

Telekomunikazio Teknologiaren Ingeniaritzako Gradua

## GRADU AMAIERAKO LANA

***ZUNAHIZ PLATAFORMA***

**Ikaslea:** CASTRILLO, GONZALEZ, MIREIA

**Zuzendaria:** HUARTE, ARRAYAGO, MAIDER

# AURKIBIDEA

Laburpena.....	6
Resumen .....	7
Abstract.....	8
Taulen zerrenda .....	9
Irudien zerrenda.....	10
Laburduren zerrenda.....	12
1. Sarrera .....	13
2. Testuingurua.....	14
3. Teknologia eremua.....	16
3.1 Orrialde euskarriak .....	16
3.1.1 Web orrialdea .....	16
3.1.2 Mahaigaineko aplikazioa.....	16
3.1.3 Web aplikazioa.....	17
3.2 Garapen modua .....	18
3.2.1 Scriptcase .....	18
3.2.2 Hasieratik kodetu.....	19
4. Helburuak.....	21
4.1 Helburu nagusia .....	21
4.2 Helburu partzialak.....	21
4.2.1 Plataforma diseinua definitu.....	21
4.2.2 Plataforma egitura definitu.....	21
4.2.3 Emaitzen azterketa burutu .....	22
5. Onurak.....	23
5.1 Onura sozialak .....	23
5.2 Onura teknikoak.....	23
5.3 Onura ekonomikoak.....	24
6. Alternatiben analisisa.....	24
6.1 Orrialde euskarriak .....	24
6.1.1 Web orrialdea .....	24
6.1.2 Mahaigaineko aplikazioa.....	25
6.1.3 Web aplikazioa.....	25
Hautatze irizpideak .....	25

Erabiltzailearekin elkarrekintza.....	25
Erraztasuna .....	25
Baliabide erabilera.....	26
Erosotasuna .....	26
6.2 Garapen modua .....	26
6.2.1 Scriptcase erabili .....	27
6.2.2 Kodea hutsetik hasi .....	27
Hautatze irizpideak .....	27
Lana arintzea .....	27
Erraztasuna .....	27
Lizentziak ekidin .....	27
Diseinura egokitu .....	28
6.3 Datu basea.....	28
6.3.1 MySQL.....	29
6.3.2 MongoDB.....	29
Hautatze irizpideak .....	30
Eskalagarritasuna .....	30
Abiadura.....	30
Inplementazio erraztasuna.....	30
Aurretiazko ezaguera.....	30
7 Metodologia.....	31
7.1 Teknologiak aztertu .....	31
7.1.1 JavaScript lengoaia.....	31
7.1.2 Node JS .....	32
7.1.3 Express JS .....	34
7.1.4 Mongo DB.....	34
7.1.5 Angular.....	35
7.2 Ingurune prestaketa.....	36
7.2.1 Atom testu editorea .....	36
7.2.2 Cygwin .....	37
7.2.3 Node JS .....	38
7.2.4 Nodemon .....	38
7.2.5 Express JS .....	39
7.2.6 Mongo DB.....	39
7.2.7 Angular.....	41

7.3 Programa garapena .....	42
7.3.1 Aplikazioaren arkitektura .....	42
7.3.2 API Kodeketa .....	42
7.3.3 CLIENT Kodeketa .....	44
7.4 Probak .....	45
7.5 Emaidzak .....	45
8 Lanaren antolaketa.....	46
8.1 Lan-taldea .....	46
8.2 Lan paketeen deskribapena .....	46
8.2.1 LP 1 Proiektuaren kudeaketa eta gainbegiraketa .....	46
8.2.2 LP 2 Proiektuaren diseinua .....	47
8.2.3 LP 3 Artearen egoeraren analisisa eta ikastaroak .....	48
8.2.4 LP 4 Webgunearen garapena.....	48
8.2.5 LP 5 Emaidza aztertu .....	49
8.2.6 LP 6 Emaidza helarazi .....	50
8.2.7 LP 7 Dokumentazioa.....	50
8.3 Mugarriak.....	51
9. Gastu aitortpena .....	53
9.1 Kostuak .....	53
9.1.1 Barne orduak .....	53
9.1.2 Amortizazioak .....	53
9.1.3 Azpikontratazioak .....	54
9.1.4 Gastuak.....	54
9.2 Kostu aitortpena orotara.....	54
10 Arrisku analisia.....	55
10.1 Proiektuko kideren baten baja (A1) .....	55
10.2 Programazio lengoaiarekin arazoak (A2) .....	55
10.3 Aurrekontuarekin arazoak (A3) .....	56
10.4 Datu galera (A4) .....	56
10.5 Ezaugarri okerrak (A5) .....	56
10.6 Erabiltzaileen datuen galera (A6) .....	56
10.7 Arriskuen analisisien laburpena.....	57
11 Ondorioak .....	58
12. Erreferentziak .....	59
I. Eranskina: Egitura .....	62

API egitura .....	62
Client egitura.....	62
II. Eranskina: Probak.....	65
Proba motak .....	65
Unitate probak .....	66
Integrazio probak.....	66
End-to-end probak.....	66
Proben garapena.....	66
Unitate probak .....	67
Integrazio probak.....	73
End-to-end probak.....	75
III. Eranskina: Emaitzak .....	76
Ikasle ikuspegia.....	76
Hasiera orrialdea .....	77
Main page.....	79
Profile .....	80
Duties .....	81
Foroa.....	81
Chat .....	82
Irakasle ikuspegia .....	83
Main page.....	83
Profile.....	84
Duties .....	84
Students .....	85
Foroa.....	85
Chat .....	85
IV. Eranskina: Aplikazioa.....	86
Plataforma .....	86
Abiaraztea .....	86

# LABURPENA

Gaur egun, teknologiaren erabilera eguneroko bizitzan derrigorrezkoa den mundu honetan, gero eta gehiago erabiltzen diren baliabideak dira irakaskuntza plataformak.

Zer esanik ez, eguneroko bizitza pribatua lagun edo ezezagunekin partekatzea ahalbidetzen duten sare sozialei buruz. Lehen munduko gizartean bizi diren pertsonen, eta batez ere gazteen denboraren portzentai handia hartzen dutelarik.

Baina mundu horietan murgildurik daudenek ba al dakite baliabide horiek modu egokian eta erantzukizunez erabiltzen?

Egia da horien erabilera ez dela inon irakasten. Erabiltzen joan heinean modu intuitiboan ikasten da hauen funtzionamendua, eta askotan, horrela izateak onura baino kalte gehiago sor ditzake.

Nerabeei sare sozialetara igotzen diren edukiek eta idazten diren iruzkinek duten garrantziaz ohartarazteko asmoz sortzen da proiektu hau. Irakaskuntza inguruan murgilduta egonik, eskolako betebeharrak egin eta irakaslearekin partekatzen diren aldi berean, norbanako baten ekintzen eta hitzek duten eragina irakatsi da Zunahiz izeneko plataforma bat sortuz.

Horretarako, azterketa bat egin ondoren, gaur egunean web aplikazioak sortzeko erabiltzen diren teknologia batzuk aukeratu eta erabili dira proiektu honetarako aproposenak izango zirelakoan.

Irakaskuntza plataforma garatu eta amaierako emaitza ikusiz, modu egokian lan egiten duen web aplikazio bat sortu dela ikusi da eta Bilboko Hezkuntza Fakultateko ikasle batekin aztertuz, plataforma honen bitartez ikasle gazteengan sortu nahi den eragina lortzen dela ondorioztatu da.

# RESUMEN

Hoy en día, donde el uso de la tecnología es obligatorio en la vida cotidiana, el empleo de los recursos de plataformas de enseñanza también va en aumento.

Ni qué decir sobre las redes sociales, las cuales permiten compartir la vida privada tanto con amigos como con extraños. Estas, consumen una gran parte del tiempo de la gente, pero sobre todo de los jóvenes que viven en el Primer Mundo.

¿Pero aquellos que están inmersos en el mundo de las redes sociales, realmente saben utilizar estos recursos de manera apropiada y responsable?

Es cierto que no se enseña a utilizarlos. Se aprende su funcionamiento de una manera intuitiva, a medida que se van utilizando y esto, puede que produzca más perjuicios que beneficios.

Este proyecto se crea con el objetivo de advertir a los adolescentes sobre la importancia del contenido que se sube a las redes sociales y los comentarios que se escriben en él. Estando inmersos en un entorno de enseñanza, se quiere crear una plataforma llamada Zunahiz donde se pretende enseñar las consecuencias que pueden producir los actos y las palabras de una persona, al mismo tiempo que publican sus deberes de clase e interactúan con el profesor.

Para hacer esto, después de realizar un estudio, se escogieron unas tecnologías que hoy en día son muy utilizadas en la creación de aplicaciones web, pensando que estas serían las más apropiadas para el desarrollo de este proyecto.

Así pues, al completar el desarrollo de la plataforma de enseñanza y analizar el resultado final, se ha podido deducir que se ha producido la creación exitosa de una aplicación web y al examinar esta aplicación con un alumno de la Facultad de Educación de Bilbao, se ha concluido que mediante esta aplicación se consigue generar en los alumnos la influencia que se pretendía.

# ABSTRACT

Nowadays, where the use of technology is essential for everyday life, the use of teaching platform resources is also increasing.

Not to mention the social networks, which allow sharing private life with friends and strangers alike. These consume a large part of the time of the people, but especially of the young people who live in the First World.

But those who are immersed in the world of social networks, do they really know how to use these resources appropriately and responsibly?

It is true that they are not taught to use them. You learn how they work in an intuitive way, as they are being used and this may cause more harm than good.

This project is created with the objective of warning teenagers about the importance of the content that is uploaded to social networks and the comments that are written on it. Being immersed in a teaching environment, it is wanted to create a platform called Zunahiz where it is intended to teach the consequences that can produce the acts and words of a person, at the same time they publish their class duties and interact with the teacher.

In order to do this, after carrying out a study, technologies that are currently used in the creation of web applications were chosen, thinking that these would be the most appropriate for the development of this project.

Thereby, when completing the development of the teaching platform and analysing the final result, it has been possible to deduce that the successful creation of a web application has occurred and when examining this application with a student of the Faculty of Education of Bilbao, has concluded that through this application it is possible to generate in the students the influence that was intended.



# TAULEN ZERRENDA

<i>Taula 1: Orrialde euskarriaren hautatze irizpidea</i> .....	26
<i>Taula 2: Garapen moduaren hautatze irizpidea</i> .....	28
<i>Taula 3: Datu basearen hautatze irizpidea</i> .....	30
<i>Taula 4: Proiektuko partaideak</i> .....	46
<i>Taula 5: Mugarriak</i> .....	51
<i>Taula 6: Barne orduen kostua</i> .....	53
<i>Taula 7: Amortizazioen kostua</i> .....	53
<i>Taula 8: Gastuen kostua</i> .....	54
<i>Taula 9: Guztizko kostu aitortpena</i> .....	54
<i>Taula 10: Probabilitate-eragin matrizea</i> .....	57

# IRUDIEN ZERRENDA

<i>Irudia 1: Moodle plataforma</i>	14
<i>Irudia 2: Web orrialde adibidea</i>	16
<i>Irudia 3: Mahaiganeko aplikazio adibidea</i>	17
<i>Irudia 4: Web aplikazio adibidea</i>	17
<i>Irudia 5: Scriptcase tresnaren erabilera adibidea</i>	18
<i>Irudia 6: MEAN Stack logotipoa</i>	32
<i>Irudia 7: MEAN Stack-eko osagaiek sorturiko egitura</i>	32
<i>Irudia 8: Ataza asinkronoen funtzionamendu adibidea (Ataza egin)</i>	33
<i>Irudia 9: Ataza asinkronoen funtzionamendu adibidea (Ataza amaitu)</i>	33
<i>Irudia 10: Callback funtzio adibidea</i>	34
<i>Irudia 11: JSON objektu baten adibidea</i>	35
<i>Irudia 12: DOM hierarkiaren adibidea</i>	36
<i>Irudia 13: ATOM testu editorea</i>	37
<i>Irudia 14: Cygwin terminala</i>	37
<i>Irudia 15: Nodemon instalatzeko komandoa</i>	38
<i>Irudia 16: Express instalatzeko komandoa</i>	39
<i>Irudia 17: Datu basearen eduki-karpeta</i>	39
<i>Irudia 18: Mongo terminalak</i>	40
<i>Irudia 19: Mongo demonioaren terminala</i>	40
<i>Irudia 20: Mongon datu basera eskaerak egiteko terminala</i>	40
<i>Irudia 21: Mongoose instalatzeko komandoa</i>	41
<i>Irudia 22: Angular instalatzeko komandoa</i>	41
<i>Irudia 23: Angular bezeroa sortzeko komandoa</i>	41
<i>Irudia 24: Aplikazioaren arkitektura</i>	42
<i>Irudia 25: API eredu baten adibidea</i>	43
<i>Irudia 26: API kontroladore baten adibidea</i>	43
<i>Irudia 27: API bideratze fitxategi baten adibidea</i>	43
<i>Irudia 28: Main page orrialde karpeta eta edukia</i>	44
<i>Irudia 29: Comment ereduaren egitura</i>	44
<i>Irudia 30: Zerbitzu adibide bat</i>	44
<i>Irudia 31: Gantt diagrama</i>	52
<i>Irudia 32: API egituraren klaseen diagrama</i>	63
<i>Irudia 33: CLIENT egituraren klaseen diagrama</i>	64
<i>Irudia 34: Zunahiz plataformako proben planifikazioa</i>	65
<i>Irudia 35: Jasmine eta Karma tresnen logotipoak</i>	67
<i>Irudia 36: Momentuko data sortzeko testaren kodea</i>	68
<i>Irudia 37: Izen-abizenen bihurketarako testaren kodea</i>	69
<i>Irudia 38: Plataformako pertsona-bilatzailea</i>	69
<i>Irudia 39: Izen-abizenak eskuratzeko testaren kodea</i>	70
<i>Irudia 40: Data konparaketarako testaren kodea</i>	71

<i>Irudia 41: Hilabetea gehitzeko testaren kodea .....</i>	<i>72</i>
<i>Irudia 42: Hilabetea kentzeko testaren kodea.....</i>	<i>72</i>
<i>Irudia 43: Guztira burututako unitate probak .....</i>	<i>73</i>
<i>Irudia 44: Postman tresnaren logotipoa.....</i>	<i>74</i>
<i>Irudia 45: Postman tresna .....</i>	<i>74</i>
<i>Irudia 46: TypeScript-en logotipoa .....</i>	<i>74</i>
<i>Irudia 47: TypeScript bidezko integrazio proba .....</i>	<i>75</i>
<i>Irudia 48: Ikasleen ikuspuntuko erabilera kasuen diagrama.....</i>	<i>76</i>
<i>Irudia 49: Plataformako Login orrialdea .....</i>	<i>77</i>
<i>Irudia 50: Pasahitza sartzeko eremua .....</i>	<i>77</i>
<i>Irudia 51: Token bidezko sesio mantentzearen sekuentzia diagrama.....</i>	<i>78</i>
<i>Irudia 52: JWT-aren egitura.....</i>	<i>78</i>
<i>Irudia 53: Main page orrialdea.....</i>	<i>79</i>
<i>Irudia 54: Semaforo baten adibidea .....</i>	<i>80</i>
<i>Irudia 55: Profile baten orrialdea .....</i>	<i>80</i>
<i>Irudia 56: Duties orrialdea.....</i>	<i>81</i>
<i>Irudia 57: Foroa orrialdea.....</i>	<i>81</i>
<i>Irudia 58: Chat orrialdea.....</i>	<i>82</i>
<i>Irudia 59: Irakasleen ikuspuntuko erabilera kasuen diagrama .....</i>	<i>83</i>
<i>Irudia 60: Aukerak botoia.....</i>	<i>83</i>
<i>Irudia 61: Semaforo kudeaketa .....</i>	<i>84</i>
<i>Irudia 62: Irakasleak profilak kudeatzeko botoiak .....</i>	<i>84</i>
<i>Irudia 63: Eginbeharrak kudeatzeko orrialdea .....</i>	<i>84</i>
<i>Irudia 64: Ikasleak kudeatzeko orrialdea.....</i>	<i>85</i>
<i>Irudia 65: API-a abiarazteko prozesua.....</i>	<i>86</i>
<i>Irudia 66: CLIENT-a abiarazteko prozesua.....</i>	<i>86</i>
<i>Irudia 67: Plataforma hasieratzea.....</i>	<i>87</i>

# LABURDUREN ZERRENDA

<b>AJAX</b>	Asynchronous JavaScript And XML
<b>API</b>	Application Programming Interface
<b>CMS</b>	Content Management System
<b>CPU</b>	Central Processing Unit
<b>CSS</b>	Cascading Style Sheets
<b>DOM</b>	Document Object Model
<b>EHU</b>	Euskal Herriko Unibertsitatea
<b>FTP</b>	File Transfer Protocol
<b>GrAL</b>	Gradu Amaierako Lana
<b>HTML</b>	Hypertext Markup Language
<b>HTTPS</b>	Hypertext Transport Protocol Secure
<b>IKT</b>	Informazio eta Komunikazio Teknologiak
<b>JS</b>	JavaScript
<b>JSON</b>	JavaScript Simple Object Notation
<b>MEAN</b>	MongoDB, ExpressJS, AngularJS, NodeJS
<b>Moodle</b>	Modular Object-Oriented Dynamic Learning Environment
<b>PHP</b>	Hypertext Preprocessor
<b>RAD</b>	Rapid Application Development
<b>RAM</b>	Random Access Memory
<b>REST</b>	Representational State Transfer

# 1. SARRERA

Gaur egun, teknologia egunerokotasunean gero eta gehiago erabiltzen den baliabide nagusienetako bat da. Honen helburu nagusia, denbora aurrezteko eta lan esfortzua murrizteko tresna erabilgarriak sortzea da. Hortaz, hainbat arlo ezberdinetan erabiliz, gizakion beharrianak betetzeko eta bizitza errazteko erabiltzen da hau.

Honen sorkuntzaren hasiera oso urrun dagoen arren, azken mendeotan inoizko aurrerapen eta garapen gehien egin dira arlo guztietan, baina batez ere lan honen garapenerako kontutan izango dugun alderdi nagusienean, informazio eta komunikazio teknologia (IKT) alderdian.

Irakaskuntzari dagokionez, beti erabili da teknologia ikasketak errazteko: kalkulagailuak, ordenagailuak, inprimagailuak...baina Interneten erabilerak eta teknologia mugikor berrien sustapenak, aurrerakuntza ugari ekarri ditu hezkuntzara azken urteotan: arbel digital eta ikasgela birtualen sorrera batik bat.

Gaur egungo gazteen joerak aztertuz, hauen teknologia erabilera batez ere alderdi sozialera zuzentzen dela oso nabaria da. Díde ikerketa taldearen arabera Internet mundura sartzen hasteko batezbesteko adina 7 urtekoa da [1]. Urteak aurrera doazen heinean, sare sozialen erabilera beharriana sentitzen dute gazte hauek talde baten parte sentiarazteko eta hauek erabiltzen hasten dira honek ekarri ditzaken onura eta kalteak jakin gabe. *ChildWise* ikerketa agentziaren arabera, gazte hauen %30ak bakarrik erabiltzen du Interneta blogetan idazteko, eta %62ak sare sozialetan ibiltzeko erabiltzen duela onartzen du [2].

Proiektu honetan, irakaskuntza inguruan murgilduta egonik, nerabeei sare sozialen erabileraz kontzientziatu eta norbanako baten ekintzek eta hitzek izan ditzaketen eraginak irakatsi nahi dira irakaskuntza plataforma baten bidez. Horretarako lehenik eta behin nerabeak ondo ezagutu eta haien beharrianak identifikatu dira. Ondoren, beharrian horien ezaugarriak betetzen dituen irakaskuntza plataforma bat proposatu eta sortu da. Eta azkenik, sortutako plataforma horrek ondo lan egiten duela eta beharrezko ezaugarri eta funtzionaltasunak betetzen dituela probatu eta baieztatu da.

## 2. TESTUINGURUA

Marc Prensky idazleak 2001.urtean asmatu zuenetik, gaur egungo nerabeak “natibo digital” modura izendatzen dira, teknologia jadanik nahiko garatuta zegoen garaian jaioak direlako, baina batez ere Internet sortu ondoren jaioak direlako [3]. Hauek, teknologiek oso gustura aurkitzen dira haien egunerokotasuna osatzen eta hobetzen dutela ikusten dutelako.

Honen ondorioz, ikasle hauen hezkuntzan baliabide teknologikoak erabiltzeak ikasleen interesa ekintza akademikoetan handitzeaz gain, onurak sor ditzake ikasleen ikasgelako eraginkortasun eta produktibitatean.

Gaur eguneko irakaskuntza plataformak begiratu, oso garatuta dauden eta erabiliak diren plataforma ezberdinak ikus daitezke: Blackboard, Dokeos, Sakai eta Moodle besteak beste [4,5,6,7]. Hauetatik, munduan zehar erabilien den irakaskuntza plataforma 65 milioi erabiltzaile baino gehiagorekin *Moodle* da [8]. **Modular Object-Oriented Dynamic Learning Environment** izeneko irakaskuntza kudeatzeko sistema libre hau, 2001.urtean Martin Dougiamas australiarrak sortutako plataforma anitz onartzen dituen web aplikazio bat da [9].

The screenshot displays the Moodle Learning Management System (LMS) interface. At the top, the user 'Mireia Castillo Gonzalez' is logged in. The breadcrumb trail indicates the user is in a course titled 'PTA\_46'. The main content area is titled 'INFORMAZIO OROKORRA' and 'BERRIAK', showing a recent update from 2017-2018ko aurkezpena. The central section is 'TEORIA: 1. ZATIA' with a sub-section 'UML'. A list of resources is provided, including 'UML Sarrera', 'Erabilera kasuen txantiloia', 'Eskarrien ariketaren definizioa', 'Taldea lanaren hasierako aurkezpena. Taldeka.', 'Erabilera kasuen diagrama. Bakarkakoa', 'Zure erabilera kasuen igo jpg diagrama batean', 'Ekintzen diagrama. Bakarkakoa', 'Zure ekintza diagrama(k) igo jpg bat(zu)en bidez (gehienez hiru)', 'Banatu ardurak kaleetan diagramelakoren batean', 'Klaseen diagrama. Bakarkakoa', 'jpg formatuan igo', 'Sekuentzien diagrama', 'Ariketaren sekuentzia diagrama jpg batez', 'Daukagun erreminta nahiko mugatuta dagoenez ariketa hau librea da.', and 'Egoeren diagramak'. On the left, a navigation menu includes 'NABIGAZIOA', 'PARTAIDEAK', 'KUDEAKETA', 'JARDUERAK', and 'NIRE IKASTAROK'. On the right, several activity widgets are visible: 'BILATU FOROETAN', 'AZKEN BERRIAK' (listing 'PROIEKTUAK' and 'PTA\_GTELEC30\_27358\_46\_KLASE MAGISTRALETARAKO MATERIALA'), 'DATOZEN EKITALDIAK' (listing 'Proiektua aurkezteko billegia asteazkena, uztailak 4, 18:00'), and 'DUELA GUTXIKO AKTIBITATEA' (listing 'Noiztik martxan: ostirala, 2018(e)ko ekainaren 15(a)n, 12:51(e)tan').

