✓ Interbentzioa sortu (5956 ms)
  ✓ Interbentzioa eguneratu (567 ms)
  ✓ Interbentzioa ezabatu (2123 ms)
  ✓ Interbentzio guztiak ikusi (165 ms)
  ✓ Kasua ezabatu (302 ms)
  ✓ Erabiltzailea ezabatu (396 ms)
```

6.9 Irudia: E2E testak - Langilea

Hedapena

Atal honetan proiektuaren hedapenerako jarraitu diren pausuak azalduko dira.

Hedapena egiteko VPS (Zerbitzari Birtual Pribatu) motako zerbitzari bat erabili da. Hau zerbitzari fisiko baten barneko partizio birtuala da, partizio bakoitzari baliabide eskusiboak esleituko zaizkio. Baliabideei esker, sistema eragile bat instalatzeko eta askatasun-maila handiarekin lan egiteko aukera ematen da.

7.1 Domeinua eta SSL ziurtagiria

SSL (Secure Sockets Layer edo konexio seguruko geruza) [23] segurtasun orokorreko estandarra da, nabigatzaile baten eta web-zerbitzari baten artean zifratutako datuak transferitzeko aukera ematen duena. Gaur egun, milioika enpresek eta erabiltzaileek erabiltzen dute informazio konfidentziala lapurtzeko eta manipulatzeko arriskua murrizteko (erabiltzaile-izenak, pasahitzak, mezu elektronikoak, etab.). Konexio segurua ezartzeko, SSL ziurtagiri bat instalatzen da web-zerbitzari batean (“ziurtagiri digitala” ere esaten zaio), eta bi funtzio betetzen ditu:

- Web gunearen nortasuna egiaztatzea, bisitariei leku faltsu batean ez daduela bermatuz.
- Transmittitutako informazioa zifratzea.

SSL ziurtagiria dohainik lortzeko Let’s Encrypt [24] ziurtagiri jaulkitzailea erabiliko da. Let’s Encrypt-ek, ziurtagiri bat emateko, domeinu izen bat beharko da. Ziurtagiri jaulkitzaile honen helburua Internet leku seguruagoa bihurtzea da, eta garrantzitsua da argitzea eskaintzen dituen ziurtagiriak ordaintzekoak izan daitezkeen beste guztiak bezain baliozkoak eta onak direla.

Let’s Encrypt-ek eskaintzen dituen ziurtagiriak laurogeita hamar eguneko iraupena dute. Ziurtagiri jaurtitzzaileen gehiengoak urte bateko iraupena izan ohi dute gutxienez. Arazo honi aurre egiteko, Certbot [25] erreminta erabiliko da. Hau, zerbitzarian instalatuko da eta automatikoki berrituko du SSL ziurtagiria.

7. HEDAPENA

Erabili den domeinua, SOS Arrazakeriak jadanik zuen berdin.org domeinua da. Domeinu izen hau lehendik zuten Nominalia [26] enpresan egongo da alojatuta.

Jarraipena eta Kontrola

Atal honetan proiektuaren jarraipena eta kontrola nola egin diren azalduko da.

Alde batetik, proiektuaren irismena nola aurreratu den azaltzen da, eta, beste alde batetik, hasierako plangintzarekiko denbora-aldaketak azalduko dira. Denbora aldaketak aztertzeko, taula bat egin da. Taula honetan, hasieran estimatutako ordu kopurua, behar izan diren ordu errealak eta hauen arteko ezberdintasuna erakutsiko da.

8.1 Denboraren kudeaketa

Denboraren gestioak zenbait aldaketa izan ditu hasierako planifikaziotik. Hau zenbait arrazoi ezberdinegatik izan da. Hasteko, ez zen kontutan izan teknologia berriak erabiltzeak ikaste-prozesu bat emango zuela. Beraz, proiektuaren garapenaren hasiera esperotakoa baina motelagoa izan zen, aipatutako ikaste-prozesua ematen ari zelako. Gerora, gauzak azkartzen joan dira. Gainera, datu-basearen diseinuak espero baina denbora gehiago eman du. Zenbait erabaki hartu ostean bezeroarekin bildu, eta gauzak aldatzea adosten zen. Ondorioz, diseinua guztiz bukatzen denbora asko eman da, bilera asko egin baitira. Plangintzan estimatutako orduak eta benetan eman diren orduen arteko alderapena 8.1 taulan erakusten da. Taula honetan, bi zutabe gehitu dira: errealitatean emandako ordu kopurua, eta desbiderapena.

