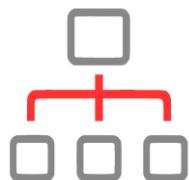


Informatika Fakultatea

Informatika Ingeniaritzako Gradua

▪ Gradu Amaierako Lana ▪

Software Ingeniaritza



ProMeta

Metaereduetan oinarritutako softwarearen garapenerako
prozesuen definizio eta ezarpenerako sistema

Egilea: Julen Etxaniz Aragoneses

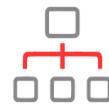
Tutorea: Juan Manuel Pikatza Atxa

Data: 2021

Laburpena

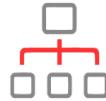
Proiektuaren helburua metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema eraikitzea da. Izan ere, softwarea garatzeko egokiak diren hainbat metodologia existitzen dira. Garrantzitsua da metodologia horien informazioa eredu bidez definitzea, etorkizunean malgutasunez kudeatu eta hobekuntzak egin ahal izateko. Gainera, beharrezko da eredu informazioa erabiliz metodologia ezartzen duen sistema eraikitza, garapen taldeak proiektuetan erabiltzeko. Proiektuaren garapenerako, OpenUP metodologia erabili da eta proiektuaren dokumentazioa eta memoria idazteko CCII-N2016-02 estandarra.

Hitz gakoak: metaeredua, metodologia, araua, bizi-zikloa, MDE, CMS

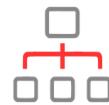


Aurkibide Orokorra

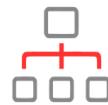
Laburpena.....	2
Aurkibide Orokorra.....	3
Irudien Aurkibidea.....	10
Taulen Aurkibidea.....	11
1 Sarrera	12
2 Helburuak	14
2.1 Arazoaren Planteamendua	14
2.2 Produktuaren Planteamendua.....	15
2.3 Interesatuen Deskribapena.....	15
3 Aurrekariak	17
3.1 Software Kalitatea.....	17
3.2 Bezeroen Eskakizun Gogorrak.....	17
3.3 Metodologiak.....	18
3.4 Arauak.....	20
3.5 BPM.....	21
3.6 Inferentzia Motorrak.....	22
3.7 MDE.....	24
3.8 CMS.....	25
3.9 ProWF	25
4 Egungo Egoera	27
4.1 Deskribapena	27
4.2 Abantailak eta Desabantailak	27
4.2.1 Abantailak.....	27
4.2.2 Desabantailak	27
4.3 Proposatutako Hobekuntzak	28
4.4 Prestakuntza	29
5 Arauak eta Erreferentziak.....	31
5.1 Aplikatutako Legedia eta Araudia	31
5.2 Bibliografia	31
5.3 Metodoak.....	32
5.3.1 OpenUP.....	32
5.3.2 ABRD	32
5.3.3 MDE	33
5.4 Tresnak.....	33
5.4.1 Git	33



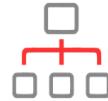
5.4.2	GitHub.....	33
5.4.3	GitHub Pages	33
5.4.4	Toggle Track.....	33
5.4.5	Java	33
5.4.6	Eclipse IDE.....	33
5.4.7	EPF Composer.....	34
5.4.8	Xtext.....	34
5.4.9	XSLT	34
5.4.10	Visual Studio Code.....	34
5.4.11	Microsoft Office	34
5.4.12	PlantUML.....	34
5.4.13	Draw.io	35
5.4.14	Drupal.....	35
5.4.15	XAMPP.....	36
5.4.16	Pantheon	36
5.4.17	MariaDB	37
5.4.18	phpMyAdmin.....	37
5.4.19	PHP	37
5.5	Ereduak	37
5.5.1	CCII-N2016-02.....	37
5.5.2	OpenUP.....	37
5.5.3	ABRD	38
5.5.4	UMA.....	38
5.5.5	Ecore	38
5.6	Metrikak	38
5.6.1	Irismena	38
5.6.2	Denbora	38
5.6.3	Kostua	38
5.6.4	Kalitatea	39
5.7	Prototipoak	39
5.7.1	ProMeta ModelEditor	39
5.7.2	ProMeta IO-System	39
6	Definizioak eta Laburdurak	40
6.1	Bizagi	40
6.2	BPM	40
6.3	CCII	40
6.4	CCII-N2016-02	40