Irudia 1: Moodle plataforma

Euskal Herriko Unibertsitateko (EHU) ikasketa prozesuan erabilitako plataforma da hau, eta honen erabileraren bitartez irakaskuntzako ekintza guztiak bai irakasle aldetik zein ikasle aldetik (zereginak kudeatu, gehitu eta erantzun, baliabideak partekatu, foro erabilera eta beste hainbeste) modu egoki eta erraz batean gauzatu daitezkeela ikusi da. Plataforma honen bidez, irakasgai baten inguruko irakasle eta ikasleen arteko hartu-emana erraz kudeatzen da.

Baina proiektu honetan, irakaskuntza plataforma bat erabiltzeaz gain, nerabeei sare sozialek izan ditzaketen eraginak eta hauen erabileraz irakastea ere bilatzen da, eta plataforma honen ez beste ezeinen bidez holakorik ez da lortu momentuz. Gainera, existitzen den plataforma hori, haur hezkuntzako gazteentzat erabiltzeko zaila gerta daiteke hasieran, batez ere horrelako plataformarik inoiz erabili ez dutenean.

Hartara, *Moodle* plataforma horren eta sare sozialen ezaugarriak dituen web aplikazio bat diseinatu eta sortu izan nahi da proiektu honetan. Erreminta berri honen bitartez, hezkuntza eremua interesgarriago eta interaktiboagoa bihurtu nahi da ikasleen arreta mantenduz.

Ikusten denez, irakasle ikuspuntutik ere oso onuragarria gerta daiteke horrelako web aplikazio bat, garapen teknologiko ezberdinen bitartez haien eginbeharreko lana erakargarriagoa eta efizienteagoa bihurtu daitekeelako.

Guzti hau betetzeko, sortutako plataforma gaur eguneko teknologia eta erabileretara egokitzeko asmoz artearen egoera ikertu eta kontutan izateaz gain, Bilboko Hezkuntza Fakultateko Lehen Hezkuntzako Gradu ikasle batekin lan egin da. Ikasle honek, bere Gradu Amaierako Lan (GrAL) modura nerabeen beharrianak aztertu eta hauek betetzen dituen plataforma baten zehaztapenak proposatu zituen. Hortaz, abiapuntu horiek kontutan izanik sortu da *Zunahiz* izeneko plataforma hau.

## 3. TEKNOLOGIA EREMUA

Gaur egun, aplikazio berrien garapena oso ekintza garrantzitsua da IKT eremuan zehar. Orrialde horiek euskarri egitura ezberdinen bidez eta teknologia ezberdin asko erabiliz sor daitezke, bilatzen den erabileraren arabekoak direnak.

### 3.1 Orrialde euskarriak

#### 3.1.1 WEB ORRIALDEA

Web orrialde bat, testu, irudi, soinu, estekak eta beste hainbat gauza izan ditzaken dokumentu bat da eta hau web nabigatzaile baten bidez atzitu daiteke. Orrialde hauetako edukia normalean estatikoa izaten da. Hau da, informazioa irakurri edo edukia ikus daiteke, baina inoiz ez haiekin elkar eragin [10].



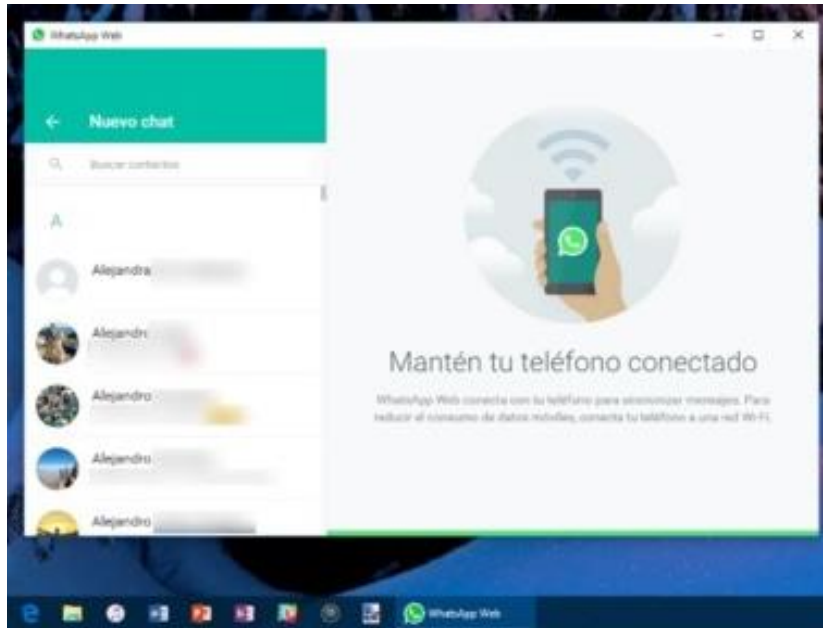
Irudia 2: Web orrialde adibidea

#### 3.1.2 MAHAIGAINEN APLIKAZIOA

Mahaigaineko aplikazio bat, ordenagailuan instalatzen edota exekutatzeko programa bat da. Adibidez: Word, Excel, OpenOffice etab.

Horrez gain, aplikazio hauek webgune bitartez datuekin ere lan egin dezakete. Horietako batzuk Outlook, Whatsapp eta eMule direlarik [10].

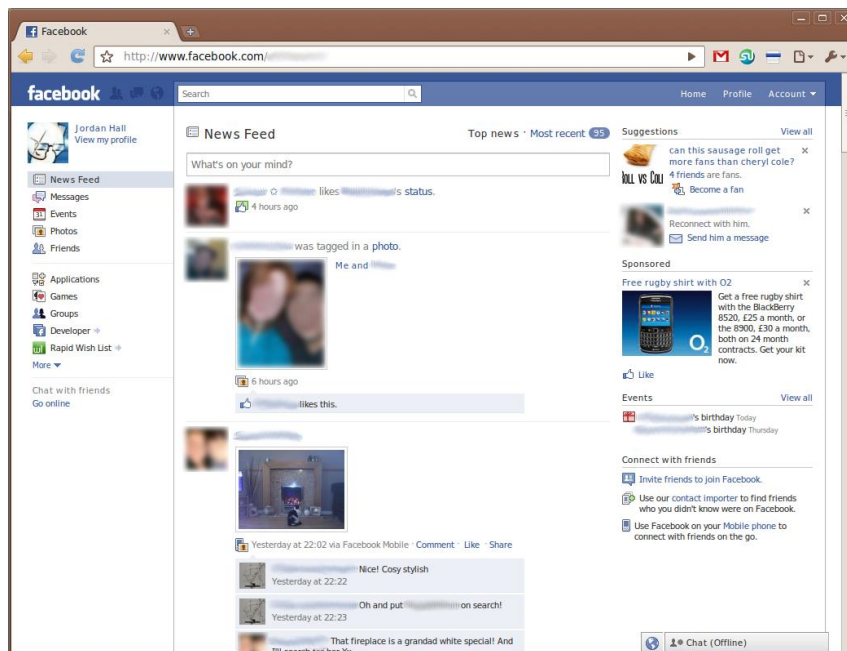




Irudia 3: Mahaigaineko aplikazio adibidea

### 3.1.3 WEB APLIKAZIOA

Web aplikazio bat, informazioa duen eta horrekin elkar eragin edo bere edukia aldatu daitekeen web orrialde berezia da. Mahaigaineko aplikazio batetik desberdintzen duena, honek ordenagailuan instalatzeko edo bertatik exekutatzeko beharrik ez duela da, honetara nabigatzaile batetik atzitzen delarik. Hauen adibide gisa, Hotmail, Google Docs eta Facebook har daitezke [10].



Irudia 4: Web aplikazio adibidea

Proiektu honen ezaugarriak eta beharrianak kontutan izanik, plataforma garapenerako web aplikazio bat egitea egokiena dela erabaki da. Honen erabakia, 6.1 atalean justifikatzen delarik.

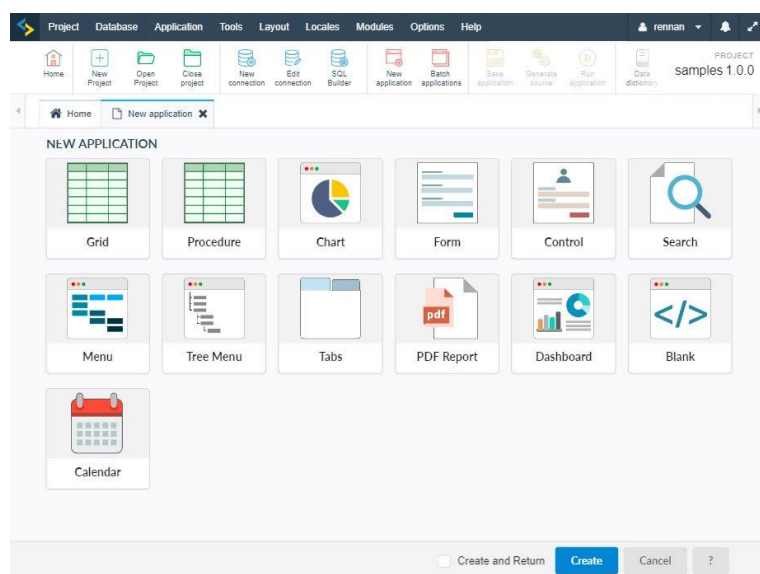
## 3.2 Garapen modua

Gaur egun web aplikazioak sortzeko modu asko daude. Alde batetik, teknologia ezberdinak erabiliz, hutsetik has daiteke proiektu bat baina, hau ekiditeko eta plataforman sorkuntza errazteko erabiltzen diren hainbat teknika ere daude. Atal honetan, aplikazio horien sorkuntzarako erabiltzen diren metodo erabilienak aztertuko dira. Alde batetik, proiektua berez hutsetik hastea eta bestetik, Scriptcase izeneko tresna baten bitartez garatzea [11].

Atal honetan aukera bakoitza zertan datzan azalduko da. Gero, 6.2 atalean, aukera bakoitzaren abantailak eta desabantailak aztertuz proiektu honen garapenerako zein aukeratuko den eta zergatik azaltzen delarik.

### 3.2.1 SCRIPTCASE

Web aplikazioak sortzea konplexuagoa denez eta programazio denbora gehiago eskatzen dutenez, garapen denbora hori murrizteko erabiltzen diren baliabideak existitzen dira. Honen adibide bat Scriptcase izeneko tresna hau da.



**Irudia 5: Scriptcase tresnaren erabilera adibidea**

Web orrialdeak sortzeko WordPress tresna famatua erabiltzen den moduan, web aplikazioak sortzeko Scriptcase tresna hau erabiltzen da. Merkatuko datu base erabilienekin konexioak ahalbidetzen dituen bitartean, web aplikazioak sortzerakoan oso erabiliak diren aplikazioak eskaintzen ditu: erregistro formularioak, taulak, egutegiak, menuak, grafikoak, erabiltzaile kudeaketa...[12].

RAD (Rapid Application Development) tresna hau Netmake garatzaile enpresak sortu zuen 2000. urtean eta Mac, Windows eta Linux sistema eragileetan erabili daiteke. Honek, PHP lengoia erabiliz (kasu gehienetan) sortzen ditu aplikazioak XAMP ingurumenean [13].

XAMP hau sistema eragilearen araberako pila da eta honek Apache (zerbitzaria), MySQL/MariaDB (datu basea) eta Perl/PHP/Python (lengoia) erabiltzen ditu [14]. Linux erabiltzen den kasuan LAMP deitzen da, Windowsen WAMP, MacOsen MAMP...

Software hau erabili ahal izateko, 500€ inguru ordaindu behar dira eta behin programa guztiz sorturik dagoela, Scriptcase tresna hau ez da gehiago erabili behar aplikazioa abiarazteko.

### **3.2.2 HASIERATIK KODETU**

Erabilera jakin bat eta ezaugarri zehatz batzuk izan behar dituen web aplikazioak egiteko, norbanakoek eta enpresetako IKT sailek gero eta gehiago erabiltzen dute proiektua haien modura eta gustuko teknologiekin sortzeko aukera. Hutsetik kodetzeak aplikazioaren abiaraztea moteltzen dituzten programa eta softwareen erabilerak aurrezten ditu.

Gainera, berezko datu base bat eta aplikazioko edukiak elikatzen dituen FTP (File Transfer Protocol) izateak erabiltzaileari begirako konpromisoa sortzea baimentzen du. Hauen bitartez, abiarazteko prozesua samurtzeak aplikazioa erakargarriago egiten duelarik erabiltzaileek orrialdean ibiltzeko grina sustatzen lagunduz [15].

Egia da diseinu bat hasieratik programatzea zeregin neketsua dela eta alde zurretik diseinatutako widget eta pluginak erabiliz lana aurreztu ahalko litzatekeela. Web aplikazio bat funtzionaltasunez eta dinamikotasunez sortzea Scriptcase bezalako CMS (Content Management System) batekin edozeinen eskumenean dago, baina sortu beharreko proiektuaren ezaugarri bereziek eta konplexutasunek azken batean, erabiltzen

diren txantiloak eta estilo orriak aldatzera eraman zaitzakete hasieran oso erraz ikusten zen prozesu hori guztia zailduz.

Segurtasun aldetik aztertuz, edukiak kudeatzeko sistema batekin baino segurtasun handiagoa lortzen da diseinu bat hutsetik hasiz. Hackerrek gailentasun pixka bat duten orrialdeetan erasotzen dute segurtasun gabeko zirrikitu ñimiñoak aprobetxatuz, hauen fama zikindu nahian. Diseinua hutsetik hasiz aldiz, bakarra den kodea sortzen da, eta honen bidez, txantiloiek izaten dituzten egituratik bereizi eta hackerren lana asko zailtzen da [15].

Ezaugarri hauez gain, norberak aplikazioa eta honen kode osoa hutsetik hasiz, programa osoaren kodea ezagutzea dakar. Hau, aplikazioa kudeatzeko eta planifikatzeko orduan, bai eta aldaketak egiteko edo akatsak bilatzeko orduan asko laguntzen duen alderdi bat delarik, hirugarren batzuen lanari adi egoteko beharrik izan gabe.

Proiektua hutsetik hasteko hainbat lengoia daude: JavaScript (JS), ASP, PHP, Python, Ruby...baina gaur egun merkatuan gehien erabiltzen diren lengoaiak PHP eta JavaScript dira.

Ezin daiteke esan bata beste baino hobea dela, lengoia bakoitza erabiltzea edo ez garatzailearen lehentasunen eta sortu beharreko funtzionaltasunen beharrianen araberakoa da.

Proiektu honi dagokionez, merkatuan dauden aukerak eta bakoitzak proiektuari eskaintzen dizkion ezaugarriak kontutan izanik, web aplikazio bat sortzea erabaki da eta honen garapena hutsetik hasiz egingo da. Esan bezala, erabaki hauen zergatia 6.1 eta 6.2 ataletan ikus daiteke.

## 4. HELBURUAK

### 4.1 Helburu nagusia

Proiektu honen helburu nagusia irakaskuntza inguruan murgilduta egonik, nerabeei sare sozialen erabilera egokiaren beharraz jabetzeko eta norbanako baten ekintzek eta hitzek izan ditzaketen eraginak irakasteko asmoa duen irakaskuntza plataforma bat sortzea da.

Helburu nagusi hori lortu ahal izateko, lehendabizi tarteko beste helburu partzial batzuk bete behar dira. Bete beharreko helburu horiek hurrengo atalean adierazten dira.

### 4.2 Helburu partzialak

#### 4.2.1 PLATAFORMA DISEINUA DEFINITU

##### *4.2.1.1 Nerabeen beharrianetara egokitu*

Lehenago aipatu den bezala, proiektu honetan Bilboko Hezkuntza Fakultateko ikasle batekin egin da lan. Honen lanaren bitartez, aztertutako beharrianak kontutan izanik eta telekomunikazio ikaslearen aldaketa batzuen proposamenak aztertuz, ezaugarri horiek betetzen dituen eta nerabeentzat egokia den plataforma baten diseinua egin da.

##### *4.2.1.2 Beharrianen zehaztapenak eta goi mailako diseinua*

Nerabeen beharrianetara egokitzen diren zehaztapenak lorturik, horiek beteko dituen plataformaren zehaztapenak eta goi mailako diseinua egin beharko da.

#### 4.2.2 PLATAFORMA EGITURA DEFINITU

##### *4.2.2.1 Zerbitzari eta bezero aldean egituratu*

Lortu behar den emaitza kontutan izanik, emaitza hori lortuko duen egitura sortu behar da. Horretarako, bezero eta zerbitzari aldeetan erabiliko diren eredu, osagai, funtzio, hasierako eta lehenetsitako balioak, ereduaren arteko erlazioak eta beste hainbat alderdi definitu beharko dira.

Alde bakoitzean definitutako egitura *I.eranskin*-ean ikus daiteke klase diagramen bidez.

#### *4.2.2.2 Plataforma sortu*

Behin erabiliko diren eredu, osagai, erlazio eta abarrekoak ondo definituta daudela eta atal guztien artean osotasun bat dagoela, atal hauek programatzeari ekin behar zaio hau izango baita proiektu honen funtsa eta denbora gehien eramango duen zatia.

### **4.2.3 EMAITZEN AZTERKETA BURUTU**

#### *4.2.3.1 Plataformaren funtzionamendua egiaztatu*

Beste helburu partzialetako bat plataformaren funtzionamendu egokia egiaztatzea da. Hemen, plataformak modu dinamikoan ondo lan egiten duela, lehen erabileran ulertzeko erraza dela, hau da, intuitiboa dela, eta eskaintzen diren funtzioek bere betebeharrekoak lortzen dituztela konprobatzen da.

Egiaztapen hauek egiteko *II.Eranskin* atalean gehiago zehaztuko diren unitate, integrazio eta end-to-end probak erabiliko dira.

#### *4.2.3.2 Zehaztapen sortzailearen onespena jaso*

Azken emaitza modura, lortutako plataforma Bilboko Hezkuntza Fakultateko ikasleari erakutsiko zaio eta honen hasierako itxaropenak betetzen direla baieztatu beharko da. Hau da, plataforma honen bitartez nerabeei irakatsi beharreko baloreak irakasten zaizkiela ziurtatu.

## 5. ONURAK

Proiektu honek onurak dakartza arlo ezberdin batzuetan. Horietatik, arlo soziala, teknikoa eta ekonomikoa nabarmenduko direlarik.

### 5.1 Onura sozialak

Proiektuaren onura nagusiak arlo honetan kokatzen dira. Proiektu honen bitartez gazteak direnei sare sozialen erabilpen egokia eta hauetan esaten diren hitzek duten garrantzia irakasten bazaizkie, aurrerapen handiak lortu direla esan daiteke.

Alde batetik, irakaskuntza arloari dagokionez, irakasleari eta gelako beste kideei behar den moduan, errespetuz tratatzen irakasten duelako, eta beste aldetik, argitaratu daitekeen edukiaren egokitasunaz jabetzen ere laguntzen duelako. Gainera, irakasleari dagokionez, plataforma honen bitartez eta ikasle bakoitzaren erabilera kontutan izanik, irakasleak ikasle bakoitzaren joera eta portaera zein den ere aztertu ahalko duelako.

Guzti honez gain, plataforma hau oso onuragarria gerta daiteke honen bitartez balore hauek nahiz eta irakaskuntza giroan ikasi, gerora etorkizuneko beste sare sozialetan aplikatu beharreko gomendioak irakasten zaizkielako ikasleei.

### 5.2 Onura teknikoak

Proiektu honen onura tekniko nagusia garatutako irakaskuntza plataforma izan da. Eskoletan oso erabilgarria gerta daiteke, ikasleengan ohitura sozial onak sortzeaz gain, ikasketak kudeatzen ikasteko plataforma egokia izan daitekeelako. Jadanik existitzen diren irakaskuntza plataformak baino sinpleagoa denez, honek zubi modura lan egin dezake, ezerezetik plataforma hain konplexuekin lan egiten hasi baino lehen.

Horrez gain, proiektuan zehar garatutako tresnak beste proiektu bat sortu nahi denerako erabilgarriak gerta daitezke. Alde batetik, tresna horiek haien osotasunean berrerabili nahi diren kasuetarako eta beste aldetik, antzeko zerbait sortu nahi denerako hauek erreferentzi edo eredu modura erabiliz.

## 5.3 Onura ekonomikoak

Plataforma honek arrakasta izatekotan libre izateari utziko balio, hezkuntzan erabiltzeko asmoz saldu ahalko litzateke eta horrela proiektuari onura ekonomikoak sortu. Baina, orain aipatutakoa ez da proiektu honen sorkuntzaren helburuetako bat.

# 6. ALTERNATIBEN ANALISIA

Atal honetan plataforma sortzeko modu ezberdinak aztertuko dira, proiektura gehien egokitzen dena zein den erabakiko delarik. Horrez gain, erabaki hori aurrera eramateko eta horrela sortu ahal izateko, esparru ezberdinetan erabili behar diren teknologia ezberdinak aztertu eta alternatiba egokienak aukeratuko dira. Horretarako, aplikazioan eskaini nahi diren funtzionaltasun batzuen arabera irteera bakoitzak zer eskaintzen duen kontutan izango da.

## 6.1 Orrialde euskarriak

Irakaskuntza plataforma euskarri ezberdinen bitartez sortzeko aukera dagoela ikusi dugu lehen, 3.1 atalean. Bakoitza zer den jadanik jakinik, hiru aukerak aztertu eta proiektu honi gehien egokitzen zaiona zein den erabaki da.

### 6.1.1 WEB ORRIALDEA

Web orrialde bat, testu, irudi, soinu, estekak eta beste hainbat gauza izan ditzaken dokumentu bat den arren, hau ez dago bezeroaren aldetik informazioa batzeko prestatuta. Horrelako web orrialde estatikoak gehienetan enpresen informazioa bezeroei bistartzeko edo ekintza berrien berri emateko erabiltzen dira: produktuen deskribapena, eskaintako zerbitzuak, enpresaren historia, harremanetarako telefono, posta elektronikoa etab.