Taulan ikus daitekeen bezala, estimatutakoa baina 35 ordu gehiago behar izan dira proiektua garatzeko.

8.2 Komunikazioa

Proiektuan zehar izandako komunikazioaren gehiengoa eposta bidez izan da. Fakultateko tutoreekin izandako komunikazioaren gehiengoa proiektu hasieran eta proiektuaren amaieran izan da. Hasieran, proiektuaren nondik norakoak aurkeztu eta finkatzeko. Gero, datu-basearen diseinua baliozkotzeko komunikazioa egon da. Proiektuaren amaieran ordea, memoriaren zuzenketak egin eta memoriarekin amaitzeko. SOS Arrazakeriako kideekin, bilera telematikoak izan ditugu Google Meet bidez. Bilerak modu erregularrean egin ditugu, egindako aldaketak aurkeztu eta hobekuntzak proposatu ahal izateko.

8. JARRAIPENA ETA KONTROLA

Ataza	Orduen estimazioa	Ordu kopurua	Desbiderapena
Aplikazioa	250 ordu	286 ordu	+36 ordu
Hasieratzea	60 ordu	66 ordu	+6 ordu
A.1	10 ordu	8 ordu	-2 ordu
A.2	10 ordu	8 ordu	-2 ordu
A.2	10 ordu	10 ordu	-
D.1	15 ordu	15 ordu	-
D.2	15 ordu	25 ordu	+10 ordu
Inplementazioa	190 ordu	220 ordu	+30 ordu
G.1	10 ordu	20 ordu	+10 ordu
G.2	140 ordu	140 ordu	-
P.1	10 ordu	20 ordu	+10 ordu
P.2	5 ordu	10 ordu	+5 ordu
H.1	20 ordu	20 ordu	-
H.2	15 ordu	20 ordu	+5 ordu
H.3	5 ordu	2 ordu	-3 ordu
Memoria	70 ordu	72 ordu	+2 ordu
M.1	60 ordu	60 ordu	-
M.2	2 ordu	2 ordu	-
M.3	8 ordu	10 ordu	+2 ordu
Kudeaketa	30 ordu	27 ordu	-3 ordu
PL.1	10 ordu	7 ordu	-3 ordu
PL.2	6 ordu	6 ordu	-
PL.3	2 ordu	2 ordu	-
B.1	10 ordu	10 ordu	-
B.2	-	-	-
JK.1	2 ordu	2 ordu	-
JK.2	-	-	-
JK.3	-	-	-
Guztira	350 ordu	385 ordu	+35 ordu

8.1 Taula: Ordu errealak

Ondorioak

Azken atal honetan, proiektuaren garapenean zehar jasotako zenbait ondorio azalduko dira. Gainera, proiektuak etorkizunean izango dituen hobekuntza eta proposamenak azalduko dira. Izaera honetako software proiektu batek mantentze eta eguneraketa konstante bat beharko baitu.

9.1 Proiektuaren ondorioak

Lehenetik eta behin, aplikazioaren implementazioaz hitz egingo da. Nahiz eta erabilitako teknologiatan alde aurreko ikasketa-prozesu bat eman, ezagutza berriak eskuratzen joan naiz proiektuaren garapenean zehar. Proiektuaren tamainak, aurretiazko analisi baten eta garapenaren planifikazioaren garrantzia ulertzeko balio izan du. Planifikazio egoki bat egiteak proiektua egunean eramaten lagundu dit. Gainera, aplikazioaren diseinua garapenaren aurretik egitearen garrantzia ulertu da, diseinatu gabeko aplikazio baten garapenak denbora gehiago behar du eta erroreak izateko aukerak handitzen ditu.

Erabilitako metodologia arinak, iterazio bakoitzaren irizpideak hasieran ezartzeak eta Sprintaren amaieran emaitza lortzeak motibazioa eragin du. Produktua garatzen jarraitzea bultzatu du, emaitzak bistakoak baitira. Sprintaren garapenentarako epe laburrak ezarri ahal izateak etengabeko aurrerapena errazten du, eta aplikazioaren implementazioan zailtasunak atzematen laguntzen du.

Hasieran bezeroak ezarritako aplikazioaren helburuak bete direnez eta proiektuaren hedapen zuzena egin denez, proiektua arrakastaz garatu dela ondoriozta daiteke.