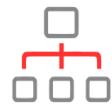
6.5	CMMI	40
6.6	CMS.....	40
6.7	DOT	40
6.8	Drupal	40
6.9	IBM.....	40
6.10	OpenUP.....	40
6.11	PlantUML	40
6.12	ProMeta	41
6.13	ProWF	41
6.14	RUP.....	41
6.15	SPICE	41
6.16	UML.....	41
6.17	UNE	41
6.18	URPS.....	41
6.19	Workflow	41
6.20	WYSIWYG	41
7	Hasierako Betekizunak.....	42
7.1	Betekizun Funtzionalak	42
7.2	Betekizun Ez-Funtzionalak	42
7.3	Sistemaren Ezaugarriak.....	43
7.3.1	Erabilgarritasuna.....	43
7.3.2	Fidagarritasuna.....	43
7.3.3	Errendimendua	43
7.3.4	Mantenugarritasuna.....	43
7.4	Sistemaren Interfazeak	43
7.4.1	Itxura eta Sentsazioa	43
7.4.2	Diseinu eta Nabigazio Betekizunak.....	43
7.4.3	Konsistentzia.....	44
7.4.4	Erabiltzailearen Pertsonalizazio Betekizunak	44
8	Irismena	45
8.1	Bizi-zikloa	45
8.2	LDE Diagrama	45
8.3	Artefaktuak	46
9	Hipotesiak eta Murriztapenak	48
9.1	Hipotesiak	48
9.2	Murriztapenak	48
10	Aukeren Azterketa eta Egingarritasuna	49



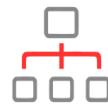
10.1	Arkitektura	49
10.1.1	Bizagi	49
10.1.2	ProWF	50
10.1.3	ProMeta	50
10.2	CMS	51
10.2.1	Wordpress	52
10.2.2	Drupal	52
10.2.3	Joomla	52
10.3	Dokumentazioa Hostinga	53
10.3.1	GitHub Pages	53
10.3.2	GitLab Pages	53
10.3.3	Netlify	53
10.4	Drupal Hostinga	53
10.4.1	000webhost	53
10.4.2	Heroku	53
10.4.3	Acquia	54
10.4.4	Pantheon	54
10.4.5	Platform.sh	54
10.5	Datu-base Kudeaketa Sistema	54
10.5.1	MySQL	54
10.5.2	PostgreSQL	54
10.6	Datu-basearen Sorrera	54
10.6.1	Inferentzia Motorra	54
10.6.2	Teneo	54
10.6.3	Xtext	54
10.7	Lanerako Ingurunea	54
10.7.1	Makina Birtuala	55
10.7.2	Ordenagailu Pertsonala	55
10.8	Bertsio Kontrola	55
10.8.1	GitHub	55
10.8.2	GitLab	55
10.9	Metaeredua	55
10.9.1	SPEM	55
10.9.2	UMA	55
10.10	Metodología	55
10.10.1	OpenUp	55
10.10.2	RUP	55



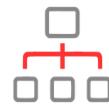
10.10.3	ABRD.....	56
10.11	Metodologia Editorea.....	56
10.11.1	EPF Composer	56
10.11.2	Rational Method Composer	56
10.11.3	Editore Grafikoa	56
10.11.4	Testu Editorea	56
10.12	Prozesua Bistaratu.....	56
10.12.1	DOT.....	56
10.12.2	XPDL	56
10.12.3	Webgunea	56
10.13	Denboraren Kontrola.....	56
10.13.1	Toggl Track	56
10.13.2	Clockify	57
10.13.3	WakaTime	57
10.14	Diagramak.....	57
10.14.1	PlantUML.....	57
10.14.2	Draw.io	57
10.14.3	UMLDoclet.....	57
11	Proposatutako Sistemaren Deskribapena	58
11.1	Azpisistemak	58
11.2	ModelEditor	58
11.2.1	Betekizunak	58
11.2.2	Analisia	59
11.2.3	Arkitektura	59
11.2.4	Diseinua.....	60
11.2.5	Garapena	61
11.2.6	Proba	63
11.3	IO-System.....	63
11.3.1	Betekizunak	63
11.3.2	Analisia	64
11.3.3	Arkitektura	64
11.3.4	Diseinua.....	65
11.3.5	Garapena	65
11.3.6	Proba	65
11.4	Hedapena	65
11.5	Etorkizunerako Hobekuntzak.....	66
12	Arriskuen Analisia	66