Garapen aldetik, ez du programazio lengoia bat ikasteko ezta datu base bat erabiltzeko beharra sortzen. HTML, CSS eta gutxi batzuetan JavaScript erabiliz sortzen baitira.



## **6.1.2 MAHAIGAINENKO APLIKAZIOA**

Mahaigaineko aplikazioak, web bertsioak baino software osoagoak diraz. Baliabide gehiago erabiltzeko ahalmena dute (CPU, RAM memoria..) eta egonkorragoak dira. Akatsak egotekotan, berreskuratze ahalmen handiagoa ere eskaintzen dute. Baina aplikazio hauek beti ez daude erabiltzaile guztien menpe, sistema eragileari lotuta baitaude. Horrez gain, erabiltzaile ekipotik bakarrik erabili daiteke bertan instalatzen direlako, ekipoko memoria zati bat horrentzat besterik erabiliko delarik.

## **6.1.3 WEB APLIKAZIOA**

Web aplikazio bat, web orrialde eta mahaiganeko aplikazio baten bitartekoa da. Honek, bezeroekin elkarrekintza sustatzen du baina ordenagailuan ezer instalatzeko beharrik gabe. Web aplikazioek, web orrialdeek baino baliabide gehiago erabili behar dituzte (aplikazioaren betebeharren arabera, datu gehiago prozesatu eta datu baseetara atzipena areagotu behar dutelako) baina ez mahaigaineko aplikazioen beste.

Garapen aldetik, hauek ezin dira HTML jakinda bakarrik kodifikatu web orrialde estatikoetan egiten den bezala. Hauetan, lengoia konplexuagoak erabiltzen dira aplikazio aldea sortu ahal izateko. Besteak beste, Java, JavaScript, PHP eta AJAX [16].

## Hautatze irizpideak

### **ERABILTZAILEAREKIN ELKARREKINTZA**

Ikasleentzako ikasketa plataforma bat denez, hauekin elkarrekintza bat bermatu behar du aplikazioak ikaslearen datu eta informazioak jaso ahal izateko, plataforma dinamikoa eginez eta estatikotasuna ekidinez.

-Irizpide honen pisua: %60

### **ERRAZTASUNA**

Plataforma garatzerako orduan, erabiliko diren teknologi eta lengoaien zailtasuna ahalik eta gutxien izatea komeni da gauzak garapen prozesuan errazteko.

-Irizpide honen pisua: %20

## BALIABIDE ERABILERA

Baliabide gehiago erabiltzeko ahalmenak, plataforma erabiltzerako orduan esperientzia hobea eta arinagoa lortzen lagun dezake.

-Irizpide honen pisua: %10

## EROSOTASUNA

Plataforma aplikazio modura ekipoetan instalatu behar ez izateak eta beste edozein ordenagailu erabiliz plataformara sarrera lortzeko aukera izateak, irakaskuntza plataforma erabiltzea askoz erosoago eta esperientzia atsegingarriagoa izatea eragin dezake.

-Irizpide honen pisua: %10

	Pisua	Web orrialdea	Mahaiganeko aplikazioa	Web aplikazioa
Erabiltzaile elkarrekintza	%60	0	9	10
Erraztasuna	%20	8	1	2
Baliabide erabilera	%10	5	8	5
Erosotasuna	%10	7	6	9
		2,8	7	7,8

Taula 1: Orrialde euskarriaren hautatze irizpidea

Hiru aukeren arteko konparaketa ikusiz, plataforma web aplikazio modura garatzea, mahaiganeko aplikazio modura garatzea baino erabaki hobea dela ondorioztatzen da.

## 6.2 Garapen modua

Behin web aplikazio bat egingo dela erabakita, aplikazio hori nola garatuko den erabaki behar da eta honetarako bi aukeren arteko alderaketa bat egin da. Lehenengoa, Scriptcase tresna bitartez egitea da eta bigarrena, aplikazioa hutsetik hastea.

### **6.2.1 SCRIPTCASE ERABILI**

Tresna honen bitartez, aplikazioak sortzeko erraztasun asko lortzen dira. Kodea ez da hutsetik sortu behar eta elementuak integratzeko aukera asko eskaintzen ditu. Erraz ulertzeko moduko kode intuitiboaz sortzen delarik, oso eroso da eta datu baseekin modu egokian lan egiten du [17]. Baina hala ere, tresna hau ez da dohainekoa eta gainera proiekturako diseinatutako elementu jakin batzuk sortzeko aukera (argitalpenen semaforoak adibidez) ez luke ahalbidetuko.

### **6.2.2 KODEA HUTSETIK HASI**

Plataforma modu honetan sortuz, proiekturako diseinatu diren elementuak gustura sortu ahal izango dira inongo mugarik izan gabe. Horrez gain, hemen erabili beharreko teknologien artean libreak direnak aurki daitezke projektuko lizentzien kostua ekidinez. Hori bai, berrerabili daitezkeen eta jada sortuta dauden moduluen kodeak izan ezik, beste dana hutsetik kodifikatu eta integratu behar da.

## Hautatze irizpideak

### **LANA ARINTZEA**

Prozesu luzea izan daitekeenez, programazio aldetik kode osoa idatzi ez beharrak prozesuan lagundu eta denboraz aurreratu dezake. Gainera, barne-orduen kostua gutxiagoa izatea eragingo luke.

-Irizpide honen pisua: % 15

### **ERRAZTASUNA**

Plataforma garatzerako orduan, garatzailearentzat erreza eta intuitiboa gertatzea garrantzitsua da lana eramangarriago eta erosoago sortu ahal izateko.

-Irizpide honen pisua: % 15

### **LIZENTZIAK EKIDIN**

Garapenean lizentzia gabeko tresnak erabiltzeak proiektua merkeagotu egingo du.

-Irizpide honen pisua: % 10

## DISEINURA EGOKITU

Plataforma hasierako diseinura egokitzea eta bertan definitutako elementuak sortu ahal izatea da proiektuaren alderdi garrantzitsuenetariko bat. Hau egin ezean, proiektua ez da osatuta geratuko eta ez da bat etorriko proiektuaren helburu nagusiarekin.

-Irizpide honen pisua: %60

	Pisua	Scriptcase	Kodea hutsetik
Lana arintzea	%15	9	0
Erraztasuna	%15	8	5
Lizentziak ekidin	%10	0	10
Diseinura egokitu	%60	5	9
		5,55	7,15

Taula 2: Garapen moduaren hautatze irizpidea

Aukera biak alderatuz, proiektuaren garapenerako kodea hutsetik hastea erabaki da.

## 6.3 Datu basea

Plataforman erabiltzen den informazioa (erabiltzaile datuak, argitalpenak..) gordetzeko beharra dago. Horretarako datu base bat sortu beharko da.

Gaur egun, merkatuan, bi motako datu baseak nagusitzen dira: datu base erlazionalak eta datu base ez erlazionalak. Hori dela eta, mota bakoitzeko tresna bat aukeratu da alderaketa egiteko, MySQL alde batetik eta MongoDB beste aldetik. Horrela, hauen erabilera ikasteko eta behar izatekotan hauen komandoak bilatzeko laguntza gehiago aurki daitekeelako sarean.

### 6.3.1 MYSQL

Datu base erlazionalak kudeatzeko sistema hau, kode irekiduna da eta hortaz, ez luke lizentzia kosturik sortuko proiektuan. Informazioa tabletan gordetzen du, eta datuen atzipenerako SQL eskaera lengoaia erabiltzen du. Gainera, datu basearen egitura definitzeko eskemak erabiltzen ditu [18].

Erlazionala izateak, hainbat abantaila ditu: erregistroak bikoiztuta ez egoteko tresnak erabiltzen ditu, erreferentziak erabiltzen direnez erregistro bat ezabatzean honen menpekoak diren beste guztiak ezabatzen ditu automatikoki, taulen arteko erlazioak erakusten dituenaz ulertzeko errazagoa gerta daiteke...baina bere desabantailak ere baditu. Datu grafiko eta multimedia datuak erabiltzeko zailtasunak ditu, testu blokeak datu modura erabiltzeko ez da oso ondo maneiatzen, datu base hauen sorrera eta mantenua oso garestia da... [19]

Hala ere, bere konektagarritasun, abiadura eta segurtasun onek MySQL zerbitzaria Internetetik datu baseak atzitzeko tresna aproposa bihurtzen dute.

### 6.3.2 MONGODB

Datu base ez erlazionalak (NoSQL) kudeatzeko sistema hau, kode irekiduna da ere eta hortaz, ez luke lizentzia kosturik sortuko proiektuan. MySQLn ez bezala, datuak BSON deituriko dokumentu batzuetan gordetzen ditu. MongoDB eskema gabeko datu basea da, eta honek dokumentuak eratzea ahalbidetzen du egitura bat definitu beharrik gabe [20].

Erlazionala ez izateak ere bere abantailak ditu: datuen eredia oinarrizkoa eta sinplea denez malgutasunez eta eraginkortasunez erabil daiteke, eskalagarria da, onartzen dituzten eragiketak sinpleak dira eta denboran oso azkar exekututzen dira, datu kopuru handia kudeatzeko gai dira...baina bere desabantailak ere baditu. Enpresa munduan oraindik oso erabiliak ez direnez hauen sinesgarritasuna ez da oso handia, heldutasuna falta dute oraindik eta ezegonkortasun arazoak eman daitezke...[21].

Hala ere, gero eta gehiago erabiltzen da datu base mota hau, gero eta ezaugarri gehiago dituelarik eta sistema eragile ezberdinetan (Windows, Linux...) erabilgarria delarik.

# Hautatze irizpideak

## ESKALAGARRITASUNA

Aplikazioa garatzen doan heinean, funtzionaltasunak gehitzen joango zaizkio, honek datu basean datu multzo berrien beharrezana sortzen duelarik. Horregatik, eskalagarritasuna irizpide garrantzitsu bat izango da aukeraketan.

-Irizpide honen pisua: %15

## ABIADURA

Datuen eskaera egitean, ematen den erantzunaren abiadura ere garrantzitsua izango da.

-Irizpide honen pisua: %25

## INPLEMENTAZIO ERRAZTASUNA

JavaScript lengoaia erabiliz proiektuan erabiliko diren beste teknologiekin lan egiteko erraztasuna kontutan hartzeko irizpide garrantzitsuena izango da.

-Irizpide honen pisua: %40

## AURRETIAZKO EZAGUERA

Software-a garatzen hasi aurretik, teknologiarekiko ezagutza maila garrantzitsua da, izan ere, horrek garatzeko behar den denboran eragin zuzena izango baitu.

-Irizpide honen pisua: %20

	Pisua	MySQL	MongoDB
Eskalagarritasuna	%15	5	8
Abiadura	%25	6	9
Inplementazio erraztasuna	%40	6	8
Aurretiazko ezaguera	%20	7	3
		6,25	7,25

Taula 3: Datu basearen hautatze irizpidea

Alternatiba bien konparaketa aztertuz, MongoDB erabiltzea erabaki da. Tresna honekiko ezagutza maila altua izan ez arren, hezkuntza plan bat abian jarriko baita ikastaro batzuen bitartez proiektuak behar duen ezagutza maila eskuratzeko.

## 7 METODOLOGIA

Lan honetan aipatu diren helburuak lortzeko, proiektuan zehar jarraitu den metodologia azalduko da. Metodologia hau bost atal nagusitan bereizten da: azterketa, prestaketa, garapena, probak eta emaitzak.

### 7.1 Teknologiak aztertu

Lehenago aipatu den bezala, proiektu honen garapenean ez da graduan zehar ikasitako lengoia ez teknologiarik erabili. 3.2 atalean ondorioztatu den moduan, proiektu hau hutsetik kodetu da inongo eduki kudeaketa sistemarik erabili gabe.

Horretarako, lehenengo pausua proiektuan zehar oinarri modura erabiliko zen lengoia aukeratzea zen.

#### 7.1.1 JAVASCRIPT LENGOAIA

Web aplikazioen garapenean gero eta protagonismo gehien hartzen doan lengoia dela ikusita, hau aukeratu zen proiektu honen oinarri nagusitzat.

Hasieran bezero aldean (nabigatzailean) bakarrik erabiltzen zen hau; gaur egun aldiz, zerbitzari eta biltegitratze mailan ere egiten du honek lan. Horri esker, gaur egun fase eta geruza guztietan lengoia bera erabiliz (JavaScript) sor daitezke aplikazioak, eta ezaugarri hau, lengoia hau erabiltzeko erabakian kontutan izandako eragile bat izan da.

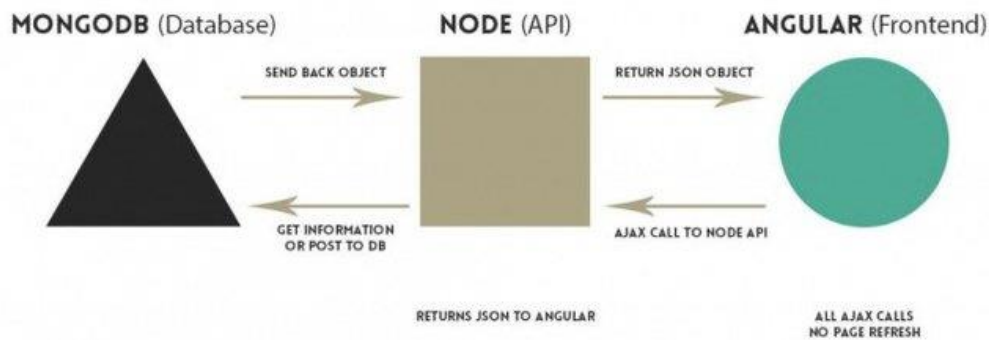
Efektu ezberdinak eta ekintza interaktiboak sortzeko aproposa izanik, hau, chat, kalkulagailu, bilatzaile eta horrelako milaka tokitan erabiltzen da.

Hortaz, behin JS erabiliko zela onartuta, ingurumen honetan MEAN Stack izeneko (MongoDB + ExpressJS + Angular + NodeJS) web aplikazioak sortzeko framework bat jadanik bazegoela eta oso erabilia zela ikusi zen.



Irudia 6: MEAN Stack logotipoa

Framework horretako osagaiak aztertu ziren eta horien bitartez plataformako helburuak lortzeko ahalmena zegoela ikusirik, plataforma framework horren bitartez garatu zen.



Irudia 7: MEAN Stack-eko osagaiak sorturiko egitura

### 7.1.2 NODE JS

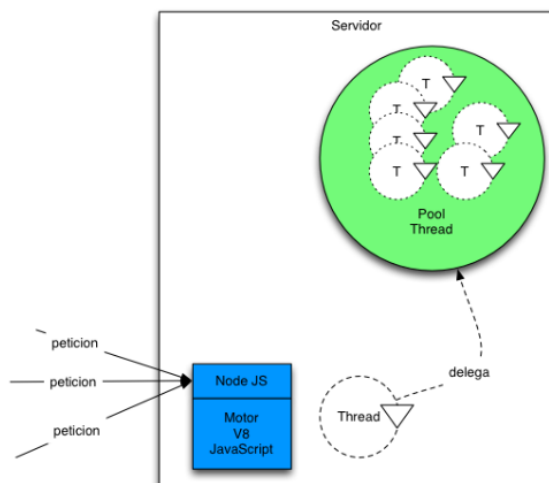
Zerbitzari aldearen funtzionamenduaz arduratzen den plataforma da hau. Honek, Googleko JavaScript motorea (V8) erabiltzen du barnetik eta nahiz eta erabilpen asko izan, web aplikazioak garatzeko erabiltzea da honen erabilera garrantzitsua. Motore horren bidez, nabigatzaileak JavaScript kodea izugarritzko abiaduran exekutatzen duelarik [22].

Teknologia honen alde tekniko garrantzitsua exekuziorako hari (thread) bakarra erabiltzen duela da, hau arduratzen delarik egin beharreko lan guztia antolatzeaz. Egin beharreko atazak modu asinkronoan egiten ditu tresna honek eta asinkronotasun hori kudeatzeko *callback* funtzioak erabiltzen ditu.

NodeJS eta ataza asinkronoen arteko funtzionamendua honakoa da:



*Node*-k, hari bakarraren bitartez eginbeharreko guztiak *libuv* liburutegiaren bitartez sorturiko *pool of threads* baten esku uzten ditu, pool honek bere barnean multithread asinkronoen ingurumen bat duelarik.



**Irudia 8: Ataza asinkronoen funtzionamendu adibidea (Ataza egin)**

Hortaz, *Node*-k *pool*-ari zereginak bidali eta behin honek eginbeharren bat bete duela, *libuv* liburutegiak gertaera bat bidaliko du *Node*-k jasoko duelarik.



**Irudia 9: Ataza asinkronoen funtzionamendu adibidea (Ataza amaitu)**

Behin gertaera jasota, *callback* funtzioak gertaera hori prozesatuko du. Hori dela eta, *Node*-rekin lan eginez programazioa asinkronoa dela esan daiteke, AJAX (Asynchronous JavaScript And XML) deien antza dutelarik [22].

```

function getFile(req,res){
  var fileId=req.params.id; Callback arduraduna
  file.findById(fileId) (err,file)=>{
    if(err){
      res.status(500).send({message: 'Errorea fitxategia lortzean'});
    }else{
      if(!file || file==false){
        res.status(404).send({message: 'Fitxategi hori ez da existitzen'});
      }else{
        res.status(200).send({file});
      }
    }
  }
};
}

```

**Irudia 10: Callback funtzio adibidea**

Irudi honetan, proiektuan garatutako funtzio bat ikusten da *Callback* funtzioen erabileraren adibide gisa.

Proiektu honetan hortaz, teknologia honen bitartez garatu da plataformaren *back-end* zerbitzaria. Honek, http protokoloaren bidez irakaskuntza plataformaren API REST (Application Programming Interface, Respresentational State Transfer) zerbitzuak eskaintzen dituelarik.

### 7.1.3 EXPRESS JS

HTTP protokoloaren metodoen bitartez (GET,POST...) egiten diren eskaerak kudeatzeko JavaScript metodo aski eskaintzen dituen NodeJS-ren modulu bat da. Horrez gain, bideraketa sistema bat ere eskaintzen du MEAN Stack barnean *back-end* aldean (hau da, zerbitzariarenean).

Framework hau eskaera/erantzun, goiburu, ibilbide, bista eta abarren kudeaketarako erabiltzen du Node-k batez ere, eta hau gabe web aplikazio bat sortzea askoz ere konplexuagoa izango litzateke.

### 7.1.4 MONGO DB

Lan honetan lehenago aipatu den moduan, NoSQL motako datu base hau erabili da proiektu honetan. Eredu ez-erlazional honen bitartez malgutasun eta eskalagarritasun handiagoko aplikazioak sortzeko aukera dagoelarik. Hau da, gorde beharreko datu kopurua itzelean handituz arren, honek ez du datu basearen funtzionamenduan eraginik izango. Gainera, ez erlazionala izateak taularik, erregistorik, *join*-ik, loturarik eta *primary key*-rik ez egotea eragiten du.

Honek, datuen transferentzia eta biltegitzea JSON (JavaScript Simple Object Notation) formatuan egiten du. Izenak esaten duen bezala, JSON-ek objektuak (egitura konplexuak bilaka daitezkeenak) JavaScript kode modura adieraztea ahalbidetzen du, datuak dokumentutan gordetzen direlarik [23].

Formatu hau arina izateaz gain, gizakiek eta ordenagailuek ulertzeko erraza da. Horren adibide gisa ondorengo irudia:

```
{  
  "izena" : "Libe",  
  "abizenak" : "Iturraspe Markaida",  
  "unibertsitatea" : "EHU",  
  "telefonoa" : "986 165 802",  
  "adina" : 20  
}
```

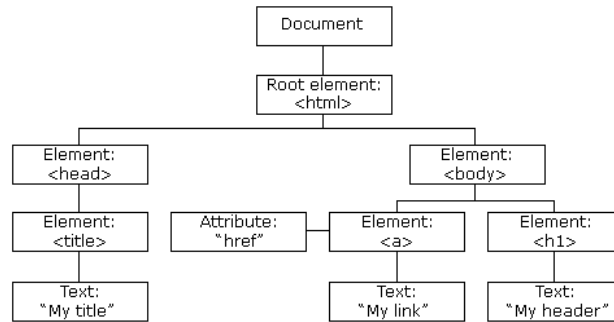
**Irudia 11: JSON objektu baten adibidea**

## 7.1.5 ANGULAR

HTML, CSS eta JavaScript soilik erabiliz erabiltzaile alde osoa garatu daitekeen arren, funtzionaltasun asko eginda eskaintzen dituen liburutegi bat erabiliz lana asko erraztuko litzateke.

Nabigatzailean beti existitu dira garapena errazteko balio izan duten funtzioen liburutegiak, horietatik erabiliena jQuery izanik [24]. Baina azken urteotan nabigatzaileetan MVC eredua (Model-View-Controller) zabaldu den heinean, bere erabilpena errazteko hainbat liburutegi ezberdin sortu dira. Horietatik gehien nabarmentzen dena *Angular* delarik, batez ere *Googlek* sortzen eta kudeatzen duelako eta kode irekiduna delako [23].

Hortaz, *Angular* liburutegi simple bat izan beharrean, funtzionaltasun aurreratuak eskaintzen dituen eta HTML zabaltzen duen framework oso bat dela esan daiteke. Honek, efektu ezberdin asko modu errazean sortzea ahalbidetzeaz gain, DOM hierarkiaren (Document Object Model) maneia errazten du.



**Irudia 12: DOM hierarkiaren adibidea**

Horrez gain, Angular bidez sistemaren bideratzea ere egin daiteke eta AJAX funtzioa erabili zerbitzariari eskaerak egiteko orrialde osoa berriz kargatzeko beharrik izan gabe.

Horrela ba, framework honek eskaintzen dituen erraztasun guztiak eta plataforma estetikoki erakargarriago egiten lagun dezakeela kontutan izanik, proiektu honen bezero aldea garatzeko berau erabili da.