9.2 Ondorio pertsonalak

Proiektuaren eskakizunek, zenbait teknologia berri eta ezezagun erabiltzera bultzatu naute. Erabilitako teknologia asko ez dira unibertsitatean irakatsi, hauek ikasi eta sakontzea erronka bat izan da niretzat. Bestalde, unibertsitatetik ezagutzen nituen teknologia asko ondo barneratu eta hauetan sakontzeko aukera izan dut. Hau oso positiboa izan da niretzat, lan mundurako prestaketa bat izan baita.

Gainera, proiektu hau burutzeak bezeroekin harremanak izaten irakatsi dit. Momentu oro, egindako aurrerapenak eta hartutako erabakiak SOS Arrazakeriako kideekin komentatu baitut.

Bukatzeko, izaera honetako proiektu baten diseinu, inplementazio eta hedapenak esperientzia asko eman dit.

9.3 Etorkizunerako hobekuntza eta proposamenak

Proiektua bukatutzat ematen da, hasieran ezarritako funtzionalitateak bete baitira eta Gradu Amaierako Lana bukatu baita. Proiektuak martxan jarraituko du ordea, SOS Arrazakeriako kideek aplikazioa erabiliko baitute. Ondorioz, aplikazioak hazi egingo da, bai funtzionalitate aldetik baita gestionatu beharreko erabiltzaile aldetik.

Epe motzera egiteko hobekuntzak hurrengoak izango dira:

- Etxeko langileen kasuak lantzerako orduan, zenbait kalkulu egin behar dira, adibidez, langileak kobratu beharko lukeen finikitoaren kalkulua. Hau, aplikazioak bakarrik egitea aurreikusten da, gaur egun eskuz kalkulatu behar baitituzte zifra hauek.
- Bizilagun eta boluntarioak datu-basean gehitzea.
- Egutegian egoitza bakoitzeko egutegi bat egoteko aukera izatea. Egutegiak sortu eta ezabatzeko aukera.

Eranskina

Bileren aktak

Lehenengo bilera

Data : 2020/11/11 16:00

Bertaratutako kideak:

- Joane Mannion
- Ana Arruarte
- Olatz Arbelaitz
- Gerardo Carrere
- Nora Ugarteburu

Lekua: Bb Collaborate bidezko bilera telematikoa.

Eguneko helburuak:

- Bezeroa, proiektuaren tutoreak eta Gradu Amaierako Lana garatuko duen ikaslea ezagutzea.
- Proiektuaren lehen hurbilpen bat eman.

Hartutako erabakiak: -

Bigarren bilera

Data : 2020/12/14 12:00

Bertaratutako kideak:

- Joane Mannion
- Gerardo Carrere
- Nora Ugarteburu

Lekua: Google Meet bidezko bilera telematikoa.

Eguneko helburuak:

- Programaren funtzionamendua adostea.
- Datu-basean gordeko den informazioa zehaztea.

Hartutako erabakiak:

- Aplikazioaren diseinu bat egingo da MarvelApp erabiliz garapenarekin hasi aurretik, honela bezeroarekin adostuko da aplikazioaren funtzionamendua eta gero garapena errazagoa izango da.

Hirugarren bilera

Data : 2021/01/22 10:00

Bertaratutako kideak:

- Joane Mannion
- Gerardo Carrere
- Nora Ugarteburu

Lekua: Google Meet bidezko bilera telematikoa.

Eguneko helburuak:

- Aplikazioaren diseinua komentatu. Diseinu hau <https://marvelapp.com/prototype/13dg3h09> linkean ikus daiteke.

Hartutako erabakiak:

- Etxeko langileen datuak gehitu erabiltzailea gehitzeko formulariora.
- Diskriminazioko datuak gehitu erabiltzailea gehitzeko formulariora.
- Diseinuarekin ados dago bezeroa, beraz, datu-basearen diseinuarekin hasi.

Laugarren bilera

Data : 2021/03/09 16:30

Bertaratutako kideak:

- Joane Mannion
- Gerardo Carrere
- Nora Ugarteburu

Lekua: Google Meet bidezko bilera telematikoa.

Eguneko helburuak:

- Datu-basearen diseinua komentatu.
- Aplikazioaren inguruko duda zehatzak adostu

Hartutako erabakiak:

- Etxeko langileen datuak gehitu erabiltzailea gehitzeko formulariora.
- Diskriminazioko datuak gehitu erabiltzailea gehitzeko formulariora.
- Diseinuarekin ados dago bezeroa, beraz, datu-basearen diseinuarekin hasi.