12.1	Arriskuak	66
12.2	Arriskuen Analisia	68
13	Proiekktuaren Antolamendua eta Kudeaketa	68
13.1	Proiekktuaren Antolamendua	68
13.1.1	Taldekideak eta rolak	69
13.1.2	Informazio-sistema.....	69
13.1.3	Komunikazio-kanalak	69
13.2	Proiekktuaren Kudeaketa	69
13.2.1	Integrazioaren Kudeaketa	69
13.2.2	Irismenaren Kudeaketa	69
13.2.3	Epeen Kudeaketa.....	70
13.2.4	Kostuen Kudeaketa	70
13.2.5	Kalitate Kudeaketa	70
13.2.6	Giza Baliabideen Kudeaketa	71
13.2.7	Komunikazioen Kudeaketa	71
13.2.8	Arriskuen Kudeaketa	71
13.2.9	Erosketen Kudeaketa	72
13.2.10	Interesatuen Kudeaketa.....	72
14	Denbora Planifikazioa	73
14.1	Mugarriak.....	73
14.2	Lan-atazak	74
14.3	Gantt Diagrama.....	75
14.4	Iterazioak	75
14.5	Neurtutako denborak	77
14.5.1	Denbora hilabeteka.....	77
14.5.2	Hilabete bateko denbora	77
14.5.3	Aste bateko denbora.....	78
14.5.4	Denbora atazaka	78
14.5.5	WakaTime	79
14.6	Desbiderapenak	79
15	Aurrekontuaren Laburpena	80
16	Dokumentuen Lehentasun Ordena	81
17	Memoriaren Eranskinak.....	82
17.1	Sarrerako Dokumentazioa	82
17.2	Analisia eta Diseinua	82
17.2.1	Arkitektura Kuadernoa	82
17.2.2	Analisiaren Eredua	82

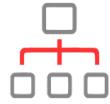


17.2.3	Diseinuaren Eredua	82
17.3	Tamaina eta Esfortzu Estimazioa	82
17.4	Kudeaketa Plana	82
17.4.1	Integrazioaren Kudeaketa	82
17.4.2	Irismenaren Kudeaketa	82
17.4.3	Epeen Kudeaketa.....	82
17.4.4	Produktuaren Kostuen Kudeaketa	82
17.4.5	Kalitate Kudeaketa	82
17.4.6	Giza Baliaabideen Kudeaketa	82
17.4.7	Komunikazioen Kudeaketa.....	82
17.4.8	Arriskuen Kudeaketa	82
17.4.9	Erosketen Kudeaketa	82
17.4.10	Interesatuen Kudeaketa.....	82
17.5	Segurtasun Plana	82
17.6	Beste Eranskinak	82
17.6.1	Hedapena	82
17.6.2	Garapena	82
17.6.3	Ingurunea	82
17.6.4	Proba	82
18	Sistemaren Espezifikazioa.....	82
18.1	Glosategia	82
18.2	Ikuspegia	82
18.3	Betebeharren Espezifikazioa.....	82
18.4	Erabilpen Kasuak.....	82
18.5	Erabilpen Kasuen Eredua	82
19	Aurrekontua.....	82



Irudien Aurkibidea

1.1. Irudia. Proiektuaren webgunean memoria bistaratu.	12
1.2. Irudia. Webguneko eranskinen eta barne informazioaren menuak.	13
3.1. Irudia. RUPen prozesu iteratiboa, bizi-zikloaren faseak, iterazioak eta jarduerak.	19
3.2. Irudia. OpenUP-en geruzak: mikro-gehikuntzak, iterazio bizi-zikloa eta proiektu bizi-zikloa.	20
3.3. Irudia. CCII-2016N-02 araua betetzen duen proiektuaren webguneararen egitura.	21
3.4. Irudia. 2018ko BPM Suiten Koadrante Magikoa, Gartner.	22
3.5. Irudia. 2019ko The Forrester Wave txostena, Forrester.	22
3.6. Irudia. EHSISen garapen ingurunea.	23
3.7. Irudia. Programazio tradisionalaren eskema.	23
3.8. Irudia. Erregeletan oinarritutako sistemek eskema: klaseak, objektuak, erregelak eta egitateak.	24
3.9. Irudia. MDERen 3 abstrakzio maila: eredu, metaeredua eta meta-metaeredua.	25
3.10. Irudia. ProWF sistemek behar dituen osagaiak eta haien arteko erlazioak.	26
5.1. Irudia. ABRD metodologiaren prozesuko faseak.	33
5.2. Irudia. OpenUP bizi-zikloa EPF Composer tresnan.	34
5.3. Irudia. PlantUMLren lengoain idatzitako sekuentzia-diagrama.	35
5.4. Irudia. Draw.io diagrama adibidea.	35
5.5. Irudia. Proiektuan sortutako Drupal webgunea.	36
5.6. Irudia. XAMPP kontrol panela Apache eta MySQL hasieratzeko.	36
5.7. Irudia. ProMeta webguneko Pantheon kontrol panela.	37
5.8. Irudia. UMA metodologiako metodoaren eta prozesuaren kontzeptuak.	38
8.1. Irudia. OpenUP metodologiako bizi-zikloaren faseak.	45
8.2. Irudia. LDE diagrama.	46
10.1. Irudia. Bizagi Modeler erabiliz sortutako prozesua.	50
10.2. Irudia. BPMn eta workflowetan oinarritutako arkitekturak.	50
11.1. Irudia. ModelEditor eta IO-System azpisistemen arkitektura.	58
11.2. Irudia. ModelEditor azpisistemako erabilpen kasuen eredu.	59
11.3. Irudia. ModelEditor azpisistemaren arkitektura.	60
11.4. Irudia. ModelEditor-en paketeak eta paketeen arteko dependentziak.	62
11.5. Irudia. Pakete diagrama adibidea.	62
11.6. Irudia. Klase diagrama adibidea.	63
11.7. Irudia. IO-System azpisistemako erabilpen kasuen eredu.	64
11.8. Irudia. IO-System azpisistemaren arkitektura.	65
12.1. Irudia. Arriskuen inpaktu eta probabilitate analisia.	68
14.1. Irudia. Mugarri garrantzitsuen diagrama.	73
14.2. Irudia. Gantt diagrama.	75
14.3. Irudia. Proiektuaren denbora hilabetaka.	77
14.4. Irudia. Hilabete bateko denbora.	78
14.5. Irudia. Aste bateko denbora.	78
14.6. Irudia. Proiektuaren denbora atazaka.	79