## 7.2 Ingurune prestaketa

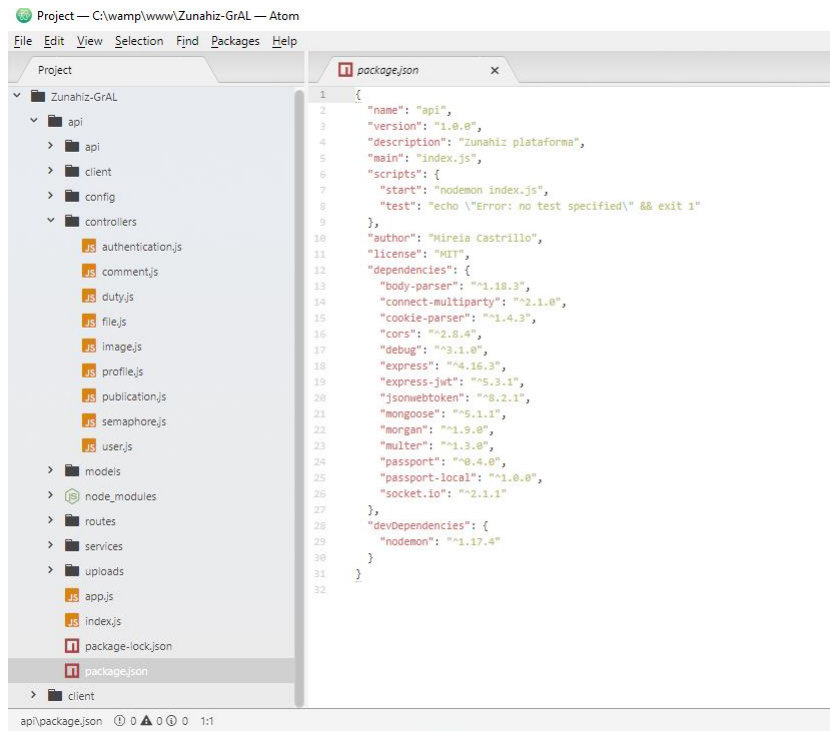
Plataforma garatzen hasi baino lehen, teknologia ezberdinen eta garapenerako beharrezkoak diren tresna guztiak ekipoa instalatu behar dira.

### 7.2.1 ATOM TESTU EDITOREA

Plataforma sortzeko beharrezkoa da testu editore bat erabiltzea, programa informatikoen iturburu-kodea sortzeko eta editatzeko.

Gaur egun merkatuan testu editore asko daude, gero eta gehiago: CoffeCup, NotePad++, Sublime Text, eta beste hainbat. Proiektu honetan software libredun *Atom* izeneko testu editorea erabili da [25].

Mahaiganeko aplikazio hau *Atom*-en web orrialde ofizialetik deskargatuz instalatu da [26].



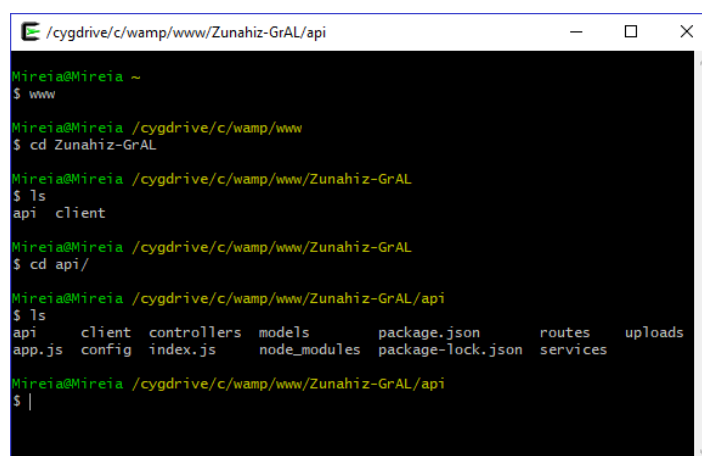
Irudia 13: ATOM testu editorea

## 7.2.2 CYGWIN

Proiektuan zehar errazago mugitzeko eta ekintzak arinagozteko asmoekin, Linux kontsola emuladore bat erabili da.

Horrela ba, *Cygwin* izeneko tresna instalatu da Windowsen Unix sistema eragilearen portaera simulatzeko.

Aplikazio hau *Cygwin*-en orrialde ofizialetik deskargatuz instalatu da [27].



Irudia 14: Cygwin terminala

### 7.2.3 NODE JS

Framework bat da zerbitzari aldean aplikazioak garatzea ahalbidetzen duena. NodeJS *back-ende*an erabiliz, JS lengoaiaren inguruko ingurumen oso bat lor daiteke. Horrela ba, hau bateragarria izango da *back-ende*an erabiltzen den *Mongo* datu basearekin eta front-end aldean erabiltzen den *Angular*ekin.

Framework hau deskargatzean eta instalatzean, honek NPM izeneko bere pakete kudeatzaile propioa ere deskargatzen du [28, 29]. Pakete kudeatzaile honen bitartez, proiektua sortzen joan ahala agertzen diren beharrezko moduluak instalatu, eguneratu edo ezabatuko dira.

### 7.2.4 NODEMON

Kodeketan oinarritzen diren proiektuetan, programa sortzen joan ahala intzidentzia eta kode arazo txikiak etengabe agertzen joan daitezke.

Etengabe agertzen diren arazo txiki hauen konponketen denbora murrizteko, *Nodemon* deitutako *Node* zerbitzariko demonio bat erabili da proiektu honen garapenean zehar. Demonio honek kodean zerbait berria egiten den bakoitzean proiektua automatikoki berrabiatzea eragiten du. Modu honetan, kodean izandako akats gehienek oharrak itzultzen dituela kontutan izanik, akatsen zuzenketan denbora asko aurrezteko lortzen da.

Horrela ba, JS sintaxia aztertzen ez duen testu editore batekin lan egin ahal izatea lortu da, testu editoreak akatsak aurkitzen ez baldin baditu ere, etengabe abiarazita dagoen demonioak helarazten dituelako akatsak eta hauen oharrak.

Demonio hau instalatzeko, *Cygwin* terminalean proiektuaren API karpetan gaudelarik (back-end atalean) honako komandoa erabili behar da:

```
/cygdrive/c/wamp/www/Zunahiz-GrAL/api
Mireia@Mireia /cygdrive/c/wamp/www/Zunahiz-GrAL/api
$ npm install -g nodemon
```

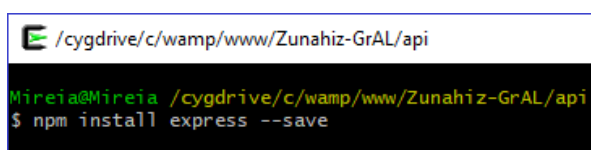
**Irudia 15: Nodemon instalatzeko komandoa**

## 7.2.5 EXPRESS JS

HTTP, ibilbide, kontroladore erabilera...hau da, API RESTaren garapena errazteko erabiltzen den *Node*-ren modulua da hau.

Egia da modulu hau gabe plataforma garatzea posible izango litzatekeela, baina honek garapen prozesu hori askoz gehiago zailduko luke.

Modulu hau instalatzeko, proiektuaren API karpetan gaudelarik berriz ere, honako komandoa erabili behar da:



```
/cygdrive/c/wamp/www/Zunahiz-GrAL/api
Mireia@Mireia /cygdrive/c/wamp/www/Zunahiz-GrAL/api
$ npm install express --save
```

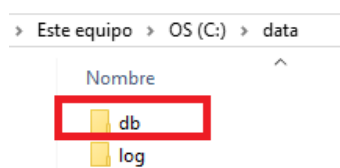
**Irudia 16: Express instalatzeko komandoa**

## 7.2.6 MONGO DB

Plataforma honetako datuak gordetzeko beharrezkoa da proiektuan datu base bat izatea. 6.3 atalean alternatiba analisisian aukeratu den bezala, proiektu honetarako datu base egokiena MongoDB dela erabaki da.

Mahaiganeke aplikazio hau *Mongo*-ren web orrialde ofizialetik deskargatuz instalatu da [30].

Hau instalatzean, karpeta bat sortu behar da datu base horien edukia gordetzeko.



**Irudia 17: Datu basearen eduki-karpeta**

Horrez gain, Mongo instalatzean, Mongo zerbitzariaren *bin* karpetaren barruan bi terminal ezberdin instalatu direla ikus daiteke: *mongo.exe* eta *mongod.exe*.

Este equipo > OS (C:) > Archivos de programa > MongoDB > Server > 3.6 > bin

Nombre	Fecha de modifica...	Tipo
bsondump	22/02/2018 19:38	Aplicación
InstallCompass	22/02/2018 19:59	Script de Window...
libeay32.dll	19/12/2016 18:30	Extensión de la apl...
mongo	22/02/2018 19:55	Aplicación
mongod	22/02/2018 20:00	Aplicación
mongod.pdb	22/02/2018 20:00	Archivo PDB

**Irudia 18: Mongo terminalak**

*mongod.exe*-ri dagokionez, hau Mongo-ren demonioa da, beti izan behar da abiarazita datu baseak erabili ahal izateko.

```

C:\Program Files\MongoDB\Server\3.6\bin\mongod.exe
2018-07-13T08:18:49.060-0700 I CONTROL [initandlisten] ** Start the se
rver with --bind_ip <address> to specify which IP
2018-07-13T08:18:49.061-0700 I CONTROL [initandlisten] ** addresses it
should serve responses from, or with --bind_ip_all to
2018-07-13T08:18:49.061-0700 I CONTROL [initandlisten] ** bind to all
interfaces. If this behavior is desired, start the
2018-07-13T08:18:49.062-0700 I CONTROL [initandlisten] ** server with
--bind_ip 127.0.0.1 to disable this warning.
2018-07-13T08:18:49.062-0700 I CONTROL [initandlisten]
2018-07-13T08:18:49.063-0700 I CONTROL [initandlisten]
2018-07-13T08:18:49.071-0700 I CONTROL [initandlisten] ** WARNING: The file sys
tem cache of this machine is configured to be greater than 40% of the total memo
ry. This can lead to increased memory pressure and poor performance.
2018-07-13T08:18:49.071-0700 I CONTROL [initandlisten] See http://dochub.mongodb
b.org/core/wt-windows-system-file-cache
2018-07-13T08:18:49.071-0700 I CONTROL [initandlisten]
2018-07-13T17:18:50.220+0200 W FTDC [initandlisten] Failed to initialize Per
formance Counters for FIDC: WindowsPdhError: PdhExpandCounterPathW failed with '
El objeto especificado no se encontró en el equipo.' for counter '\Memory\Availa
ble Bytes'
2018-07-13T17:18:50.220+0200 I FTDC [initandlisten] Initializing full-time d
iagnostic data capture with directory 'C:/data/db/diagnostic.data'
2018-07-13T17:18:50.260+0200 I NETWORK [initandlisten] waiting for connections
on port 27017

```

**Irudia 19: Mongo demonioaren terminala**

*mongo.exe* aldiz, eskaerak egiteko terminala da. Honen bitartez, datu basean dauden datuen kontsultak, datu bildumak bistaratu eta ezabatu, lerro zehatzak ezabatu eta beste hainbat funtzio egin daitezke.

```

C:\Program Files\MongoDB\Server\3.6\bin\mongo.exe
2018-06-30T09:51:38.224+0200 I CONTROL [initandlisten] ** addresses it
should serve responses from, or with --bind_ip_all to
2018-06-30T09:51:38.224+0200 I CONTROL [initandlisten] ** bind to all
interfaces. If this behavior is desired, start the
2018-06-30T09:51:38.224+0200 I CONTROL [initandlisten] ** server with
--bind_ip 127.0.0.1 to disable this warning.
2018-06-30T09:51:38.224+0200 I CONTROL [initandlisten]
2018-06-30T09:51:38.225+0200 I CONTROL [initandlisten]
2018-06-30T09:51:38.225+0200 I CONTROL [initandlisten] ** WARNING: The file sys
tem cache of this machine is configured to be greater than 40% of the total memo
ry. This can lead to increased memory pressure and poor performance.
2018-06-30T09:51:38.225+0200 I CONTROL [initandlisten] See http://dochub.mongodb
b.org/core/wt-windows-system-file-cache
2018-06-30T09:51:38.225+0200 I CONTROL [initandlisten]
> use zunahiz
switched to db zunahiz
> show collections
comments
duties
files
images
publications
semaphores
users
>

```

**Irudia 20: Mongon datu basera eskaerak egiteko terminala**

Datu basearen prestaketarekin amaitzeko, zerbitzarian *Mongoose* izeneko paketea instalatu behar da, plataformaren APItik eskaerak egin ahal izateko datu basera. Honek,



datu basean dauden datuak mapeatu egiten ditu zerbitzarian definituta dauden objektuen egituretara.

Paketea honela instalatuz:

```

E /cygdrive/c/wamp/www/Zunahiz-GrAL/api
Mireia@Mireia /cygdrive/c/wamp/www/Zunahiz-GrAL/api
$ npm install mongoose --save

```

Irudia 21: Mongoose instalatzeko komandoa

## 7.2.7 ANGULAR

*Front-end* aldeko framework hau erabiltzeko ez da aparteko aplikaziorik instalatu behar ekipoa. *Cygwin*-eko terminalean, komandoen bitartez instalatzen da hau:

```

E /cygdrive/c/wamp/www
Mireia@Mireia /cygdrive/c/wamp/www
$ npm install -g angular-cli

```

Irudia 22: Angular instalatzeko komandoa

Ondoren, *Angular* instalaturik, jadanik barneko egitura definituta izango duen bezero proiektua sortuko da honako komandoaz:

```

E /cygdrive/c/wamp/www
Mireia@Mireia /cygdrive/c/wamp/www
$ ng new client|

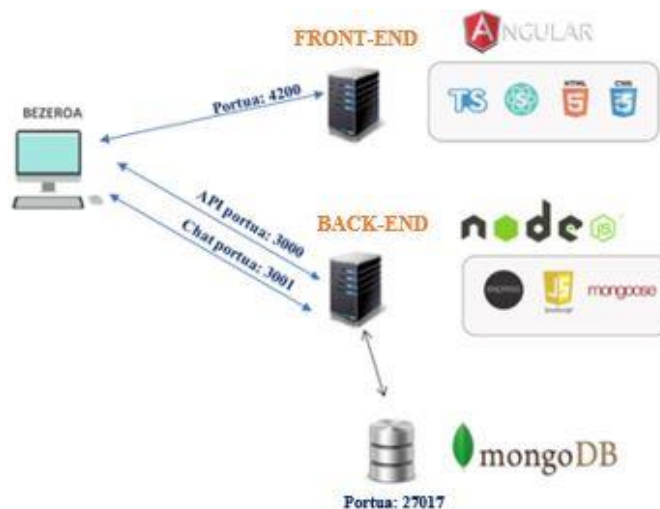
```

Irudia 23: Angular bezeroa sortzeko komandoa

Azkenik, estiloak errazten laguntzen duen *Semantic* UI framework-a eta data aukeratu ahal izateko *NgxMyDatePicker* modulua instalatu dira, bezero aldea erakargarriago bihurtzeko asmoz. Horretarako haien orrialdeko pausuak jarraituz [31, 32].

## 7.3 Programa garapena

### 7.3.1 APLIKAZIOAREN ARKITEKTURA



Irudia 24: Aplikazioaren arkitektura

Adierazitako arkitekturaren eta aurrerago azaldutako teknologien bidez, SPA (Single Page Application) bat garatu da. Hau da, ikasle batek Zunahiz plataformara jotzean orrialde bakarrean exekutatu diren Angularren modulu guztiak eskuratuko ditu. Bezeroak zerbitzariarekin behin izango du hartu emana, hasieran, ordutik aurrera soilik REST zerbitzuak eskaintzen dituen back-end zerbitzariarekin komunikatuz. Kasu honetan, front-end zein back-end zerbitzari berean egongo dira, baita datu basea ere. Web-aplikazioa orrialde bakarrean exekutatzeak, egiten diren eskaera guztiak AJAX eskaerak izatea ahalbidetzen du, horrela, web-aplikazioa oso dinamikoa eta arina izatea lortuz.

Behin lan egiteko ingurunea prestatuta egonik eta arkitektura finkaturik, plataforma garatzen hasi behar da aipatutako teknologia ezberdinen erabilera tartekatuz.

### 7.3.2 API KODEKETA

API-RESTa honako ataletan egituratuz sortu da:

- Models: Hemen datu basean gordeko diren osagaiak (*Schema* moduan ere deritzenak) eta hauek dituzten atributuak definitzen dira.

```

var mongoose=require('mongoose');
var Schema=mongoose.Schema;

var DutySchema=Schema({
  izenburua: String,
  file:{type: Schema.ObjectId,ref: 'File'},
  azalpena: String,
  urtea:String,
  hilabetea:String,
  eguna:String,
  data:String,
  noizkoUrtea:String,
  noizkoHilabetea:String,
  noizkoEguna:String,
  noizko:String,
  sortzailea: {type: Schema.ObjectId,ref: 'User'},
  semaforoak: {type: [Schema.ObjectId],ref: 'Semaphore'}
});

module.exports=mongoose.model('Duty',DutySchema);

```

#### Irudia 25: API eredu baten adibidea

- Controllers: Hemen *Schema* bakoitzak egin ditzakeen funtzioak definituko dira.

```

function deleteEginbeharra(req,res){
  var id=req.params.id;

  Duty.findByIdAndRemove(id,(err,publiUpdated)=>{
    if(err){
      res.status(500).send({message: 'Errorea eginbeharra eskuratzean'});
    }else{
      if(!publiUpdated){
        res.status(404).send({message: 'Ezin izan da eginbeharra ezabatu'});
      }else{
        res.status(200).send({publication: publiUpdated});
      }
    }
  });
}

```

#### Irudia 26: API kontroladore baten adibidea

- Routes: API-ko *Schema* bakoitzeko funtzioak erabili ahal izateko bideratze fitxategia da. Bezeroak funtzio hauek erabiltzeko erabili beharreko *path*-ak zehazten dira

```

var api=express.Router();

api.get('/eginbeharra/:id',DutyController.getEginbeharra);
api.get('/eginbeharrak',DutyController.getEginbeharrak);
api.get('/getEginbeharraWithoutFileData/:id',DutyController.getEginbeharraWithFileData);
api.get('/getEginbeharrakByMonth/:urtea&:hilabetea',DutyController.getEginbeharrakByMonth);
api.post('/eginbeharra/:id',DutyController.createEginbeharra);
api.put('/eginbeharra/:id',DutyController.updateEginbeharra);
api.delete('/eginbeharra/:id',DutyController.deleteEginbeharra);

```

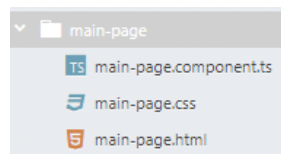
#### Irudia 27: API bideratze fitxategi baten adibidea

- Uploads: Plataformara igotzen diren fitxategiak gordetzeko karpeta.
- Horrez gain, *app.js* izeneko fitxategi bat ere dago. Horretan, datu basearekin lotura eta API-ko eta Chat-eko konexioak sortzen direlarik.

### 7.3.3 CLIENT KODEKETA

Client-a honako ataletan egituratuz sortu da:

- App: Bezero aldean erabiltzen diren zerbitzu, osagai, HTML eta CSS dokumentuak daude.
  - Orrialde bakoitzaren karpeta: Hemen, orrialde bat osatzeko erabiltzen den osagaia (*component*) eta honetan erabiliko diren CSS eta HTML fitxategiak daude.  
Hauen artean: Main-page, chat, view-user, register...



**Irudia 28: Main page orrialde karpeta eta edukia**

- Models: Bezero aldean erabiliko diren osagaiak eta hauen atributuak definitzen dira.

```
export class Comment{
  constructor(
    public _id:string,
    public content:String,
    public user:String,
    public publication:String
  ){}
}
```

**Irudia 29: Comment ereduaren egitura**

- Services: Osagai bakoitzak erabiliko dituen zerbitzuak definitzen dira hemen. Hauen bidez, API-ari eskaerak egiten zaizkio parametroak igaroz beharrezkoa den kasuetan.

```
deleteComment(id){
  return this._http.delete(this.url+'comment/'+id)
    .map(res=>res.json());
}
```

**Irudia 30: Zerbitzu adibide bat**

- *app.module* fitxategia: Modulu (osagai) bakoitza kargatzeaz eta kudeatzeaz arduratzen da fitxategi hau.
- *app.routing* fitxategia: *Front-end*-eko ibilbideak bezero aldeko osagai bakoitzarekin erlazionatzen duen fitxategia.

- Assets: *Front-end*-ean erabiltzen diren irudien biltegia (logoak, semaforoak..)

## 7.4 Probak

Plataforma garatzean joan heinean eta behin garatuta dagoelarik, honen funtzionamendu egokia baieztatu behar da proba batzuk gauzatuz.

Software erako proiektu hauetan, hiru proba mota nagusi bereizten dira: End-to-end probak, integrazio probak eta unitate probak; eta proba horiek dira proiektu honen garapenean erabilitakoak.

Atal hau oso luzea denez, eginiko probak *II. Eranskinean* atzitura datoz.

## 7.5 Emaitzak

Plataforma bukatuta egonik, honen azken bertsioa eta honen funtzionamendu egokia baieztatuz, irakaskuntza plataformaren bukaerako emaitza aztertuko da.

Atal hau oso luzea denez ere, edukia *III. Eranskinean* atzitura dator. Honetan, aplikazioaren emaitza bi ikuspuntu ezberdinetatik azaltzen da. Alde batetik, ikaslearen ikuspuntua aztertzen da eta beste aldetik, plataformaren kudeatzailearena (irakaslearena).

Hala ere, esan beharra dago, eranskin horretan erakusten den amaierako bertsioari buruz Bilboko Hezkuntza Fakultateko ikaslearekin hitz egin ostean, bere hasierako zehaztapenak bete direla ikusiz, honen onospena jaso dela.

## 8 LANAREN ANTOLAKETA

Atal honetan proiektu honetan zehar burututako faseak deskribatzen dira. Lan paketeetan (LP) banatzen delarik proiektu hau.

Lehenik eta behin, lan taldea deskribatzen da. Ondoren, LP guztien azalpena egiten da, gero Gantt-en diagrama batean irudikatuz. Eta amaitzeko, mugarriak deskribatzen dira.