Barne-langilearen fitxa

FECHA ____/____/____ CUANTAS CASAS? ____ INTERNA

NOMBRE Y APELLIDOS _____

DOMICILIO (pueblo) _____ TLF: _____ ¿WHATSAPP? ____

NACIONALIDAD/ES _____ REGULARIZADA: SI / NO ____

FECHA COMIENZO TRABAJO _____ ÚLTIMO DÍA TRABAJO _____

DÍA COMUNICACIÓN DESPIDO _____

FORMA CONSEGUIR EMPLEO: _____

¿QUIÉN TE CONTRATÓ? PERSONA ATENDIDA PARIENTE. HOMBRE PARIENTE MUJER
(Cuando se contrata para cuidar menores, contrata la persona atendida)

Contratada solo para el fin de semana SI / NO

HORARIO (concretar días y horas)

Permiso diario para salir, nº de horas y días _____

Descanso semanal (poner principio y final) _____

TOTAL HORAS SEMANA _____

LIBRA FESTIVOS: SI / NO / F (si está contratada para festivos es F)

No los libra pero los cobra (marcar X aquí)

SALARIO (lo que se lleva a casa, incluyendo fines) _____

PAGAS (número y cuantía de cada paga) _____

TE PAGAN: EN MANO BANCO NÓMINA (distinta del justificante bancario): SI / NO

CONTRATO: SI / NO _____ SEGURIDAD SOCIAL: SI / NO _____

VACACIONES: SI (si descansa un mes) NO y las cobra aparte NO y no las cobra aparte

TIPO DE TRABAJO: Cuidado de adultos como actividad principal Tareas domésticas sin cuidado

Cuidado de adultos NO actividad principal Cuidado y número de criaturas
MAYORES Cuántos _____ M / H

ENFERMAS Cuántos _____ M / H ¿VIVEN SOLOS? SI / NO (solo adultos)

MOTIVO CONSULTA: INFORMACIÓN DESPIDO/DESISTIMIENTO

DESP. MATERNIDAD MUERTE INGRESO RESIDENCIA OTRAS

Kanpo-langilearen fitxa

FECHA ____/____/____ CUANTAS CASAS? ____ EXTERNA

NOMBRE Y APELLIDOS _____

DOMICILIO (pueblo) _____ TLF. +34 _____ ¿WHATSAPP? ____

NACIONALIDAD/ES _____ REGULARIZADA: SI / NO ____

FECHA COMIENZO TRABAJO _____ ÚLTIMO DÍA TRABAJO _____

DÍA COMUNICACIÓN DESPIDO _____

FORMA CONSEGUIR EMPLEO: _____

¿QUIÉN CONTRATA? PERSONA ATENDIDA PARIENTE HOMBRE PARIENTE MUJER
(Cuando se contrata para cuidar menores, contrata la persona atendida)

HORARIO (concretar días y horas)

TOTAL HORAS SEMANA _____

LIBRA FESTIVOS: SI / NO / F (si está contratada para festivos es F)

NO LOS LIBRA PERO LOS COBRA (marcar X aquí) Cuanto cobra _____

SALARIO (lo que se lleva a casa) _____

PAGAS (número y cuantía de cada paga) _____

TE PAGAN: EN MANO BANCO NÓMINA (distinta del justificante bancario): SI / NO

CONTRATO: SI / NO _____ SEGURIDAD SOCIAL: SI / NO _____

VACACIONES: SI (si descansa un mes) NO y las cobra aparte NO y no las cobra aparte

TIPO DE TRABAJO: Cuidado de adultos como actividad principal Tareas domésticas sin cuidado
 Cuidado de adultos NO actividad principal Cuidado y número de criaturas

MAYORES Cuántos _____ M / H

ENFERMAS Cuántos _____ M / H ¿VIVEN SOLOS? SI / NO (solo adultos)

MOTIVO CONSULTA: INFORMACIÓN DESPIDO/DESISTIMIENTO

DESP. MATERNIDAD MUERTE INGRESO RESIDENCIA

OTRAS CAUSAS _____

2 Irudia: Kanpo-langilearen fitxa

Atzerritartasun formularioa

Sección 1 de 8

Recogida de Datos. Proyecto OID

Primer caso

Este formulario registra automáticamente los correos de los usuarios de SOS Racismo Gipuzkoa. [Cambiar configuración](#)

Territorio *

A Coruña

Bizkaia

Catalunya

Gipuzkoa

Nafarroa

Zaragoza

Otra...