Taulen Aurkibidea

2.1. Taula. Arazoaren Planteamendua.	14
2.2. Taula. Produktuaren Planteamendua.	15
2.3. Taula. Interesatuen deskribapena.	16
3.1. Taula. CMMI-ren 5 heldutasun-mailak.	18
7.1. Taula. Hasierako betekizun funtzionalak.	42
7.2. Taula. Hasierako betekizun ez-funtzionalak.	43
8.1. Taula. OpenUP metodologiako bete diren artefaktuak domeinutan sailkatuta.	47
10.1. Taula. CMS ezberdinen ezaugarrien konparaketa.	51
12.1. Taula. Identifikatutako arriskuen zehaztasunak.	68
13.1. Taula. Kalitate kudeaketako rolak eta ardurak.	71
14.1. Taula. Proiekuko mugarri garrantzitsuak.	73
14.2. Taula. Lan-atazen zehaztasunak.	75
14.3. Taula. Proiekuko iterazioen fasea, helburuak eta datak.	77
15.1. Taula. Software taldeko kideen kostu baremoa.	80
15.2. Taula. Aurrekontua giza baliabideen eta erreminten kostuekin.	80

1 Sarrera

Dokumentu hau Julen Etxaniz Aragonesesen Informatika Ingeniaritzako Gradu Amaierako Lanaren memoria da. Egilearen inguruko informazio gehiago jakiteko <https://julenetxaniz.eus/> webgunea kontsulta daiteke. Lan hau Euskal Herriko Unibertsitateko (UPV-EHU) Donostiako Informatika Fakultatean landu da, Juan Manuel Pikatza izanik tutorea.

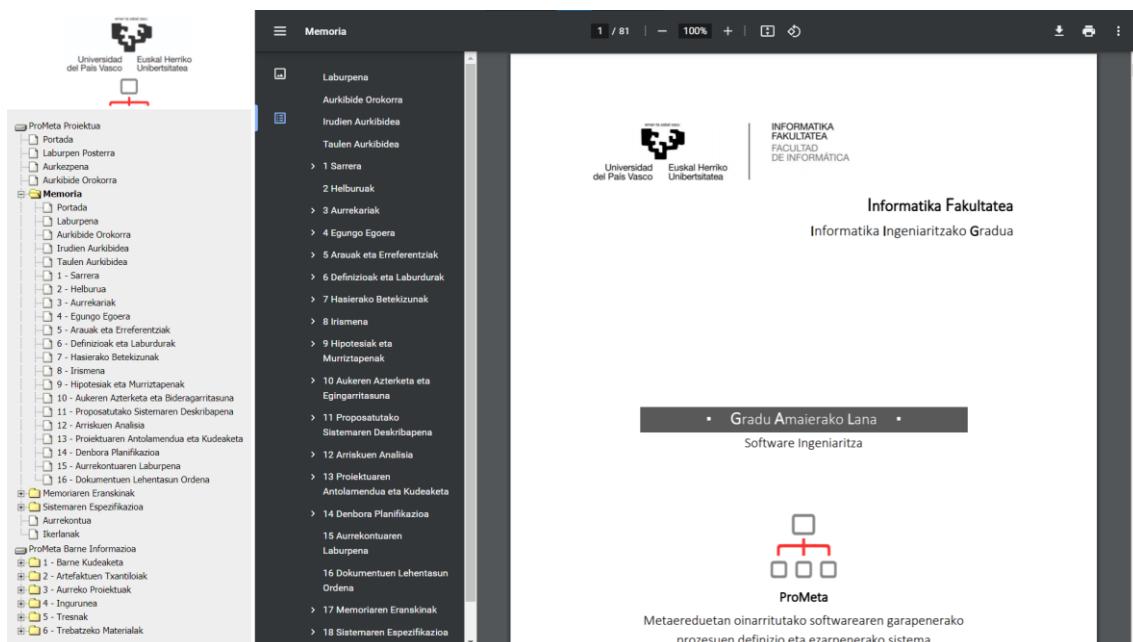
Dokumentu honetan ProMeta proiekturen inguruko aspektu guztiak azalduko dira. ProMeta izena prozesu edo profesional eta metaeredu hitzetatik dator. Portadan ikusten den proiektuaren logoak metaeredu bat irudikatzen du. Proiektuaren izenburua **Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema** da. Beraz, proiektuak bi helburu nagusi ditu: garapeneko prozesuen definizioa eta prozesuaren ezarprena.