### 8.1 Lan-taldea

Proiektu hau burutzeko beharrezkoa izan da hurrengo lan-taldearen elkarlana:

Izen-Abizenak	Erantzukizuna	Rola
Maiden Huarte	Senior Ingeniaria	Proiektuaren gainbegiraketa eta zuzenketa
Mireia Castrillo	Junior Ingeniaria	Proiektuaren burutzea
Mikel Malax	Irakaskuntza ikaslea	Proiektu-zehaztapenen sortzailea

Taula 4: Proiektuko partaideak

### 8.2 Lan paketeen deskribapena

#### 8.2.1 LP 1 PROIEKTUAREN KUDEAKETA ETA GAINBEGIRAKETA

LP 1-ren iraupena: 170	Hasiera data: 2018/02/02	Amaiera data: 2018/07/20
LP 1.1	Proiektuaren proposamena	
Proiektuaren zuzendariak gradu amaierako lanaren oinarriak azaldu ikasleari.		
<ul style="list-style-type: none"><li>• Hasiera data: 2018/02/02</li><li>• Amaiera data: 2018/02/03</li><li>• Giza baliabideak: Senior ingeniaria (5 ordu).</li><li>• Entregagaiak: Proiektuaren izenburua.</li></ul>		
M1: Proiektuaren hasiera. Proiektuaren proposamena ondo burutu da, proiektuaren gaia eta ideia nagusia finkatuz.		

LP 1.2	Proiektuaren kudeaketa eta jarraipena
<p>Proiektua gainbegiratzeko, bideratzeko eta akatsak zuzentzeko proiektu egilearen eta zuzendariaren arteko bilera periodikoak.</p> <ul style="list-style-type: none"> <li>• Hasiera data: 2018/02/05</li> <li>• Amaiera data: 2018/07/20</li> <li>• Baliabide teknikoak: Ordenagailu eramangarri bat (35 ordu).</li> <li>• Giza baliabideak: Senior ingeniaria (40 ordu).</li> </ul>	

## 8.2.2 LP 2 PROIEKTUAREN DISEINUA

LP 2-ren iraupena: 7	Hasiera data: 2018/02/05	Amaiera data: 2018/02/11
LP 2.1	Proiektuaren hasierako diseinua	
<p>Irakaskuntza ikasleak bere diseinu proposamenaren oinarriak zehaztu ikasleari eta ikasleak proposamena aztertu.</p> <ul style="list-style-type: none"> <li>• Hasiera data: 2018/02/05</li> <li>• Amaiera data: 2018/02/07</li> <li>• Baliabide teknikoak: Bibliografia (irakaskuntza ikasleak sortutako diseinu ebazpena)</li> <li>• Giza baliabideak: Junior ingeniaria (6 ordu), irakaskuntza ikaslea (2 ordu).</li> <li>• Entregagaiak: Proiektuaren lehen diseinua.</li> </ul>		
LP 2.2	Diseinua zehaztu	
<p>Ikasleak irakaskuntza ikasleari lehen diseinuari buruzko zalantzak galdetu, bere aldaketa edo hobekuntzak proposatu eta bien artean diseinua adostu.</p> <ul style="list-style-type: none"> <li>• Hasiera data: 2018/02/08</li> <li>• Amaiera data: 2018/02/11</li> <li>• Baliabide teknikoak: Ordenagailu bat (5 ordu).</li> <li>• Giza baliabideak: Junior ingeniaria (8 ordu), irakaskuntza ikaslea (3 ordu).</li> <li>• Entregagaiak: Proiektuaren amaierako diseinua.</li> </ul> <p>M2: Proiektua definituta. Proiektua definitu da honen diseinua zehaztuz eta bete beharko dituen ezaugarri eta funtzioak erabakiz.</p>		

### 8.2.3 LP 3 ARTEAREN EGOERAREN ANALISIA ETA IKASTAROAK

LP 3-ren iraupena: 84      Hasiera data: 2018/02/12      Amaiera data: 2018/05/06

#### LP 3.1      Artearen egoeraren analisisa

Programazio teknologiek in erlazionatutako artikuluen bilaketa eta irakurketa.

- Hasiera data: 2018/02/12
- Amaiera data: 2018/02/18
- Baliabide teknikoak: Ordenagailu eramangarri bat (35 ordu).
- Giza baliabideak: Junior ingeniaria (35 ordu).

#### LP 3.2      MEAN ikastaroa

MEAN ikastaroa ikasleak programatzeko teknologien oinarrizko erabilpena eta ezagutza bereganatzeko.

- Hasiera data: 2018/02/19
- Amaiera data: 2018/05/06
- Baliabide teknikoak: Ordenagailu bat (320 ordu).
- Giza baliabideak: Junior ingeniaria (320 ordu).

M3: Formakuntza burututa. Artearen egoera aztertutakoan erabaki diren teknologia eta programa ezberdinei buruzko ezaguerak barneratu dira.

### 8.2.4 LP 4 WEBGUNEAREN GARAPENA

LP 4-ren iraupena: 56      Hasiera data: 2018/04/30      Amaiera data: 2018/06/14

#### LP 4.1      Zerbitzari aldearen kodetzea

Webgunearen funtzionamendu eta metodoez arduratuko den zerbitzaria sortu.

- Hasiera data: 2018/04/30
- Amaiera data: 2018/05/20
- Baliabide teknikoak: Ordenagailu bat (180 ordu).
- Giza baliabideak: Junior ingeniaria (185 ordu)
- Entregagaiak: Webgunearen zerbitzaria.

#### LP 4.2      Bezero aldearen kodetzea

Webgunearen alderdi grafikoaz arduratuko den bezero aldea sortu.

- Hasiera data: 2018/05/21
- Amaiera data: 2018/06/14



- Baliabide teknikoak: Ordenagailu bat ( 252 ordu)
- Giza baliabideak: Junior ingeniari bat (252 ordu).
- Entregagaiak: Webgunearen bezero aldea.

#### LP 4.3

#### Metodoen konprobaketa eta integrazioa

Bezero eta zerbitzari aldean sorturiko metodo eta funtzioek eginbeharrekoak bete eta modu egokian lan egiten dutela, eta alde bien arteko integrazioa modu egokian dabilela baieztatu.

- Hasiera data: 2018/04/30
- Amaiera data: 2018/06/14
- Baliabide teknikoak: Ordenagailu bat ( 56 ordu)
- Giza baliabideak: Junior ingeniari bat (56 ordu).

M4: Webgunea garatuta. Bezero eta zerbitzari aldeak garatu, eta bien integrazioak modu egokian lan egitea lortu.

### 8.2.5 LP 5 EMAITZA AZTERTU

LP 5-ren iraupena: 7

Hasiera data: 2018/06/11

Amaiera data: 2018/06/17

#### LP 5.1

#### Webgune funtzionamendu konprobaketa

Webguneak osotasunean duen funtzionamendua aztertu, ondo dagoela baieztatu eta akats txikiak konpondu.

- Hasiera data: 2018/06/11
- Amaiera data: 2018/06/14
- Baliabide teknikoak: Ordenagailu bat (24 ordu).
- Giza baliabideak: Junior ingeniaria (24 ordu).

#### LP 5.2

#### Diseinatzailearen onspena

Diseinatzailearekin bat, garatutako webgunea berak hasieran proposatutakoarekin bat datorrela eta emaitza bere gustukoa dela baieztatu.

- Hasiera data: 2018/06/15
- Amaiera data: 2018/06/17
- Baliabide teknikoak: Ordenagailu (3 ordu)
- Giza baliabideak: Junior ingeniaria (6 ordu) eta irakaskuntza ikaslea (6 ordu).

M5: Webgunea amaituta. Diseinatzailearekin webguneak eman behar dituen emaitzak lortzen direla ikusi eta amaitutzat eman.

## 8.2.6 LP 6 EMAITZA HELARAZI

LP 6-ren iraupena: 179      Hasiera data: 2018/06/18      Amaiera data: 2018/06/24

### LP 6.1      Demo grabaketa

Webgunearen demo bat grabatu irakaskuntzako ikasleak bere GrAL defentsan erakutsi ahal izateko. Demo horretan, irakaskuntza ikasleak eskatutako funtzionamendu garrantzitsuenak erakusten direlarik.

- Hasiera data: 2018/06/18
- Amaiera data: 2018/06/20
- Baliabide teknikoak: Ordenagailu bat (15 ordu).
- Giza baliabideak: Junior ingeniaria (15 ordu) eta irakaskuntza ikaslea (3 ordu).
- Entregagaiak: Webgune demoa

### LP 6.2      Webgune azalpena

Irakaskuntza ikaslearen webgunearen funtzionamendua azaldu honek, bideoarekin batera, defentsan azalpenak eman ditzen.

- Hasiera data: 2018/06/21
- Amaiera data: 2018/06/24
- Baliabide teknikoak: Ordenagailu bat (12 ordu).
- Giza baliabideak: Junior ingeniaria (12 ordu).

M6: Emaita helarazita. Azkeneko emaitza, webgunea, irakaskuntza ikasleari bidali eta azalduta.

## 8.2.7 LP 7 DOKUMENTAZIOA

LP 7-ren iraupena: 30      Hasiera data: 2018/06/25      Amaiera data: 2018/07/20

### LP 7.1      Lana idatzi

Proiektuaren inguruko dokumentazioa idatzi da.

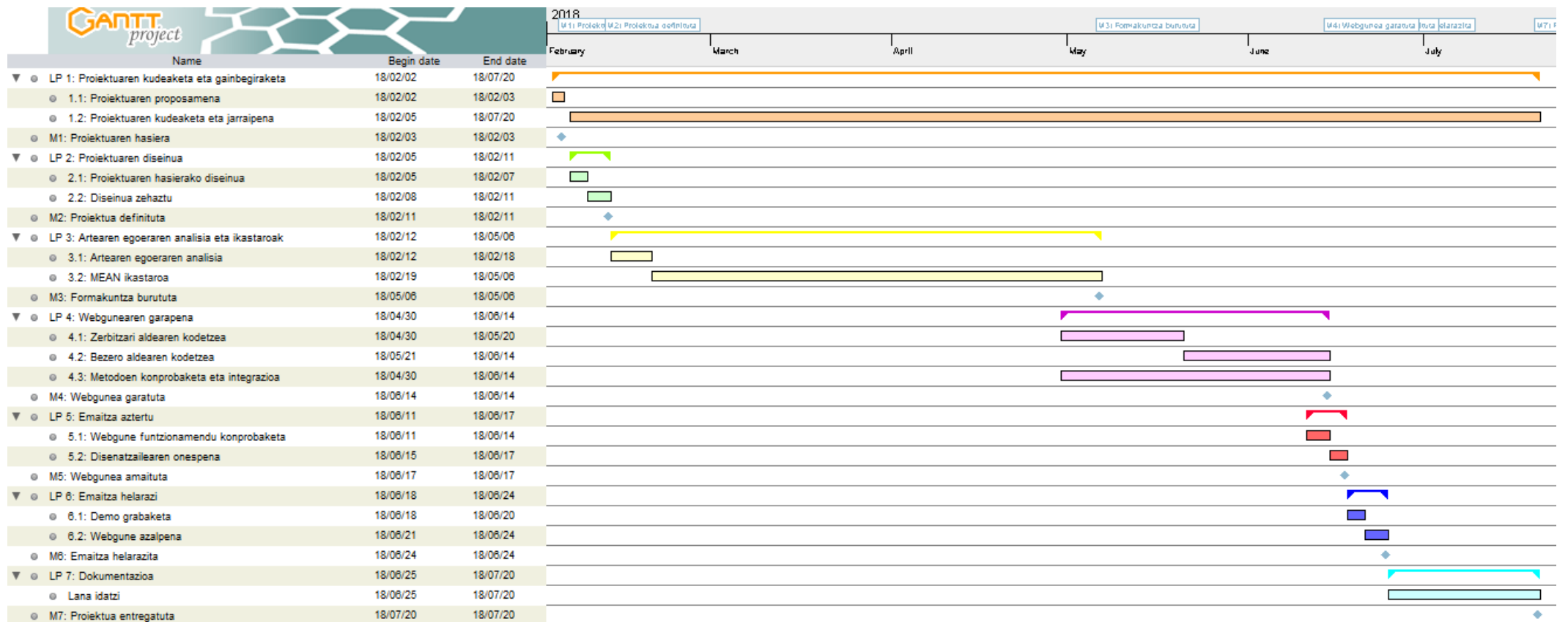
- Hasiera data: 2018/06/25
- Amaiera data: 2018/07/20
- Baliabide teknikoak: Ordenagailu bat (130 ordu).
- Giza baliabideak: Junior ingeniari bat (130 ordu).
- Entregagaiak: Memoria.

M7: Proiektua entregatuta. Dokumentazio guztia idatzi da eta epe barruan entregatu da.

## 8.3 Mugarriak

Mugarria	Deskribapena	Data
M1	Proiektuaren hasiera	2018/02/03
M2	Proiektua definituta	2018/02/11
M3	Formakuntza burututa	2018/05/06
M4	Webgunea garatuta	2018/06/14
M5	Webgunea amaituta	2018/06/17
M6	Emitza helarazita	2018/06/24
M7	Proiektua entregatuta	2018/07/20

Taula 5: Mugarriak



Irudia 31: Gantt diagrama

## 9. GASTU AITORPENA

Atal honetan proiektuan zehar izandako kostu totala azaltzen da. Kostu hau lau motetan bereiziko da: Barne orduak, amortizazioak, azpikontratazioak eta gastuak.

### 9.1 Kostuak

#### 9.1.1 BARNE ORDUAK

Atal honetan proiektuko partaide bakoitzaren lan-zama ordu kopuruan eta horren kostuak azaltzen dira, ordu kopurua, lan paketeetan erabilitakoa izanik.

Partaideia	Kostua (€/orduko)	Ordu kopurua	Partaidearen kostua (€)
Senior ingeniaria	60	45	2.700
Junior ingeniaria	30	1.049	31.470
Irakaskuntza ikaslea	25	14	350
Barne orduen kostu totala:			34.520

Taula 6: Barne orduen kostua

#### 9.1.2 AMORTIZAZIOAK

Ataza honetan proiektua garatzeko beharrezkoak izan diren baliabide naturalei dagozkien kostuak azaltzen dira.

Baliabidea	Kostua (€)	Bizitza erabilgarria (ordu)	Ordu kopurua	Baliabidearen kostua (€)
Senior ingeniariaren ordenagailua	900	5.000	45	8,1
Junior ingeniariaren ordenagailua	600	5.000	1.029	123,5
Amortizazioen kostu totala:				131,6

Taula 7: Amortizazioen kostua

### 9.1.3 AZPIKONTRATAZIOAK

Proiektu honetan ez da azpikontrataziorik egin, beraz, ez da horren eraginik egon proiektu honen kostuan.

### 9.1.4 GASTUAK

Atal honetan berrerabili ezin izango diren baliabideak eta hauen kostua izango da kontuan. Besteak beste, UdeMy plataforman eginiko ikastaroak, bulego materiala (papera, argindarra...) eta proiektu honen garapenerako erositako pendrive bat [33].

Baliabidea	Kostua (€)
Ikastaroak	20
Bulego gastuak	60
Pendrivea	12
<b>Guztira</b>	<b>92</b>

Taula 8: Gastuen kostua

## 9.2 Kostu aitortpena orotara

Ataza	Kostua (€)
Barne orduak	34.520
Amortizazioak	131,6
Azpikontratazioak	0
Gastuak	92
<b>Guztira</b>	<b>34.743,6</b>

Taula 9: Guztizko kostu aitortpena

## 10 ARRISKU ANALISIA

Proiektu honetan zehar hainbat arrisku egon dira proiektuaren atzerapena edo bertan bera uztea ekarri ahal izan dutenak. Proiektuarekin hastean arrisku posibleen analisia egin zen eta behin amaituta dagoela, arriskuak ekidin direla esan daiteke.

Arrisku bakoitzean bi faktore izan dira kontuan, alde batetik arriskua agertzeko probabilitatea eta bestetik arriskuaren eragina. Jarraian proiektuaren hasieran egin zen arrisku bakoitzaren analisia azaltzen da.

### 10.1 Proiektuko kideren baten baja (A1)

Proiektu honetan hainbat kidek hartu dute parte eta proiektuan zehar baten batek arazoak izan eta proiektua uzteko arriskua egon da. Edozein kideren bajak eragin handia izango luke proiektuan, proiektuaren atalen bat blokeatuta geldituko bailitzateke, proiektu osoaren atzerapena eraginez.

Arrisku honi erantzuna ematea ez da erraza, aurreikusi ezin den arazo bat delako. Kasu honetan, kontingentzia posible bezala proiektuko kideei arazoren bat izanez gero ahalik eta azkarren esateko eskatu zaie, horrela ordezkoren bat bilatzeko aukera egon dadin.

### 10.2 Programazio lengoaiarekin arazoak (A2)

Proiektu honetan, programatzeko JavaScript erabili den arren, proiektuaren hasieran lengoia ezberdinak aztertu ziren eta lengoia okerra aukeratu izana gerta liteke, hori oso arriskutsua izango litzatekeelarik. Programazio lengoia bakoitzak aukera ezberdinak ematen ditu eta oker bat aukeratzeak hasieran ezarritako ezaugarri batzuk ezin ezartzea ekar dezakeelako.

Prebentzio neurri modura, programatzen hasi aurretik programazio lengoia ezberdinak ondo aztertu dira, bakoitzak dituen ezaugarriak aztertuz eta azkenean JavaScript aukeratuz.

### 10.3 Aurrekontuarekin arazoak (A3)

Edozein proiektutan bezala, proiektu honetan posible da aurreikusi gabeko gastu batzuk agertzea.

Horregatik, proiektuaren hasieran ahalik eta aurrekontu zehatzena egin da, gastu ezberdinak kontuan izanik. Hartara, proiektu hau programazio proiektu bat denez, gastu gehienak nahiko erraz aurreikusi ahal izan dira.

### 10.4 Datu galera (A4)

Proiektu honetan burututako lanaren zati handi bat ordenagailuz egin da, horregatik faktore ezberdinek (argindarra joan delako, ordenagailua apurtu delako etab.) burututako lan guztia galtzea ekar dezakete.

Arrisku hau ematea ez da oso posiblea baina honek eragin handiak izango lituzke, horregatik prebentzio bezala, egindako lanaren segurtasun kopiak egin dira periodikoki eta amaierako bertsioa Github plataformara igo da [34].

### 10.5 Ezaugarri okerrak (A5)

Proiektu hau irakaskuntzako ikasle batekin elkarlanean burutu da, berak proposatutako ikasleentzako web orri bat sortzea izan baita honen helburua. Horregatik, web aplikazio hori oker diseinatzeke arriskua egon da, gaizki ulerturen baten edo informazio faltaren ondorioz.

Arazo posible hori ekiditeko, proiektuaren hasieran ikaslearekin bildu eta ezaugarriak ahalik eta zehatzen ezarri ziren eta ondoren proiektua garatzerakoan fase bakoitzean ikasleari erakutsi zaio nahi den bezala doan edo ez jakiteko.

### 10.6 Erabiltzaileen datuen galera (A6)

Internet bidez eskuragarri dagoen edozein web aplikaziotan bezala, honetan ere posible da hacker batek komunikazioa pintxatu eta datu basea erasotzea, erabiltzaile datuak eta zerbitzarian biltegitratzen direnak galaraziz.



Kasu honetan, proiektua ikasgela baten barnean probatzeko prototipo bat baino ez denez hau gertatzeko aukerarik ez da kontutan izango. Baina azaldu bezala, proiektu honek etorkizunera begira izan dezakeen erabilpena kontutan izan behar da, plataforma hau errealitatean erabiltzeko helburuaz eginda dagoenez, arrisku hau gertatzeko aukera sor daiteke etorkizun batean.

Horrela ba, benetako inplementazioa ematen bada, arazo posible hori ez gertatzeko kontingentzia plan modura *HTTPS* (Hypertext Transport Protocol Secure) protokoloa erabiltzea izango litzateke honen ebazpidea.

## 10.7 Arriskuen analisisien laburpena

Proiektu honetan eman daitezkeen arriskuak aztertuz, hauek identifikatu, gertatzeko aukera eta gertatzekotan izango luketen eragina ebaluatzeko, probabilitate-eragin matrizea osatu da:

		ERAGINA				
		Oso txikia	Txikia	Ertaina	Handia	Itzela
PROBABILITATEA	Itzela					
	Handia					
	Ertaina				A2	A4
	Txikia		A5		A1	A6
	Oso txikia		A3			

Taula 10: Probabilitate-eragin matrizea

## 11 ONDORIOAK

Proiektu honetan irakaskuntza inguruan murgildutako nerabeei sare sozialen erabilera egokiaren beharraz jabetzeko eta norbanako baten ekintzek eta hitzek izan ditzaketen eraginak irakasteko asmoa duen irakaskuntza plataforma bat sortu da.

Sorkuntza hau, Telekomunikazio Teknologiaren Ingeniaritzako Graduan zehar ikasi ez diren eta gaur egun web aplikazioen garapenean erabiltzen diren teknologi berriez baliatuz egin da. Gaur egunean erabilienak diren teknologiak aztertuz eta proiektu honetara gehien moldatu litekeena aukeratuz, teknologia berri horiek eskaintzen zituzten funtzionalitate eta tresnak lehendabizi erabiltzen ikasi eta gero proiektuaren sorkuntzarako erabili direlarik.

Halaber, plataformak erabiltzaile-kontuak kudeatu ditu autentikatzea burutzeko eta sesioa mantentzeko ohikoena ez den modu baten bidez, ebazpen modura token-ak erabiliz eta gainera, aplikazio honen proba-diseinua burutu da plataformaren kasu espezifiko batzuk prestatuz.

Emaitzari dagokionez, sortu den irakaskuntza plataforma honen funtzionamendua probatu da eta modu egokian lan egiten duela ondorioztatu da. Programazio aldeko itxaropenak betetzeaz gain, zehaztapenak eta proiektuaren ideia nagusia proposatu zituen haren onespina ere jaso delarik. Proiektuaren diseinua Bilboko Hezkuntza Fakultateko ikasle batek emandako zehaztapenenetara egokitu, eta ezaugarri horiek betetzen dituen zerbitzari eta bezeroak sortu baitira.

Amaitzeko, proiektu honek etorkizunera begira izan dezakeen erabilpena kontutan izan behar da. Plataforma hau errealitatean erabiltzeko helburuaz neurrira egindako lana da eta irakaskuntzako gela batean benetako irakasleekin probak egiteko asmoa dago.