¿Quién recoge la información? *

Texto de respuesta corta

Nombre de la víctima *

Texto de respuesta corta

Apellidos de la víctima *

Texto de respuesta corta

Sexo de la víctima *

1. Hombre

2. Mujer

3. Otros

4. Prefiero no decirlo

Edad *

Texto de respuesta corta

Pais de origen *

Texto de respuesta corta

Nacionalidad *

Texto de respuesta corta

Teléfono *

Texto de respuesta corta

Correo electrónico

Texto de respuesta corta

67

DNI/NIE/Pasaporte *

Texto de respuesta corta

Situación Documental *

1. Nacionalidad española
2. NIE
3. Irregular
4. Nacional UE
5. Demandante de asilo

Situación residencial *

1. Con domicilio
2. Sin domicilio
3. En acogida
4. Otros recursos residenciales

Nivel de estudios *

1. Sin estudios
2. Primaria
3. Secundaria
4. Bachiller
5. Formación Profesional
6. Universitarios
7. Tercer Grado

Municipio de residencia *

Texto de respuesta corta

Color de piel / Etnia / Rasgos fenotipicos *

Texto de respuesta larga

Tipo *

1. CONFLICTOS Y AGRESIONES RACISTAS
2. DENEGACIÓN DE ACCESO A PRESTACIONES Y SERVICIOS PRIVADOS
3. DENEGACIÓN DE ACCESO A PRESTACIONES Y SERVICIOS PÚBLICOS
4. DISCRIMINACIÓN LABORAL
5. EXTREMA DERECHA Y DISCURSO DEL ODIO
6. RACISMO INSTITUCIONAL
7. SEGURIDAD PRIVADA
8. SEGURIDAD PÚBLICA

Después de la sección 1 Ir a la siguiente sección

Sección 2 de 8

Conflictos y Agresiones

Agresión física e insultos racistas

CONFLICTOS Y AGRESIONES

- EN EL ESPACIO PÚBLICO
- CONFLICTOS VECINALES
- OTROS: INDICAR

Después de la sección 2 Ir a la sección 6 (Agente discriminador)

Sección 3 de 8

DENEGACIÓN DE ACCESO A PRESTACIONES Y SERVICIOS PRIVADOS

Descripción (opcional)

DENEGACIÓN DE ACCESO A PRESTACIONES Y SERVICIOS PRIVADOS *

- BANCOS
- LOCALES DE OCIO Y RESTAURANTES
- VIVIENDA
- Otra...

Después de la sección 3 Ir a la sección 7 (Derivamos)

Sección 4 de 8

DENEGACIÓN DE ACCESO A PRESTACIONES Y SERVICIOS PÚBLICOS

Descripción (opcional)

DENEGACIÓN DE ACCESO A PRESTACIONES Y SERVICIOS PÚBLICOS *

- EMPLEO
- ASISTENCIA SANITARIA
- OTRAS ADMINISTRACIONES
- EMPADRONAMIENTO
- SERVICIOS SOCIALES

Después de la sección 4 Ir a la sección 5 (RACISMO INSTITUCIONAL)

Sección 5 de 8

RACISMO INSTITUCIONAL

Descripción (opcional)

RACISMO INSTITUCIONAL *

- CIES
- NACIONALIDAD
- OTROS INDICAR
- TRAMITES EXTRANJERIA

Después de la sección 5 Ir a la sección 6 (Agente discriminador)

Sección 6 de 8

Agente discriminador

Descripción (opcional)

Agente discriminador *

Entidad pública

Entidad privada


Particular

Otros

Relato de los Hechos *

Texto de respuesta larga

Fecha de los hechos *

Mes, día, año 

Municipio *

Texto de respuesta corta

Identificación del agente discriminador

Texto de respuesta corta

Personas que pueden testificar

Texto de respuesta corta

Otros elementos probatorios

Texto de respuesta larga

Estrategia a seguir *

1. Derivamos
2. Asumimos

Después de la sección 6 Ir a la siguiente sección

Sección 7 de 8

Derivamos

Descripción (opcional)

Entidad u organización

Texto de respuesta corta

Después de la sección 7 Enviar formulario

Sección 8 de 8

Asumimos caso

Descripción (opcional)

Tipo de acción

1. Vía Penal
2. Defensor del pueblo
3. Denuncia pública
4. No desea interponer denuncia
5. Institución causante
6. Otras

Bibliografia

- [1] SOS Racismo quienes somos. <https://sosracismo.eu/quienes-somos/>, 2016. Accessed: 2021-03-17. Ikusi 1 orrialdea.