Proiektua OpenUP metodologia erabilita gauzatu denez, bertako pausoak jarraitu dira. Gainera, dokumentazio formal eta profesionala lortzeko asmoz, ezaguna den CCII-N2016-02 arau estandarra eta Informatika Fakultateko GrAL eredua erabili dira memoriaren eta proiektuaren webgunearen atalak definitzeko.

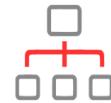
Proiektuaren webgunea honakoa da: <https://juletx.github.io/ProMeta/>. Webgune honen helburua ProMeta proiektuaren dokumentazio guzia biltzea eta proiektuko bezero zein interesdunek eskura izatea da. Memorian atal batzuetan webgunean dauden eranskinak aipatuko dira.

Webgunea bi zatitan dago banatuta, ezkerrean nabigazio menua dago eta eskuinean nabigazio menuan aukeratutakoa agertuko da. Ezkerreko menua erabiliz, proiektuko edozein dokumentu ikustea lortu daiteke: memoria, eranskinak, posterra, barne kudeaketarako dokumentuak etab. Gainera, webguneak CCII-N2016-02 estandarra betetzen duenez, irakurle adituak oso azkar identifikatuko du behar duen dokumentua. Behin menuko aukera batean klikatzen denean, zati nagusian PDF formatu gisa irekiko da dokumentua.

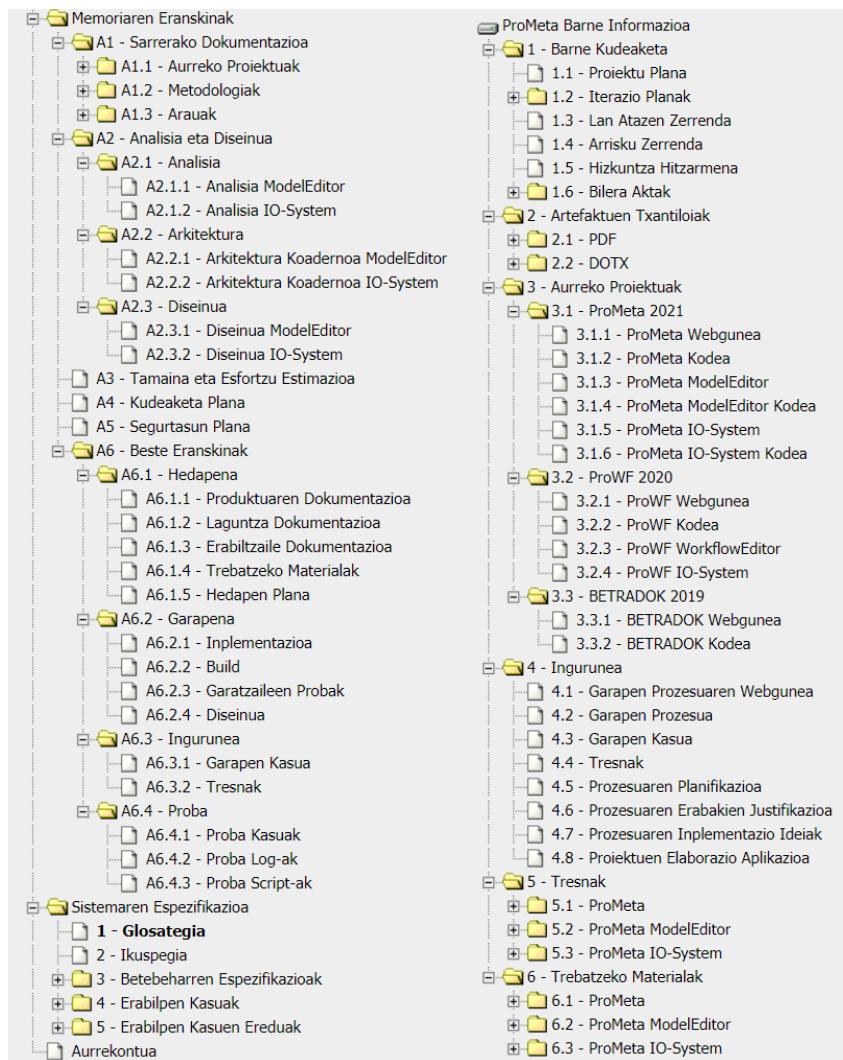
Adibidez, 1.1. Irudian memoria bistaratzeko eskuineko aldean. Memoriako ataletara nabigatu daiteke menuko aukerak erabiliz edo PDF dokumentuko menua erabiliz. PDF dokumentuko menuan maila guztietako izenburuak agertzen dira, aurkibide orokorrean bezala.