## 12. ERREFERENTZIAK

- [1] Díde ikerketa taldea, <<El uso de redes sociales en niños y adolescentes>>. Erabilgarri: <http://educaryaprender.es/datos-uso-de-redes-sociales-ninos-y-adolescentes/>
- [2] Euroresidentes, <<Estadísticas sobre el uso de Internet>> Erabilgarri: <https://www.euroresidentes.com/tecnologia/noticias-internet/estadisticas-sobre-el-uso-de-internet>
- [3] Marc Prensky, <<Digital Natives, Digital Immigrants>. Erabilgarri: <https://edorigami.wikispaces.com/file/view/PRENSKY+-+DIGITAL+NATIVES+AND+IMMIGRANTS+1.PDF>
- [4] Blackboard Academic Suite: <https://bb.mediaec.uaic.ro>
- [5] Dokeos: <https://www.dokeos.com>
- [6] Sakai: <https://www.sakaiproject.org>
- [7] Moodle: <https://moodle.org>
- [8] Universidad Internacional de Valencia, <<Características, tipos y plataformas más utilizadas para estudiar a distancia>>. Erabilgarri: <https://www.universidadviu.es/caracteristicas-tipos-y-plataformas-mas-utilizadas-para-estudiar-a-distancia/>
- [9] Wikipedia, Moodle: <https://eu.wikipedia.org/wiki/Moodle>
- [10] Arume informáticos, <<Página web, aplicación web y aplicación de escritorio>>. Erabilgarri: <http://www.arumeinformatica.es/dudas/pagina-web-aplicacion-web-y-aplicacion-de-escritorio-cual-es-la-diferencia>
- [11] Scriptcase: <https://www.scriptcase.net/es/>
- [12] Scriptcase blog, <<Aplicaciones Web vs Websites>>. Erabilgarri: <https://www.scriptcaseblog.net/es/aplicaciones-web-webapp-vs-websites-entienda-la-diferencia/>
- [13] Wikipedia, Scriptcase: <https://en.wikipedia.org/wiki/Scriptcase>
- [14] Wikipedia, XAMP: [https://en.wikipedia.org/wiki/List\\_of\\_Apache%E2%80%93MySQL%E2%80%93PHP\\_packages](https://en.wikipedia.org/wiki/List_of_Apache%E2%80%93MySQL%E2%80%93PHP_packages)
- [15] Diseño Web Madrid, << Diseño web desde cero: ¿Qué ventajas tiene frente a las plantillas?>>. Erabilgarri: <https://webtematica.com/disenio-web-desde-cero-que-ventajas-tiene-frente-las-plantillas>
- [16] Difrente, <<Diferencia entre sitio web y la aplicación web>>. Erabilgarri: <https://difentre.com/diferencia-entre-el-sitio-web-y-la-aplicacion-web/>
- [17] Eaprende, Scriptcase: <https://www.eaprende.com/blog/2017/05/31/scriptcase-2/>
- [18] Wikipedia, MySQL: <https://es.wikipedia.org/wiki/MySQL>

- [19] Wikipedia, Base de datos relacional: [https://es.wikipedia.org/wiki/Base\\_de\\_datos\\_relacional](https://es.wikipedia.org/wiki/Base_de_datos_relacional)
- [20] Wikipedia, MongoDB: <https://es.wikipedia.org/wiki/MongoDB>
- [21] Wikipedia, NoSQL: <https://es.wikipedia.org/wiki/NoSQL#Ventajas>
- [22] Cecilio Alvarez, << ¿Cómo funciona Node.js?>>.  
Erabilgarri: <https://www.genbetadev.com/frameworks/como-funciona-node-js>
- [23] José Manuel Alarcón, <<Qué es el stack MEAN y cómo escoger el mejor para ti>>.  
Erabilgarri: <https://www.campusmvp.es/recursos/post/Que-es-el-stack-MEAN-y-como-escoger-el-mejor-para-ti.aspx>
- [24] Wikipedia, JQuery: <https://es.wikipedia.org/wiki/JQuery>
- [25] Wikipedia, Atom: [https://es.wikipedia.org/wiki/Atom\\_\(software\)](https://es.wikipedia.org/wiki/Atom_(software))
- [26] Atom: <https://atom.io/>
- [27] Cygwin: <https://cygwin.com/install.html>
- [28] Node Js: <https://nodejs.org/es/>
- [29] Wikipedia, Npm: <https://es.wikipedia.org/wiki/Npm>
- [30] MongoDB: <https://www.mongodb.com/download-center>
- [31] SemanticUI: <https://semantic-ui.com/introduction/getting-started.html>
- [32] MyDatePicker: <https://github.com/kekeh/ngx-mydatepicker>
- [33] Udemy, ikastaro plataforma: <https://www.udemy.com>
- [34] Proiektuaren errepositorioa: <https://github.com/mcastrillo003/Zunahiz-GrAL>

---

# ERANSKINAK

---

- I. Eranskina: Egitura
- II. Eranskina: Probak
- III. Eranskina: Emaitzak
- IV. Eranskina: Aplikazioa

# I. ERANSKINA: EGITURA

Esan bezala, lortu behar den emaitza kontutan izanik, emaitza hori lortuko duen egitura sortu behar da. Horretarako, plataformak bete beharreko funtzionalitateak argi izanik, hauek betetzeko beharrezkoak diren ereduak eta hauen metodoak definitu behar dira.

Horretarako, egitura definitzeko atal hau bitan banatzen da. Alde batetik, zerbitzariako egitura eta bestetik, bezero aldeko egitura.

## API egitura

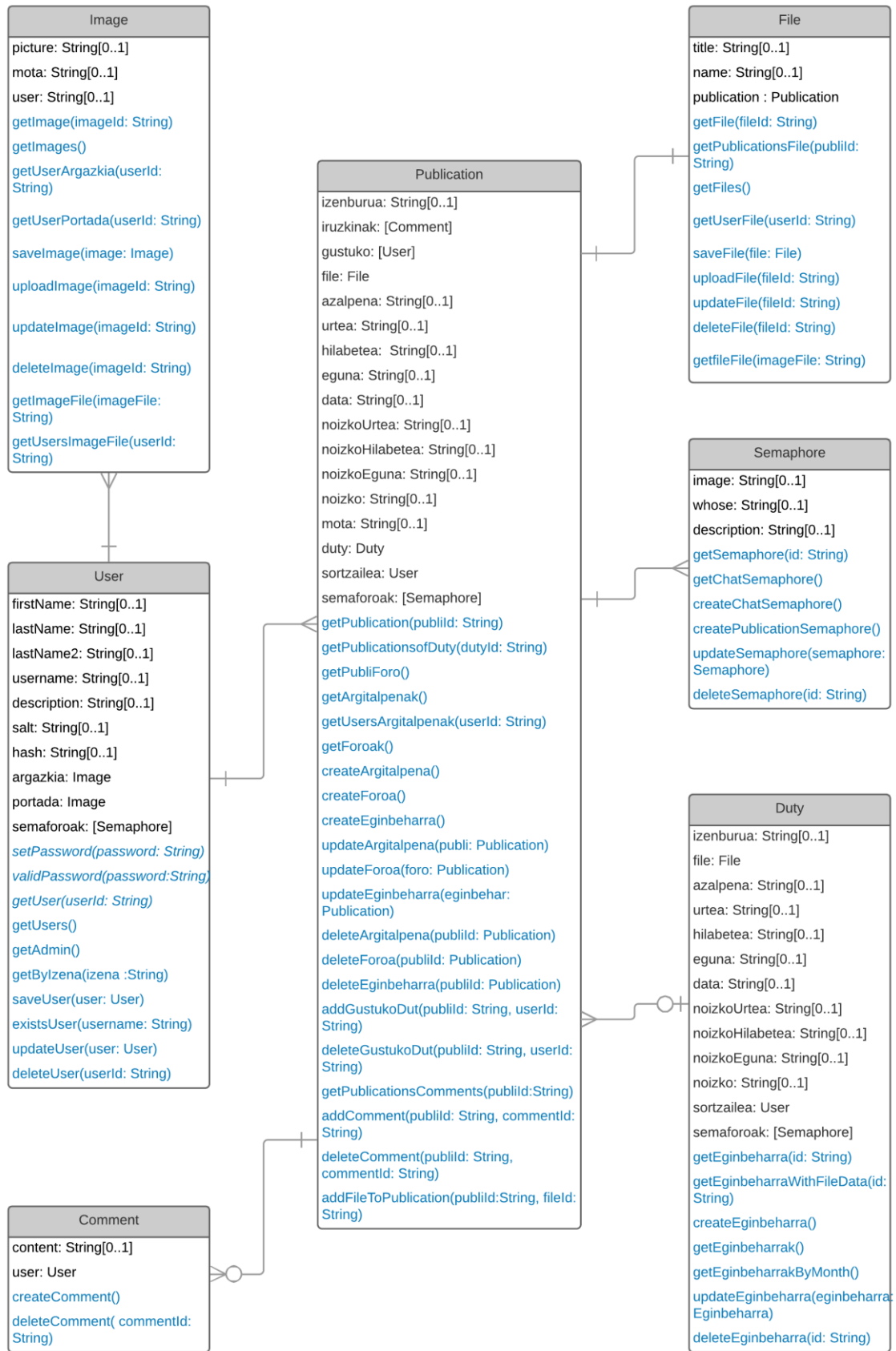
Zerbitzari aldeko egitura honetan, datu basearekin hartu-emanen ibiliko diren ereduak definitzen dira.

32.irudian ikus daitekeen moduan, beharrezkoak diren ereduak baino ez dira definitu. Nahiz eta batzuk antzekoak gerta daitezkeen, atributu edota metodo ezberdinak dituztela ikus daiteke, horrek bitan ezberdintzea eragiten duelarik.

## Client egitura

Bezero aldeko egitura honetan, datu basearekin hartu-emanen ibiliko diren ereduak definitzeaz gain, datu basean gorde ez arren bezero aldean lan egiteko beharrezkoak diren bestelako eredu batzuk ere definitu behar dira.

33.irudian ikus daitekeen moduan, kasu honetan zerbitzari aldeak erabiltzen dituen ereduez gain beste bi eredu berri daude. Alde batetik, erabiltzailearen sesioko informazioa gordetzeko balio duen *Credential* eredu mota eta beste aldetik, *Chat* zerbitzuan mezuak elkarbanatzea ahalbidetuko duen *Mezua* eredu mota.



Irudia 32: API egituraren klaseen diagrama



Irudia 33: CLIENT egituraren klaseen diagrama



## II. ERANSKINA: PROBAK

Software garapena modu egokian egin dela egiaztatzea eta proben fase bat burutzea premiazkoa da software proiektu batean. Proben bitartez diseinua garatzen ari denean, sortzen dena fidagarria den edo ez eta egindako lanak zehaztutako espezifikazioak beten dituen edo ez frogatzen da.

Gainera, proba on batek garapen fasean sortzen diren akatsak laster identifikatzen laguntzen du. Honen ondorioz, hauteman diren akats hauek proiektuko lehen faseetan konpondu ahalko dira eta ondorengoetara hedatzea ekidingo da, hauek konpontzeko behar den denbora eta kostua murriztuz.

### Proba motak

Software erako proiektu hauetan, hiru proba mota nagusi bereizten dira: End-to-end probak, integrazio probak eta unitate probak.

Proiektu honen garapenean erabilitako teknologiek, proiektuaren funtzionalitate ezberdinak probatzeko erabili behar izan diren framework-ak baldintzatu dituzte. Izan ere, proba ezberdinak burutzeko, produkzioan erabilitako lengoia eta tresnekin eta proiektuaren testuinguruarekin bat datozen testatzaileak erabili behar izan baitira.

Azaldutako hiru proba mota hauek piramide modura antolatzen dira:



Irudia 34: Zunahiz plataformako proben planifikazioa

Piramidea aztertuz, proben oinarrian unitate probak daudela ikus daiteke. Hortaz, proba mota hauek proben kopuru handiena osatu beharko luketela ondoriozta daiteke. Integrazio proben kopurua txikiago bat izanik eta azkenik, end-to-end probak egonik.

## **UNITATE PROBAK**

Proba mota honetan, web aplikazioaren modulu edo osagai bat modu isolatu batean frogatzen da bestelako baliabiderik erabili gabe (datu base gabe, nabigatzaile gabe...). Kasu honetan, beste modulu edo zerbitzuen eraginak simulatu egiten dira.

Proba hauek oso bizkor prestatzen diren probak dira eta garatzen doazen funtzionalitateek egin behar dutena egiten dutela egiaztatzeko oso aproposak dira. Hala ere, modu isolatu batean kudeatzen direnez, proba hauen bitartez ezin da egiaztatu aplikazioko beste osagaiekin bateratzean guztiek ondo lan egingo dutenik. Hau dela eta, proba hauek fidagarritasun gutxien eskaintzen dutenak direla esan daiteke.

## **INTEGRAZIO PROBAK**

Proba hauetan jadanik moduluak ez dira modu isolatuan frogatzen, testuinguru jakin batean baizik. Hau dela eta, proba mota hauekin, unitate-probekin ez bezala, modulu ezberdinak elkarren artean batak bestearengan duten eragina ikus dezakegu. Proba mota hauen automatizazioak programatze denbora gehiago eta zailtasun handiagoa eskatzen dute, baina unitate-probek baino konfiantza handiagoa eskaintzen digute.

## **END-TO-END PROBAK**

Proba mota honetan erabiltzaile batek izango lukeen portaera simulatuz, aplikazioa osotasunean frogatzen da. Proba hauek, oro har, ez dira automatizatzen eta oso geldoak izaten dira. Gainera, oso ahulak dira, programan eginiko aldaketa txiki bakoitzak testa apurtu dezakeelako. Bestetik, fidagarritasun gehien eskaintzen dizkiguten probak dira hauek, besteetan ez bezala, honako probetan ez dugulako ezer simulatzen eta emaitzak argi ikus daitezkeelako.

## Proben garapena

Atal honetan, orain bertan aipatutako proba ezberdinak plataformaren sorreran zehar nola garatu diren ikusiko da.

## UNITATE PROBAK

Proiektu honen garapena Angular erabiliz egin denez, unitate probak Angular-ekin eta JavaScript lengoaiarekin bateragarriak diren Jasmine eta Karma tresnekin egin dira.



**Irudia 35: Jasmine eta Karma tresnen logotipoak**

*Jasmineri* dagokionez, test framework honek kodean idatzitako modulu ezberdinak modu isolatu batean frogatzeko aukera ematen du.

*Karmari* dagokionenez, honek Jasminerekin egindako proba ezberdinak nabigatzailean modu lagungarri batean ikusteko aukera eskaintzen du.

Proiektu honetan unitate probak plataformaren garapean gertatu diren 6 funtzio zailenak ondo sortu eta lan egiten duten egiaztatzeko erabili dira.

### 1. Ordua

Test hau argitalpen bat sortzerakoan honen data atributua, hau da, sorrera eguna eta ordua adierazten duena ondo sortzen dela frogatzeko erabili da. Proiektuaren garapenean zehar, momentuko urtea, hilabetea eta eguna alde batetik, eta bestetik ordua eta minutuak lortzen dituzten funtzioak jadanik existitzen direla ikusi da baina hauek biek independenteki ez dute proiektuaren data atributuak beharreko egitura sortzen.

Arazo honi aurre egiteko, funtzio horiek erabiliz data atributuari beharrezko egitura emateko funtzioa sortu eta honek bere eginkizuna ondo betetzen duela frogatu behar izan da.

```

describe('ordua', () => {
  it('ordua kate modura', () => {
    const result = data();
    expect(result).toContain('2018');
    expect(result).toContain('07');
    expect(result).toContain('09');
    expect(result).toContain('13');
    expect(result).toContain('06');
  })
})

```

### Irudia 36: Momentuko data sortzeko testaren kodea

Kasu honetan, ez da akatsik agertu proba egitean eta hortaz, osagaia hasieratik ondo kodetu dela ondorioztatzen da.

## 2. Izena bihurtu

Honako testa erabiltzaileen izen-abizenak modu erakargarri baten bistaratu ahal izateko, diseinuan proposaturiko moduan lortzen direla egiaztatzen da. Hau da, datu basean konparaketak errazteko letra xehez dauden izen eta abizen atributuak bezeroari lehenengo hizkia letra larriz bueltatzea lortu behar baita.

Hori lortzeko, datu basetik lortzen diren izen-abizenen lehenengo hizkia letra larrietara igaro eta gero bidaliko zaizkio bueltan erabiltzaileari honi bistaratu ditzen.

Funtzio hau frogatzeko hainbat kasu ezberdin planteatu dira eta eman beharko luketen emaitza itzultzen ote duten edo ez ikustea da. Hau da, behin osagaia programatuta, kasu ezberdinak planteatzen dira eta bakoitzean eman beharko lukeen emaitza zehazten da.

Kasu honetan, hauek dira kontutan izan beharreko kasuak:

1. Izena eta abizen biak letra xehez
2. Izena bakarrik letra xehez
3. 1. abizena bakarrik letra xehez
4. 2. abizena bakarrik letra xehez
5. Izena eta 1. abizena letra xehez
6. Izena eta 2. abizena letra xehez

```

describe('izena bihurtu', () => {
  it('izen abizenak hizki larrietara pasa: guztiak', () => {
    const result = name('mireia', 'castrillo', 'gonzalez');
    expect(result).toBe('Mireia Castrillo Gonzalez');
  })

  it('izen abizenak hizki larrietara pasa: izena', () => {
    const result = name('mireia', 'Castrillo', 'Gonzalez');
    expect(result).toBe('Mireia Castrillo Gonzalez');
  })

  it('izen abizenak hizki larrietara pasa: abizena1', () => {
    const result = name('Mireia', 'castrillo', 'Gonzalez');
    expect(result).toBe('Mireia Castrillo Gonzalez');
  })

  it('izen abizenak hizki larrietara pasa: abizena2', () => {
    const result = name('mireia', 'Castrillo', 'gonzalez');
    expect(result).toBe('Mireia Castrillo Gonzalez');
  })

  it('izen abizenak hizki larrietara pasa: izena&abizena1', () => {
    const result = name('mireia', 'castrillo', 'Gonzalez');
    expect(result).toBe('Mireia Castrillo Gonzalez');
  })

  it('izen abizenak hizki larrietara pasa: izena&abizena2', () => {
    const result = name('mireia', 'Castrillo', 'gonzalez');
    expect(result).toBe('Mireia Castrillo Gonzalez');
  })
})

```

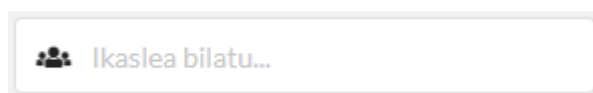
**Irudia 37: Izen-abizenen bihurketarako testaren kodea**

37.irudi honetan funtzio honen proba ezberdinak ikus daitezke adibide gisa.

Test honen bidez, izen-abizenak bezeroan behar bezala bistaratzeko moduan ondo lortzen direla egiaztatu da.

### 3. Izena bihurtu

Honako testa bezero aldeko pertsona-bilatzailearen funtzionamendu egokia bermatzeko egin da. Hau da, erabiltzaileak bilatzaile bat erabiltzean norbanako baten izena edota abizenak letra larriz edo xehez sar ditzake eta ondorioz, bilaketan sarturikoa eta datu baseko edukia bat ez etortzea gerta daiteke.



**Irudia 38: Plataformako pertsona-bilatzailea**

Hori ekiditeko, datu basean izen-abizenak osorik letra xehez gordetzeaz gain, bezerotik bilaketa-eskaeran datozen atributuak letra xehetara igaroko dira eta hutsuneak deuseztatuko dira. Horrela, izenaren atzean eta abizenaren aurretik hutsune

bakarra jarri ezean bestelako kopuru bat jarritz ere, izena eta abizena atributuak ondo bereiztea lortu behar baita.

Kasu honetan, osagai hau ondo egin dela konprobatzeko erabiltzaileak bilatzailean sarturiko balioak kontutan izanik 9 kasu ezberdin aztertu dira:

1. Izena soilik sartu da bilatzailean
2. Izena + hutsune bat
3. Izena + hutsune bat baino gehiago
4. Izena eta 1. abizena
5. Izena + 1. abizena + hutsune bat
6. Izena + 1. abizena + hutsune bat baino gehiago
7. Izena + 1. abizena + 2. abizena
8. Izena + 1. abizena + 2. abizena + hutsune bat
9. Izena + 1. abizena + 2. abizena + hutsune bat

Proba hauetan erabilitako kodea honako da:

```
describe('izen abizenak lortu', () => {
  it('izena soilik', () => {
    const result = filterName('mireia');
    expect(result).toBe('mireia');
  })

  it('izena + hutsune', () => {
    const result = filterName('mireia ');
    expect(result).toBe('mireia');
  })

  it('izena + hutsune bat baino gehiago', () => {
    const result = filterName('mireia  ');
    expect(result).toBe('mireia');
  })

  it('izena + abizena1', () => {
    const result = filterName('mireia castrillo');
    expect(result).toBe('mireia castrillo');
  })

  it('izena + abizena1 + hutsune', () => {
    const result = filterName('mireia castrillo ');
    expect(result).toBe('mireia castrillo');
  })

  it('izena + abizena1 + hutsune bat baino gehiago', () => {
    const result = filterName('mireia castrillo  ');
    expect(result).toBe('mireia castrillo');
  })

  it('izena + abizena1 + abizena2', () => {
    const result = filterName('mireia castrillo gonzalez');
    expect(result).toBe('mireia castrillo gonzalez');
  })

  it('izena + abizena1 + abizena2 + hutsune', () => {
    const result = filterName('mireia castrillo gonzalez ');
    expect(result).toBe('mireia castrillo gonzalez');
  })

  it('izena + abizena1 + abizena2 + hutsune bat baino gehiago', () => {
    const result = filterName('mireia castrillo gonzalez  ');
    expect(result).toBe('mireia castrillo gonzalez');
  })
})
```

**Irudia 39: Izen-abizenak eskuratzeko testaren kodea**

Test honen bidez, hasieran egin zen hutsuneen kudeaketa planteamendua ez zela egokia ikusi zen. Hau da, hutsuneak aztertzen ez dituzten kasuetan ez zen akatsik sortzen eta besteetan bai. Hortaz, metodo honen bidez, hutsune kudeaketan arazo bat zegoela jakiteak zergatia bizkor aurkitzen eta erraz zuzentzen lagundu zuen.

#### 4. Data

Test hau gaurko egunetik aurrerako eginbeharrak ondo lortzen direla frogatzeko egin da. Ikasleak irakaskuntza plataformako hasiera orrialdean gaurko egunetik aurrera burutu beharreko eginbeharrak soilik eta ez iraganekoak ere ikustea lortu behar da. Honetarako 3 kasu ezberdin baino ez dira aztertuko:

1. Iraganeko data onartzen aztertu
2. Gaurko data onartzen aztertu
3. Etorkizuneko data onartzen aztertu

```
describe('data gaurko egunarekin konparatu', ()=>{
  it('iraganeko eguna', ()=>{
    const result=dataEguneratua('2018','02','22');
    expect(result).toBe(false);
  })
  it('gaurko eguna', ()=>{
    const result=dataEguneratua('2018','07','09');
    expect(result).toBe(true);
  })
  it('etorkizuneko eguna', ()=>{
    const result=dataEguneratua('2018','10','25');
    expect(result).toBe(true);
  })
})
```

**Irudia 40: Data konparaketarako testaren kodea**

Test honen bidez, gaurko eguna baino aurreragoko datak kontutan ez direla izaten ikusi da. Hau da, eginbeharren osagai hori ondo sortu dela hasieratik.