- [2] GitHub. <https://github.com/>, 2021. Accessed: 2021-05-21. Ikusi 7, 13 orrialdeak.
- [3] Overleaf: textu editorea. <https://www.overleaf.com/>, 2021. Accessed: 2021-05-21. Ikusi 7 orrialdea.
- [4] React: Js-erako liburutegia. <https://es.reactjs.org/>, 2021. Accessed: 2021-05-21. Ikusi 11 orrialdea.
- [5] Angular: Js-erako liburutegia. <https://angular.io/>, 2021. Accessed: 2021-05-21. Ikusi 12 orrialdea.
- [6] Node.js: Js-erako exekuzio ingurunea. <https://expressjs.com/es/>, 2021. Accessed: 2021-05-21. Ikusi 12 orrialdea.
- [7] Express.js: Node.js web-aplikazioen azpiegitura. <https://expressjs.com/es/>, 2021. Accessed: 2021-05-21. Ikusi 12 orrialdea.
- [8] MySQL: datu-base erlazionalak kudeatzeko sistema. <https://www.mysql.com/>, 2021. Accessed: 2021-05-21. Ikusi 12 orrialdea.
- [9] Visual Studio Code: kode editorea. <https://code.visualstudio.com/>, 2021. Accessed: 2021-05-21. Ikusi 13 orrialdea.
- [10] Atom: kode editorea. <https://atom.io/>, 2021. Accessed: 2021-05-22. Ikusi 13 orrialdea.
- [11] MarvelApp: prototipoak egiteko tresna. <https://marvelapp.com/>, 2021. Accessed: 2021-05-21. Ikusi 13 orrialdea.
- [12] Sonar Cloud: kodearen kalitatea aztertzeo aplikazioa. <https://sonarcloud.io/>, 2021. Accessed: 2021-05-21. Ikusi 13 orrialdea.
- [13] Scrum metodologia. <https://www.scrum.org/resources/what-is-scrum>, 2016. Accessed: 2021-05-25. Ikusi 13 orrialdea.
- [14] MVC: modelo-bista-kontrolatzaile arkitektura. <https://desarrolloweb.com/articulos/que-es-mvc.html>. Accessed: 2021-05-30. Ikusi 15 orrialdea.
- [15] Erabiltzailearen historia. https://es.wikipedia.org/wiki/Historias_de_usuario, 2016. Accessed: 2021-05-25. Ikusi 16 orrialdea.
- [16] i18next: interfaze eleaniztunak. <https://www.i18next.com>. Accessed: 2021-05-30. Ikusi 43 orrialdea.
- [17] Axios: Js liburutegia. <https://axios-http.com/>. Accessed: 2021-05-16. Ikusi 43 orrialdea.
- [18] Sequelize: Node.js-rako orm. <https://sequelize.org/>. Accessed: 2021-05-16. Ikusi 44 orrialdea.
- [19] Jest: Js-erako test liburutegia. <https://jestjs.io/>, 2021. Accessed: 2021-05-16. Ikusi 47 orrialdea.

BIBLIOGRAFIA

- [20] Puppeteer: Nabigatzailea *headless* kontrolatzeko aukera ematen duen proba ingurua. <https://www.selenium.dev/>. Accessed: 2021-05-16. Ikusi 47, 48 orrialdeak.
- [21] Puppeteer: Node.js liburutegia, chrome-eko nabigatzailea *headless* kontrolatzeko aukera ematen duena. <https://pptr.dev/>. Accessed: 2021-05-16. Ikusi 48 orrialdea.
- [22] Chromium: nabigatzailea. <https://www.chromium.org>. Accessed: 2021-05-16. Ikusi 52 orrialdea.
- [23] SSL: konexio segururako geruza. https://www.verisign.com/es_LA/website-presence/online/ssl-certificates/index.xhtml. Accessed: 2021-05-30. Ikusi 55 orrialdea.
- [24] Let's Encrypt: Ssl ziurtagiria. <https://letsencrypt.org/es/>. Accessed: 2021-05-30. Ikusi 55 orrialdea.
- [25] CertBot: Ssl ziurtagirien automatizazioa. <https://certbot.eff.org/>. Accessed: 2021-05-30. Ikusi 55 orrialdea.
- [26] Nominalia: domeinuen erregistroa. <https://www.nominalia.com/>. Accessed: 2021-05-30. Ikusi 56 orrialdea.