1.1. Irudia. Proiektuaren webgunean memoria bistaratu.



Esan bezala, memoriaz gain dokumentu eta esteka asko daude webgunean. 1.2. Irudian menuko dokumentu gehienak ikus daitezke. Ezkerrean memoriaren eranskinak, sistemaren espezifikazioa eta aurrekontua daude, CCII arauak eta OpenUP metodologiak zehazten dituenak. Eskuinean proiektuaren barne informazioa agertzen da. Bertan proiektuaren barne kudeaketa, txantiloik, aurreko proiektuak, ingurunea, tresnak eta trebatzeko materialak daude.



1.2. Irudia. Webguneko eranskinen eta barne informazioaren menuak.

2 Helburuak

Izenburuak dioen moduan, ProMeta **Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema** da. Beraz, proiektuak bi helburu nagusi ditu: garapeneko prozesuen definizioa eta prozesuaren ezarpena.

Lehenengo helburuan, softwarearen garapeneko prozesuak definitzeko metaeredu bat definitu beharko dugu eta gutxienez OpenUP metodologiaren eredua. Eredua aldatu ahal izateko, editore grafiko bat eta testu editore bat sortuko ditugu. Bi editoreen arteko bateragarritasuna bermatu beharko dugu, edozein momentutan bien artean aldatu ahal izateko.

Bigarrengoa, metodologien ereduak informazioa erabiliz prozesua ezartzen duen webgune bat sortu beharko dugu. Horrek garapen taldeari prozesua jarraitzen lagunduko dio, bakoitzak uneoro egin behar duena argi utziz. Webguneak prozesuen informazioa eta proiektuena bistaratu eta aldatzeko aukera emango du, garapen prozesurako behar den informazio guztia bateratuz.

Enpresa edo garatzaileen ikuspegitik bi dira helburuak. Alde batetik, ekoizpen-prozesu sistematiko bat izatea, ezinbestekoak etengabeko hobekuntza gauzatu eta kalitatezko produktuak sortzeko. Bestetik, ekoizpen-prozesu hori sistematizatzeko baliabideak metodologia, arau eta estandarretatik ateratzea. Beraz, ekoizpen-prozesuko artefaktuak berrerabili ahal izateko azpiegitura teknologiko bat sortu beharko dugu.

Informatika Ingeniaritza Graduko ikasle bezala, Software Ingeniaritzan espezializatuta, lan honen egilearen helburua proiektuaren motibazioarekin bat dator: software garapenerako garrantzitsuak diren aspektuak bereganatzea. Adibidez, gaur egungo bezeroen eskakizun ez-funtzionalak betetzea, softwarearen kalitateari dagozkionak.

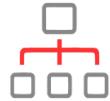
Proiektuaren helburuak hobeto ulertzeko, komenigarria da hiru aspektu zehaztea: arazoa, produktua eta interesatuak. Ondorengo azpiataletan deskribatuko dira horiek taulen bidez.

2.1 Arazoaren Planteamendua

Proiektuaren testuingurua ulertu ahal izateko, lehenik konpondu nahi dugun arazoa ulertu behar da. 2.2. Taulan agertzen den informazioa jakin behar dugu: arazoa zein den, nori eta nola eragiten dion eta gure soluzioa zein den.