#### 5. Hilabetea gehitu

Egutegi bat sortzerakoan kontutan izan behar da hilabete batetik hurrengorako trantsizioa. Hortaz, test honen bitartez, trantsizio hori modu egokian egiten dela bermatu nahi izan da plataformako egutegia zentzugabea ez izateko.

Honetarako 3 kasu ezberdin aztertu dira:

1. 0Xtik 0Xrako aldaketa
2. 09tik 10rako aldaketa

### 3. 12tik 01rako aldaketa

Hauen probak honakoak izanik:

```
describe('hilabetea gehitu', () => {
  it('hilabetea < 09', () => {
    const result = hilabeteaGehitu('05');
    expect(result).toBe('06');
  })
  it('hilabetea >= 09 && hilabetea < 12', () => {
    const result = hilabeteaGehitu('09');
    expect(result).toBe('10');
  })
  it('hilabetea == 12', () => {
    const result = hilabeteaGehitu('12');
    expect(result).toBe('01');
  })
})
```

**Irudia 41: Hilabetea gehitzeko testaren kodea**

Test honen bidez aldaketa kritikoetan, hau da, 09tik 10ra eta 12tik 01ra akatsik ematen ez dela ikusiz, algoritmo hau ondo dagoela ondorioztatu da.

### 6. Hilabetea kendu

Aurreko atalean esan bezala, egutegi bat sortzerakoan kontutan izan behar da ere hilabete batetik aurrerako trantsizioa. Hortaz, test honen bitartez, trantsizio hori modu egokian egiten dela bermatu nahi izan da berriro ere plataformako egutegia zentzugabea ez izateko.

Honakoan ere 3 kasu ezberdin aztertu dira:

4. 01tik 12ra aldaketa
5. 0Xtik 0Xra aldaketa
6. 10tik 09ra aldaketa

Hauen probak honakoak izanik:

```
describe('hilabetea kendu', () => {
  it('hilabetea == 01', () => {
    const result = hilabeteaKendu('01');
    expect(result).toBe('12');
  })
  it('hilabetea > 01 && hilabetea <= 09', () => {
    const result = hilabeteaKendu('04');
    expect(result).toBe('03');
  })
  it('hilabetea > 09 && hilabetea <= 12', () => {
    const result = hilabeteaKendu('10');
    expect(result).toBe('09');
  })
})
```

**Irudia 42: Hilabetea kentzeko testaren kodea**



Test honen bidez aldaketa kritikoetan, hau da, 01tik 10ra eta 12tik 01ra akatsik ematen ez dela ikusiz, algoritmo hau ondo dagoela ondorioztatu da.

```
Karma v1.7.1 - connected  
Chrome 67.0.3396 (Windows 10.0.0) is idle  
Jasmine 2.99.0  
.....  
25 specs, 0 failures  
  
ordua  
ordua kate modura  
  
izena bihurtu  
izen abizenak hizki larrietara pasa: guztiak  
izen abizenak hizki larrietara pasa: izena  
izen abizenak hizki larrietara pasa: abizena1  
izen abizenak hizki larrietara pasa: abizena2  
izen abizenak hizki larrietara pasa: izena&abizena1  
izen abizenak hizki larrietara pasa: izena&abizena2  
  
izen abizenak tortu  
izena soilik  
izena + hutsune  
izena + hutsune bat baino gehiago  
izena + abizena1  
izena + abizena1 +hutsune  
izena + abizena1 +hutsune bat baino gehiago  
izena + abizena1 +abizena2  
izena + abizena1 +abizena2 +hutsune  
izena + abizena1 +abizena2 +hutsune bat baino gehiago  
  
data gaurko egunarekin konparatu  
iraganeko eguna  
gaurko eguna  
etorkizuneko eguna  
  
hilabetea gehitu  
hilabetea<09  
hilabetea>=09 && hilabetea<12  
hilabetea==12  
  
hilabetea kendu  
hilabetea==01  
hilabetea>01 && hilabetea<=09  
hilabetea>09 && hilabetea<=12
```

**Irudia 43: Guztira burututako unitate probak**

Atal honetan zehar eta goiko irudian ikus daitekeen bezala, unitate proba hauek web aplikazioaren 6 osagai balioztatzeko erabili dira eta kasuren baten hasieran errorea egon baldin bada ere, aldaketa txiki batzuk eginik osagaiak ondo sortuta daudela ikusi da.

## INTEGRAZIO PROBAK

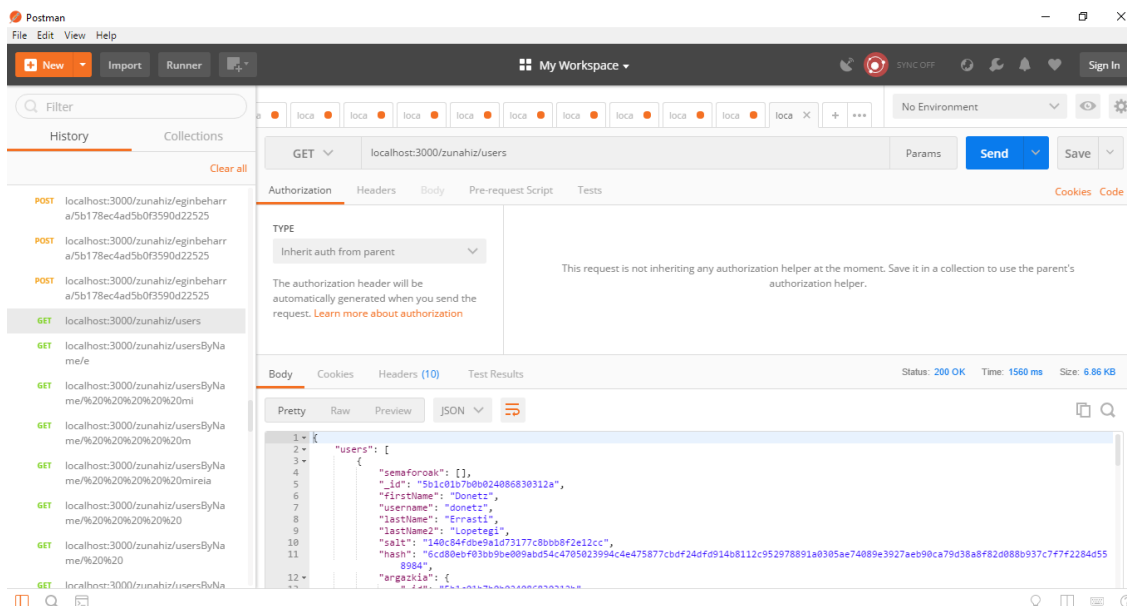
Proiektu honen integrazio probak TypeScript eta Postman-en bitartez egin dira gehien bat.

*Postman* HTTP request metodo desberdinak idatzi eta automatizatzeko balio duen framework-a da. Honen bitartez, zerbitzariak eskaintako REST zerbitzuak behar bezala funtzionatzen dutela era bizkor batean balioztatzeko erabili da, eskaera baten aurrean jasotakoa espero dena dela ikusiz.



#### Irudia 44: Postman tresnaren logotipoa

Proiektuan zehar Postman bidezko integrazio probak hainbat aldiz egin dira, batez ere APIa sortzerakoan. Hortaz, proba baten adibidea jarriko da honen erabileraren ideia bat egiteko:



#### Irudia 45: Postman tresna

*TypeScript* aldiz, Angular-en produkzio-kodea idazteko erabili denez, osagai desberdinek elkarren artean duten eragina frogatzeko ere erabili da. Hain zuzen ere, baieztatu nahi ziren integrazioak “*console.log*”-en bitartez kudeatu dira, proba desberdinak eginik eta frogatu nahi diren osagaien parametroek dituzten balioak aztertuz. Modu honetan, osagai desberdinek elkarren artean duten eragina ikusi ahal izan da, besteak beste, muturreko balioak balioztatuz, array tamaina desberdinak...



#### Irudia 46: TypeScript-en logotipoa

Proiektuan zehar, hainbat aldiz egin dira momentuko balioen bistarapenak “**console-log**” erabiliz, lortutako edo bidalitako balioak lortu beharrekoak ziren edo ez aztertzeko eta guzti hau probak automatizatu gabe.

Aplikazioa osotasunean frogatu behar izan ez den arren, behin eta berriz berrabiatzeko premia egon da. Honen hobekuntza bat integrazio probetarako Jasmine eta Karma ere erabiltzea izango litzateke. Modu honetan, unitate probetan bezala atzealdean etengabe exekutatu litzatekeen testak idatzi ahalko ziren. Aplikazioaren proben garapenean bide hau ez da jorratu proiektuaren mugen ondorioz, horrelako integrazio probak Jasmine bitartez idazteko hauek behar bezala egiten ikasteko denbora dezente behar delako.

Kasu honetan ere TypeScript bidezko hainbat proba egin direnez, horren adibide gisa hurrengo irudia jarriko da:



```
▼ User {_id: "5b17b828e6850e0d2c9267d3", username: "mireia", firstName: "mireia", LastName: "", LastName2: "", ...} view-user.component.ts:123
  argazkia: "5b17b828e6850e0d2c9267d4"
  description: ""
  firstName: "mireia"
  hash: ""
  lastName: ""
  lastName2: ""
  portada: "5b17b828e6850e0d2c9267d5"
  salt: ""
  username: "mireia"
  _id: "5b17b828e6850e0d2c9267d3"
  __proto__: Object
▶ {token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiJ1Yy5k1fQ.LybMbfSIRSI1A-iQ9_R-MimE4aIGkgUbkhgrwnABltQ"} view-user.component.ts:124
params: 5b229dd7bba26204ecb10837 view-user.component.ts:173
ni naiz? false view-user.component.ts:180
```

**Irudia 47: TypeScript bidezko integrazio proba**

## END-TO-END PROBAK

Proiektuan end-to-end probak *Google Chrome* nabigatzailearen bidez burutu dira, aplikazioa osotasunean abiarazi eta osagaiak behar bezala erantzuten dutela egiaztatuz.

Burututako proba hauek, lehenago aipatu den bezala, ez dira automatizatzen eta oso geldoak dira, dena martxan jarri arte itxaron beharra dagoelako. Dena den, proiektuan zehar asko erabili den baliabide bat izan da, funtsean, aplikazioa edo osagai jakin bat ea ondo dabilen ikusteko orduan, erantzun fidagarriena lortzen delako.

Proba mota honetan giltza diren aplikazioak balioztatu dira: semaforoan funtzionamendua egokia dela, fitxategiak egoki igo eta irekitzen direla, aplikazioaren orrialdeen arteko fluxua egokia dela, ... Eta bestetik, osagaiak nabigatzailean nola ikusten diren egiaztatzeke ere erabili da proba mota hau, esaterako, luzera desberdineko izen-abizenak sartuz, argitalpenen izenburu eta deskribapen tamaina ezberdinak sartuz, elementu kopuru desberdinak bistaratu, ... kasu guztietan pantailan dena behar bezala marrazten dela ikusiz.

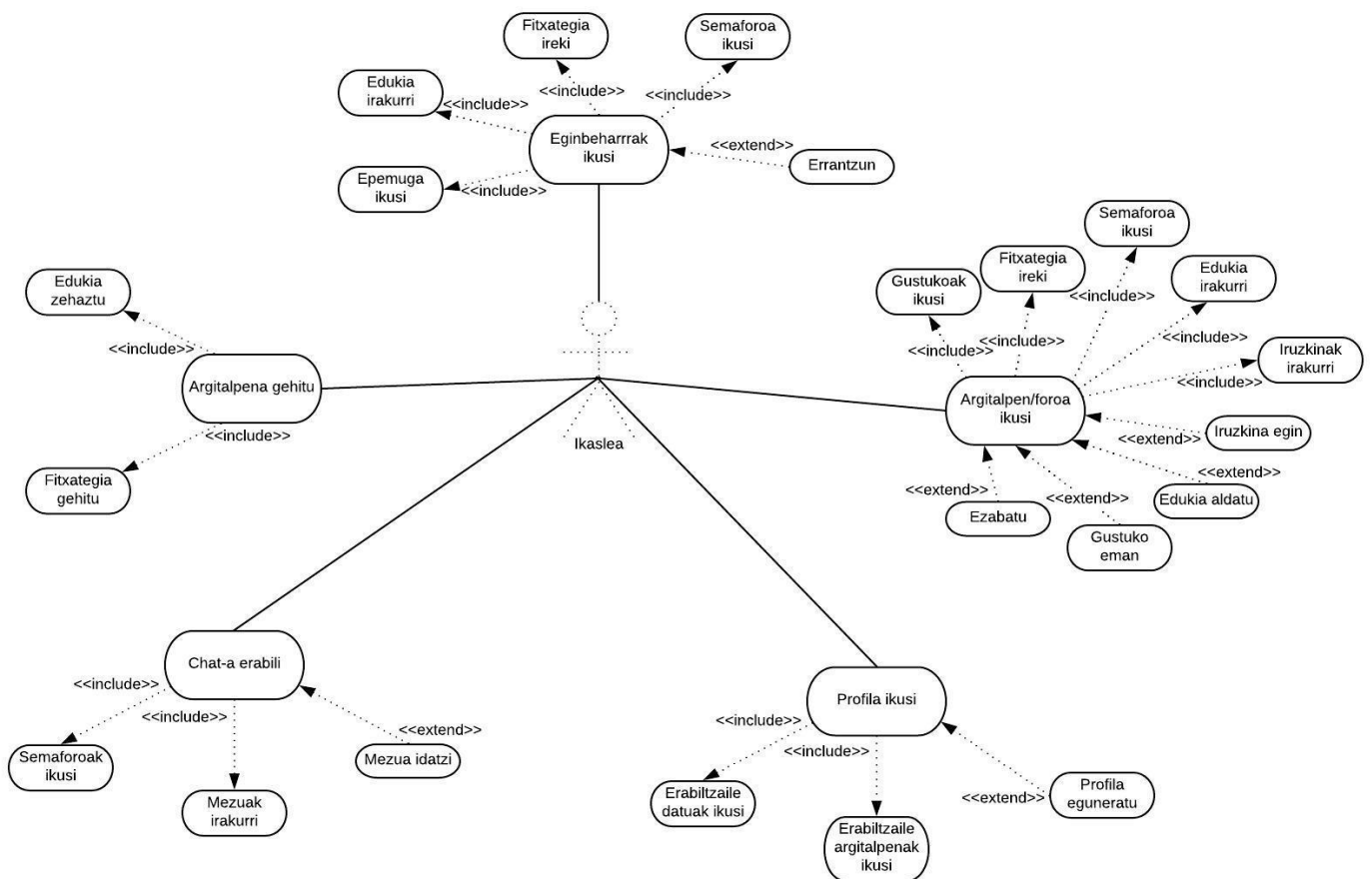
### III. ERANSKINA: EMAITZAK

Garatu den irakaskuntza plataformak bere xedea eta beharrezko funtzionalitateak betetzen dituela bermatzeko, aplikazioa bi ikuspuntu ezberdinetatik aztertu behar da.

Alde batetik, hasierako diseinuan ikaslearentzat beharrezkoak ziren funtzionaltasunak inplementatzea lortu den ikusiko da eta horrela izanik, nola lortu diren. Beste aldetik, plataformaren administrari eta kudeatzaile denak (kasu honetan, irakasleak) plataforma bat kudeatzeko beharrezkoak diren funtzionalitate minimoak dituela ere ikusiko da.

#### Ikasle ikuspegia

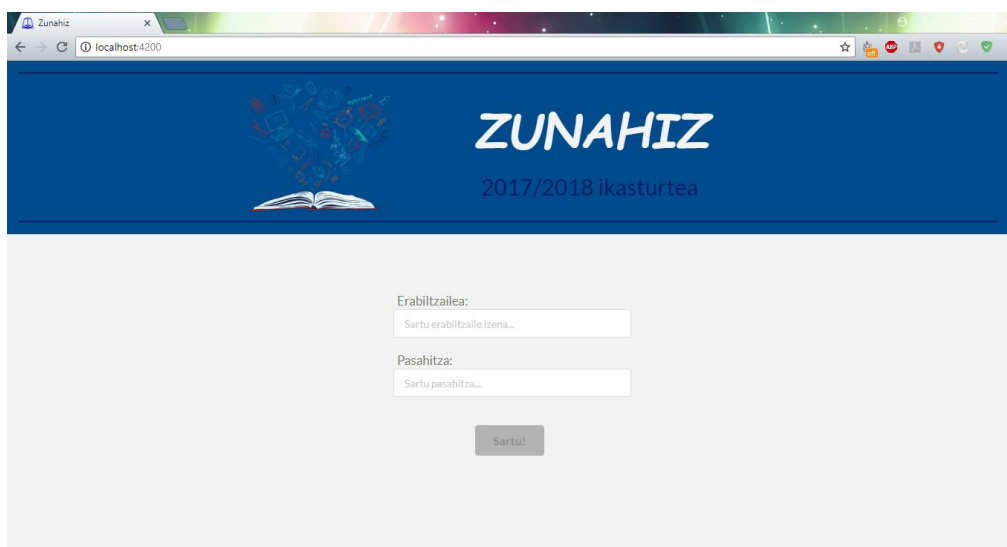
Sortu den irakaskuntza plataforma hau, ikasleei 48.irudian ikusten diren funtzionalitateak eskaintzeko diseinatu da.



Irudia 48: Ikasleen ikuspuntuko erabilera kasuen diagrama

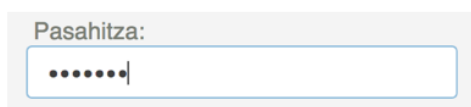
## HASIERA ORRIALDEA

Orrialde honetan erabiltzaileek haien burua identifikatuz plataformara sartzea lortuko dute. Horretarako, identifikatzaile modura bakarra den *username* parametroa eta honi dagokion pasahitza erabiliko dira.



**Irudia 49: Plataformako Login orrialdea**

Segurtasunari dagokionez, pasahitza sartzerakoan ez dira teklatuan sartutako hizkiak gordin agertzen.

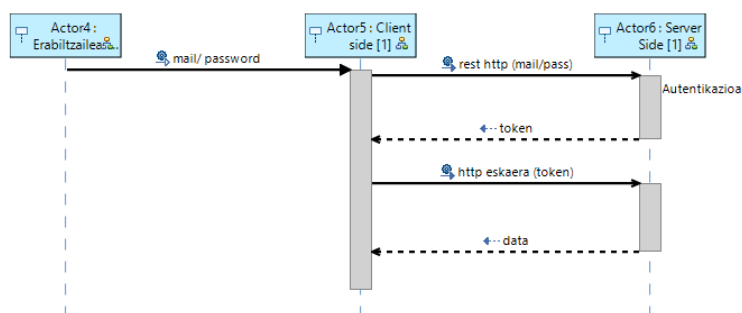


**Irudia 50: Pasahitza sartzeko eremua**

Horrez gain, pasahitzak ez dira datu basean gordin biltegitzen. Horren ordez, *hash*-eatu egiten dira eta *hash*-a da biltegitzen dena. Modu honetan, segurtasun eraso baten ondorioz norbaitek datu basera atzipena izango balu, ezingo luke erabiltzaileen pasahitza lapurtu. Bestalde, segurtasun neurri gehigarri gisa, *hash*-eatutako pasahitzaz gain ausazko era baten sortuko den *salt* eremu bat ere erabiliko da. *Salt* eremu honek, *hash*-aren atzizki bezala jokatuko du, bere deskodifikatzea oraindik zailagoa eginik.

Behin plataformara atzipena lorturik, hainbat gauza burutu ahal dira. Hauetako askotan, aplikazioa erabiltzen ari den erabiltzailea nor den identifikatzea ezinbestekoa izango denez, sesioa mantentzen duen modulu baten diseinua ere burutu da.

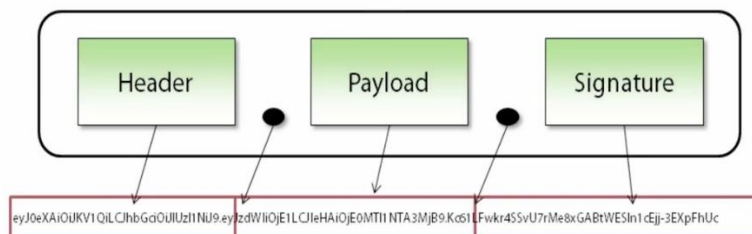
Sesioa mantentzeko plataforma honetan *token*-ak erabili dira.



**Irudia 51: Token bidezko sesio mantentzearen sekuentzia diagrama**

*Token*-ak erabiltzaile bakoitza identifikatzeko balio duten sinatutako ziurtagiriak dira. Erabiltzailea autentikatzen denean, itzuleran token bat jasoko du. Erabiltzaileak token-a bere localStorage-an gordeko du. Momentu horretatik aurrera zerbitzarira bidalitako HTTP eskaera bakoitzaren goiburuan token-a ere bidaliko dio. *Token*-a JSON objektu modura sartuko du, JWT (Jason Web Token) objektu gisa, hain zuzen ere.

## What is a JWT?



**Irudia 52: JWT-aren egitura**

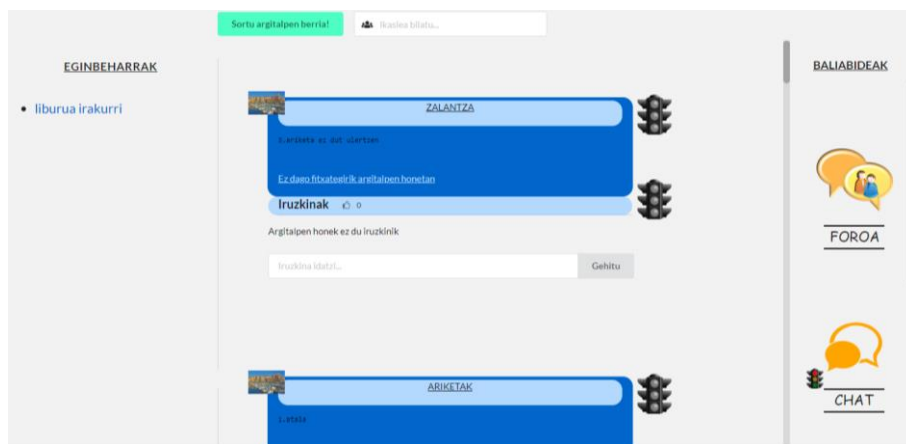
JWT-a “.” batez banatuta dauden 3 karaktere-katez osatutako String luze bat da. Hauetako karaktere bakoitzak eremu bat osatzen du.