Arazoaren Deskribapena	Proiektu informatikoen elaboraziorako ezagutza erabilgarria pilatzen da zenbait jarduera modu automatikoan egiteko, adostutako gida bat jarraituz. Gida hauek softwarea garatzeko prozesuetan, metodologietan eta estandarretan oinarritzen dira. Beharrezkoa da gida hauek definitu eta kudeatu ahal izatea, mantendu eta hobetu ahal izateko. Horrek giden edukia eta horiek exekutatzen dituen sistema definitzea eskatzen du.
Interesatu Kaltetuak	Proiektu informatikoak garatzen dituzten erakundeak.
Arazoaren Eragina	Estandarizazio maila baxua software proiektuen garapen prozesuan. Desadostasunak, atzerapenak eta akatsak eragiten ditu honek.
Soluzioaren Abantailak	Softwarea garatzeko prozesua definitu, kudeatu, mantendu eta hobetzeko erraztasuna. Garapen denbora asko gutxituko du eta bizi-ziklorako metodologia bat erabiliz estandarizazio-maila handituko du.

2.1. Taula. Arazoaren Planteamendua.



2.2 Produktuaren Planteamendua

Arazoa ulertu ondoren, produktuaren inguruko planteamendua azalduko dugu 2.2. Taulan. Bezeroa eta beharra, gure produktua, alternatibak eta horiekiko abantaila eta hobekuntzak zein diren jakin behar dugu.

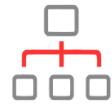
Bezeroa	Software proiektuak garatzen dituen enpresa.
Beharra	Softwarea garatzeko prozesuaren bizi-zikloa definitu eta exekutatzen duen sistemaren beharra, prozesua mantendu eta hobetu ahal izateko.
Produktua	ProMeta: Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema.
Abantaila	Softwarea garatzeko prozesua definitu, kudeatu, mantendu eta hobetzeko erraztasuna.
Alternatiba	ProWF: Software proiektuen elaboraziorako workflowetan oinarritutako sistemaren sorkunza eta bizi-zikloa definitzeko metodologia baten ezarpena. Ez ditu OpenUp bizi-zikloaren fase guztiak, hasiera eta elaborazio fasearen zati bat bakarrik.
Hobekuntzak	ProMeta sistemak metaereduak erabiltzen dituenez erraztasuna ematen du aldaketak egiteko. Etorkizunean sistemaren atal bat aldatzea erabakitzera, metaeredua edo ereduak aldatzea nahikoa da. OpenUp metodologiarren bizi-zikloa osatzea du helburu. IO-System sistemaren editorea hobetzea ere aurreikusten da. Gainera, sistema zerbitzari batean jarriko da, edozein erabiltzailerentzat eskuragarri egon dadin.

2.2. Taula. Produktuaren Planteamendua.

2.3 Interesatuen Deskribapena

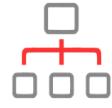
Arazoa eta produktuaren planteamendua jakinik, proiektuaren interesatuak eta horien erantzukizunak deskribatzen dira 2.3. Taulan.

Izena	Deskribapena	Erantzukizunak
Software Enpresa	Software proiektuak sortu eta sortutako proiektuaren partaideak aukeratzen ditu.	Proiektua sortu Partaideak aukeratu
Administratzailea	Sistema kudeatzeaz arduratzen da.	Erabiltzaileen kontuak kudeatu Sisteman aldaketak egin
Proietkuko Kudeatzailea	Software proiektuaren liderra da, interesdunen arteko harremanak koordinatzen ditu, eta proietkuko taldea helburuetan enfokatuta egotea du ardura.	Proiektuaren ardurak eta rolak esleitu Prozesuaren gida automatikoak jarraitu
Proietkuko Analista	Software proiektuaren eskakizunak eta betebeharrok harrapatzen ditu.	Proiektuaren eskakizunak eta beharrak eskuratu



Izena	Deskribapena	Erantzukizunak
		Prozesuaren gida automatikoak jarraitu
Proiekutuko Prozesu Ingeniarria	Software proiektuaren eskakizunak eta betebeharrok aztertuz, prozesuak sortzen ditu.	Prozesua sortu/editatu, metaeredu eta ereduetan aldaketak eginez. Prozesuaren gida automatikoak jarraitu
Proiektuaren beste taldeklideak	Software proiektuaren arkitektoak, garatzaileak, testerrak...	Prozesuaren gida automatikoak jarraitu

2.3. Taula. Interesatuen deskribapena.



3 Aurrekariak

Kapitulu honetan, proiektua burutu ahal izateko garrantzia izan duten iraganeko elementu esanguratsuak jasotzen dira. ProMeta proiektua ProWF proiektuaren jarraipena denez, aurrekariak antzekoak dira eta aurreko proiekutan oinarrituta daude.