- Header: Token-a sortzeko erabilitako kodifikazioaren informazioa ematen duen eremua.
- Payload: Sesioak iraun bitartean erabiltzaileak biltegitratzea nahi den atributuak doaz eremu honetan: erabiltzailearen id-a, izena, profil-irudia ... Horrez gain, iraungitze marka bat ere sar daiteke, token-a soilik zehaztutako denbora tartean baliagarria izan dadin.

- Signature: Zerbitzarian biltegitratuta dagoen gako batekin kodifikatutako ziurtagiri sinatua da. Honek token-a soilik autentikatutako erabiltzaileak eta zerbitzariak deskodifikatu ahal izatea ahalbidetzen du.

Ikus daitekeenez, token-ak erabiltzeak zerbitzarian sesio egoerari buruzko ezer gorde behar ez izatea ahalbidetzen du; erabiltzailea izango da bere sesioaren informazioa baitan duen token-a biltegitratuko duena. Horrela, zerbitzaria *stateless* izatea lortzen da, aplikazioa oso eskalagarria bilakatuz.

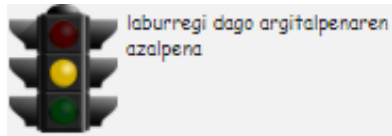
## MAIN PAGE



**Irudia 53: Main page orrialdea**

Orrialde honetan ikasleak egin ahal dituen gauzak:

- Gaurko egunetik aurrerako eginbeharrak ikusi
- Argitalpen berriak sortu
- Plataformako beste ikasleak bilatu
- Forora sartu
- Chatera sartu
- Argitalpenak ikusi: Izenburua, azalpena, fitxategia, iruzkinak eta irakasleak baloratutako semaforo kolore eta iruzkina ikusi eta argitalpen sortzailearen profilerara joan.
  - Semaforoak: Irakasleak argitalpeneko eta iruzkinetako edukiaren balorazioa ikasleei adierazteko erabiltzen da, honek bakarrik alda dezakeelarik hauen balioa. Semaforo irudiaz gain, testu atal bat ere badu irakasleak semaforoaren kolorearen zergatia azaltzeko.



Irudia 54: Semaforo baten adibidea

## PROFILE

A screenshot of a user profile page. At the top right is a large landscape photo of a rocky coastline with a small building on a cliff. Below it is a blue button labeled 'Eguneratu profila'. Underneath are two tabs: 'DATUAK' and 'EGINDAKO ARGITALPENAK'. To the left of the profile information is a smaller photo of a city at night. The profile details are: 'Izena: Mireia', 'Abizenak: Castrillo Gonzalez', and 'Deskribapena:'. At the bottom left are two icons: 'PLE' (a circular logo with various symbols) and 'FOROA' (a speech bubble icon with two people). The footer contains the email 'mcastrillo003@ikasle.ehu.eus' and the name 'Mireia Castrillo'.

Irudia 55: Profile baten orrialdea

Orrialde honetan ikasleak egin ahal dituen gauzak:

- Portada, argazkia eta datuak ikusi
- Egindako argitalpenak ikusi
- Profila eguneratu (jabea den kasuan soilik)
- Forora edo *Symbaloo* plataformara joan



## DUTIES

ariketak
laburpena
liburua irakurri
Itzulpena
Iritzi artikulua

**LIBURUA IRAKURRI**

Uda honetan euskarazko liburu bat irakurri beharko duzue. Gai libreduna, baina gutxienez 128 orrialdekoa izan beharko da.

Ez daغو fitxategirik argitalpen honetan

Noizko: 2018/9/1

Argitalpena sortu

Irudia 56: Duties orrialdea

Orrialde honetan ikasleak egin ahal dituen gauzak:

- Eginbeharrak ikusi
- Eginbeharra erantzun
- 

## FOROA

Sortu foro berria!

**EGINBEHARRAK**

2018

IRAILA

1 liburua irakurri

**ZALANTZA**

3. ariketa ez dute ulertzen

Ez daغو fitxategirik argitalpen honetan

**Iruzkinak**

Argitalpen honek ez du iruzkinik

Iruzkina idatzi...

Gehitu

**IKASLEAK**

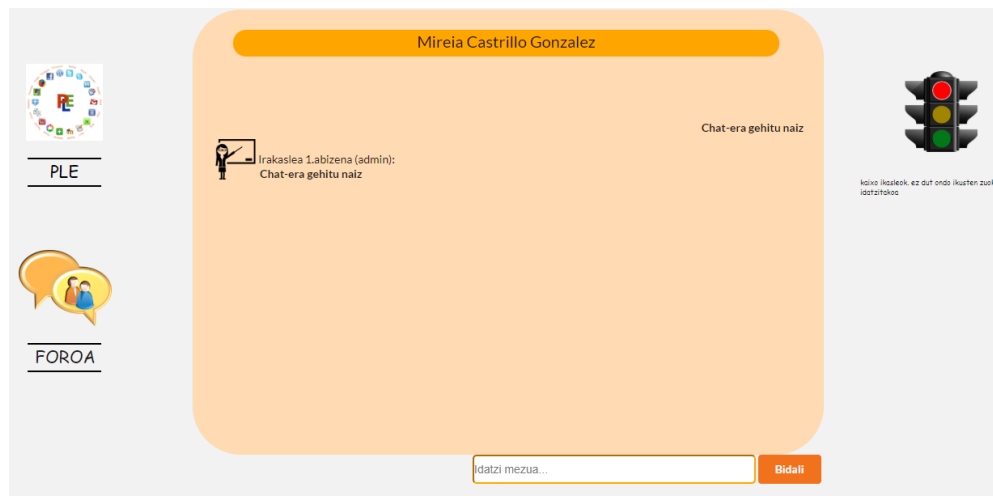
- Donetz Errasti Lopetegi
- Eneritz German Zabala
- Itxaso Gonzalez Zugazaga
- Itziar Eguskiza Garcia
- Jon Urteaga Urizarbarrena
- Maider Castrillo Gonzalez
- Maite Fontecha Garate
- Mikel Blazquez Tatlegi
- Mireia Castrillo

Irudia 57: Foroa orrialdea

Orrialde honetan ikasleak egin ahal dituen gauzak:

- Eginbeharrak egutegi batean ikusi
- Foroko argitalpenak ikusi: Izenburua, azalpena, fitxategia, iruzkinak eta irakasleak baloratutako semaforo kolore eta iruzkina ikusi eta argitalpen sortzailearen profilerara joan.
- Foro-argitalpen berria sortu
- Plataformako ikasleen zerrenda ikusi

## CHAT



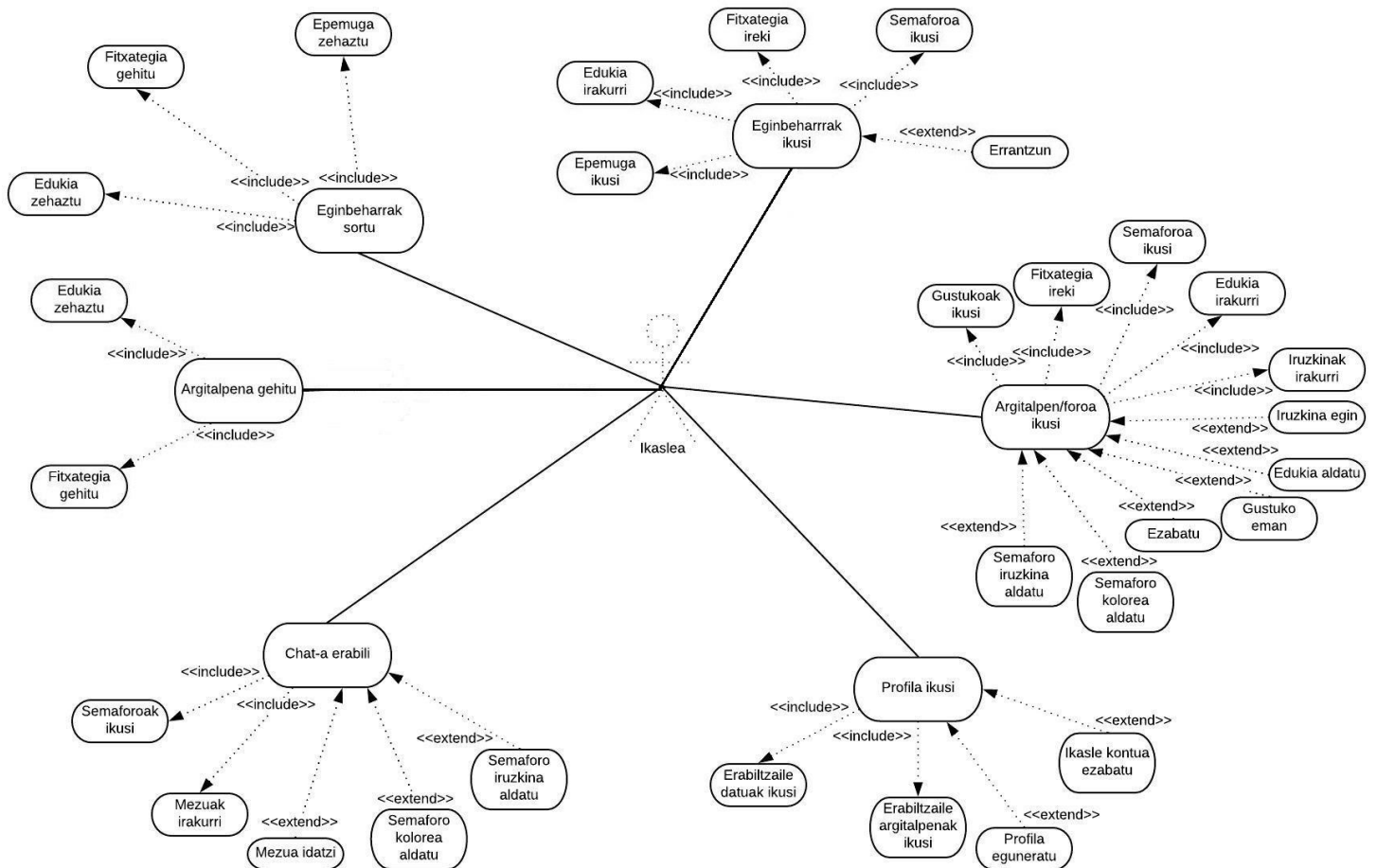
**Irdia 58: Chat orrialdea**

Orrialde honetan ikasleak egin ahal dituen gauzak:

- Mezuak irakurri eta idatzi
- Semaforoa ikusi
- Forora edo *Symbaloo* plataformara joan

# Irakasle ikuspegia

Irakaslearen ikuspegia ikasleen antzekoa da baina pare bat funtzio gehiturik, semaforo, eginbehar eta ikasleen kudeaketekin zerikusia dutenak.

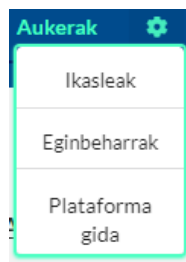


**Irudia 59: Irakasleen ikuspuntu erabilera kasuen diagrama**

## MAIN PAGE

Orrialde honetan irakasleak egin ahal dituen aparteko gauzak:

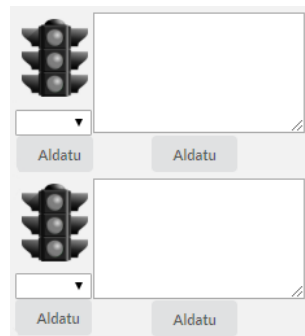
- Aukera botoian klik eginez, beste hiru orrialdetara joateko aukera



**Irudia 60: Aukerak botoia**

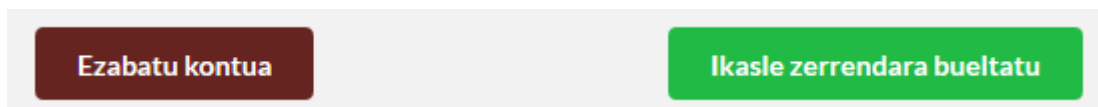
- Ikasleak: *Students* orrialdera berbidalketa
- Eginbeharrak: *Duties* orrialdera berbidalketa
- Plataforma gida: Plataformako osagaien (semaforo, eginbehar..) azalpena duen PDF fitxategia

- Semaforoak kudeatzeko aukera



Irudia 61: Semaforo kudeaketa

## PROFILE



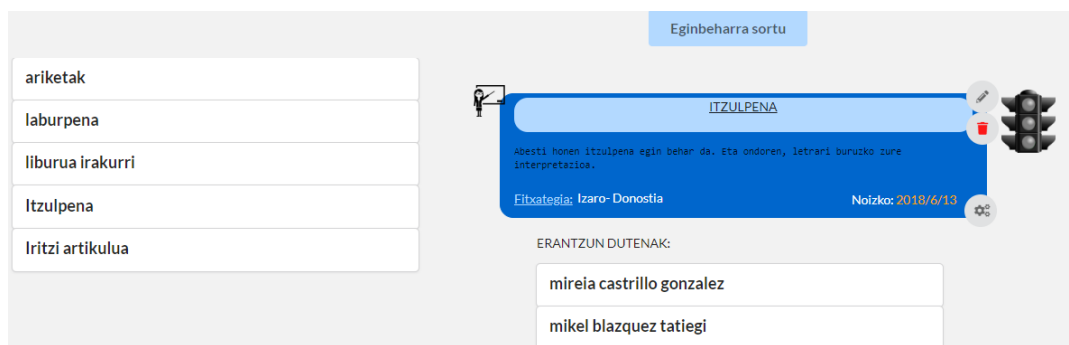
Irudia 62: Irakasleak profilak kudeatzeko botoiak

Orrialde honetan irakasleak egin ahal dituen aparteko gauzak:

- Ikasle kontua ezabatu
- Ikasle zerrendara bueltatu

## DUTIES

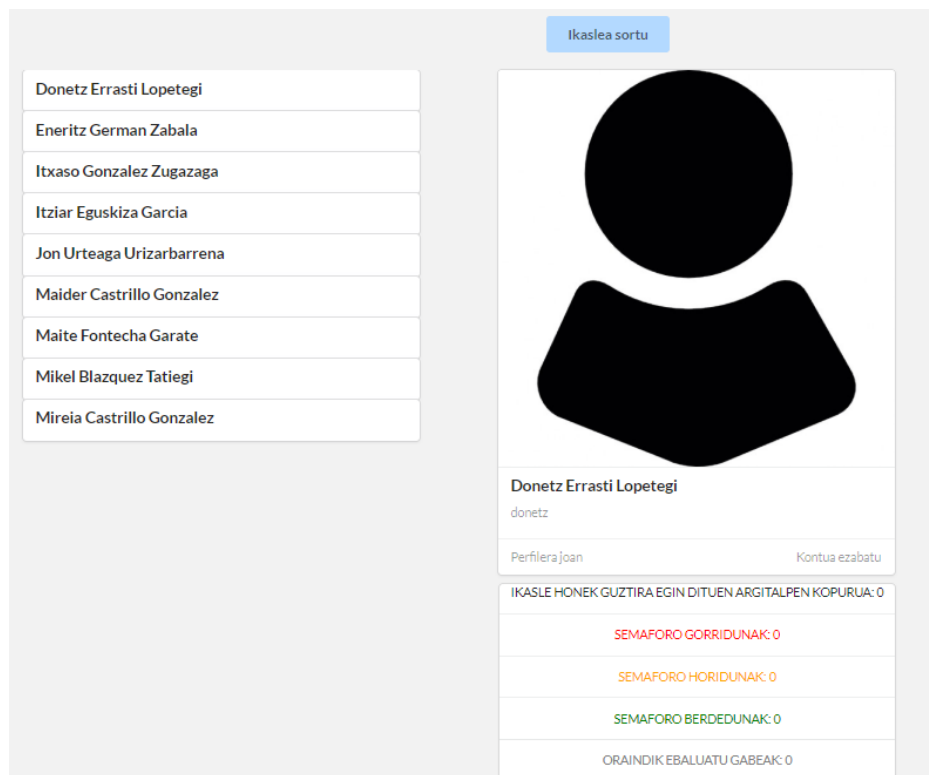
Orrialde honetan irakasleak eginbeharrak kudeatu ahal ditu eta bakoitzean zein ikasleek erantzun duen ikusi:



Irudia 63: Eginbeharrak kudeatzeko orrialdea

## STUDENTS

Orrialde honetan irakasleak ikasleak kudeatu ahal ditu eta hauen estatistikak ikusi:



The screenshot displays a user interface for managing students. On the left, there is a list of student names. On the right, a detailed profile for 'Donetz Errasti Lopetegi' is shown, including a placeholder for a profile picture, a search bar, and a traffic light status indicator.

Ikaslea sortu
Donetz Errasti Lopetegi
Eneritz German Zabala
Itxaso Gonzalez Zugazaga
Itziar Eguskiza Garcia
Jon Urteaga Urizarbarrena
Maidier Castrillo Gonzalez
Maite Fontecha Garate
Mikel Blazquez Tatiegi
Mireia Castrillo Gonzalez

**Donetz Errasti Lopetegi**  
donetz

Perflera joan Kontua ezabatu

IKASLE HONEK GUZTIRA EGIN DITUEN ARGITALPEN KOPURUA: 0

- SEMAFORO GORRIDUNAK: 0
- SEMAFORO HORIDUNAK: 0
- SEMAFORO BERDEDUNAK: 0

ORAINDIK EBALUATU GABEAK: 0

**Irudia 64: Ikasleak kudeatzeko orrialdea**

## FOROA

Orrialde honetan irakasleak duen desberdintasun bakarra, honek semaforoak kudea ditzakeela da.

## CHAT

Orrialde honetan irakasleak duen desberdintasun bakarra, honek semaforoak kudea ditzakeela da.

## IV. ERANSKINA: APLIKAZIOA

### Plataforma

Proiektu honetan garatu den irakaskuntza plataforma erabilgarria izan daiteke irakaskuntza eta hezkuntza arloko beste hainbat alderditan honi bestelako funtzionaltasun bat eman nahirik, edota beste proiektu batzuetan erabiltzea ere oso interesgarria izan daiteke.

Horregatik, aplikazioa lainora igo da bertatik baliabidea eskuratu ahal izateko. Horrela ba, Github plataformaren bitartez proiektuaren jatorrira heltzen den edozeinek proiektuaren errepositorioa deskargatzeko aukera izango du.

Proiektuaren Github helbidea:

<https://github.com/mcastrillo003/Zunahiz-GrAL>

### Abiaraztea

Behin proiektua deskargatuta dagoelarik, hau abiarazteko Cygwin terminal bi erabili behar dira. Terminal baten bidez plataformako zerbitzaria abiarazten da eta bigarren terminalaren bidez bezero aldea.

Lehenengo terminalean, proiektuko API karpetera arte sartu behar da eta bertan “**npm start**” agindua exekutatu.



```
/cygdrive/c/wamp/www/Zunahiz-GrAL/api
Mireia@Mireia /cygdrive/c/wamp/www
$ cd Zunahiz-GrAL/api/
Mireia@Mireia /cygdrive/c/wamp/www/Zunahiz-GrAL/api
$ npm start
```

Irudia 65: API-a abiarazteko prozesua

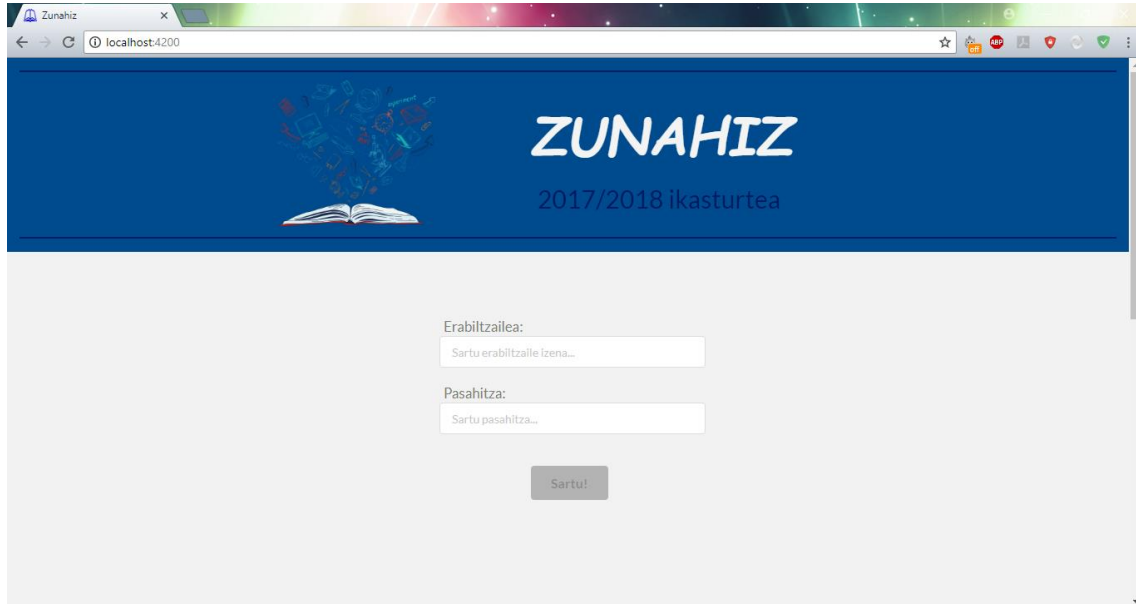
Bigarren terminalean aldiz, CLIENT karpetera arte sartu eta bertan ere “**npm start**” agindua exekutatu behar da.



```
/cygdrive/c/wamp/www/Zunahiz-GrAL/client
Mireia@Mireia /cygdrive/c/wamp/www
$ cd Zunahiz-GrAL/client/
Mireia@Mireia /cygdrive/c/wamp/www/Zunahiz-GrAL/client
$ npm start
```

Irudia 66: CLIENT-a abiarazteko prozesua

Behin zerbitzaria eta bezeroa abiarazi direla, nabigatzaile baten bidez plataformara sartuko da. Horrela ba, nabigatzailean “localhost:4200” jarriz, Zunahiz plataformako hasiera orrialdea lortuko da.



**Irudia 67: Plataforma hasieratzea**

\*\*CLIENT karpeta “martxan jarri” fitxategian komando hauek idatzita datoz, proiektua deskargatu eta nola abiarazi ez dakienarentzat.