Hasteko, software kalitatea eta gaur egungo bezeroen eskakizun gogorrak aztertuko dira. Ondoren, bezeroen eskakizunengatik agertu diren beharrak azalduko dira: softwarea garatzeko metodologiak eta proiektuak aurkezteko araua. Hori azaldu eta gero, *BPM* sistemek eskaintzen dituzten abantailak eta sortzen dituzten menpekotasunak aztertuko dira. Jarraian, inferentzia motorrak eta haien ahalmena komentatuko da, *CLIPS* eta *EHSIS* inguruneak aztertuz. Segidan, *MDE* softwarea garatzeko metodologia azalduko da eta *CMS* softwareak deskribatuko dira. Amaitzeko, ProWF sistemaren ezaugarriak aztertuko dira, aurretik azaldutako kontzeptuak kontuan hartuz.

3.1 Software Kalitatea

Softwarearen kalitatea softwarearen ezaugarriak kontrolatzean eta ziurtatzean datza, bezero eta erabiltzaileen onurarako. Softwarea produktua ez da fabrikatzen eta ez da fisikoki degradatzen, baina garapen-prozesu bat du. Hala ere, softwarea ez da ia inoiz perfektua izaten. Proiektu ororen helburua ahalik eta kalitate oneneko softwarea ekoiztea da, bezero eta erabiltzaileen itxaropen eta betekizunak gaindi ditzan. Softwareak akatsak eta gorabeherak izan ditzake, baina ez dira edozein ekipo fisikoren antzekoak, horietan ausazko hutsegiteak eman baitaitezke. Softwarearen kasuan, denak sistematikoak dira eta, ondorioz, zuzendu beharrekoak.

Softwarearen ziurtagiria bere kalitatea ziurtatzeko prozesuaren ondorioa da, baina ez da inoiz azken helburua. Softwarearen kalitatea ez da ziurtatzen, bermatu behar direnak kalitatezko softwarea eraikitzeko prozedurak dira. Procedurak, bezeroek eskatutako kontsentsu maila altuko nazioarteko estandarretan oinarritutakoak izan behar dute eta, herrialde bakotzean, normalizazio agentzia ofizialaren berariazko lan-taldeak egokitutakoak. Proceduren kalitate-eredu ezberdinak daude, garrantzitsuenak *CMMI*¹ eta *SPICE*² dira, helburu berdina dutenak. Kalitate-ereduek, softwarea garatzeko praktika onenak definitzen dituzte, softwarea garatzen duten erakundeen prozesuak hobetzen laguntzeko. Ziurtagiri-emaileek, praktika on horiek hartu eta beraien emaitza neurgarriak egiaztatzen dituzte garatzailekin lankidetzan. Ebaluazio-prozesu batean parte hartu ahal izateko inplikatu guztiak ziurtagiri egokiaren jabe izan behar dira.

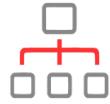
3.2 Bezeroen Eskakizun Gogorrak

2007ko urriaren 30ean onartutako *Ley de Contratos del Sector Público* legeak, **BOE-261-2007-18874**, bere **69. artikuluan**, hornitzaleek kalitate bermeak erakusteko arau europarrei erreferentzia egin eta, dagozkien ziurtagiriak arauarekin bat datozen erakundeek emanak izan behar dutela ezarri zuen.

Ondoren, aurreko legearekin bateratuta, 2011ko azaroaren 14an Estatuko Aldizkari Ofizialean onartu zen legearen ostean, **BOE-A-2011-17887**, bezeroen eskakizunak sendo gogortu ziren, softwarearen kalitateari dagokionez. Lege horren **80. artikuluak** kalitatea bermatzeko arauak betetzen direla egiaztatzea zuen helburu, horretarako erakunde independenteek emandako ziurtagiriak beharrezkoak ziren merkatuak exijitzen bazituen. Erakunde horiek kalitatea bermatzeko Europako arau jakin batzuei erreferentzia egin

¹ *CMMI*: Sigla(ingeles), Capability Maturity Model Integration. Software-sistemak garatzeko, mantentzeko eta erabiltzeko, prozesuak hobetzeko eta ebaluatzen eredu da, *CMMI* Instituatuak administratutakoa.

² *SPICE*: Akronimoa(ingeles), Software Process Improvement and Capability dEtermination. ISO/IEC 15504. Garapen-prozesuak hobetzeko, ebaluatzen, informazio-sistemak eta software-produktuak mantentzeko eredu da.



edo baliokideak izan behar ziren. Hortaz, lege horren ostean software garapenean kalitatea bermatzeko *CMMI* edo *SPICE* erakundeei lotutako ziurtagiriak lortzea beharrezkoa bilakatu zen.

CMMI erakunde baten softwarea garatzeko prozesuaren heldutasuna ebaluatzeko eta neurtzeko metodoak erabiltzen dituen prozesu bat da. Erakundeen heldutasun-maila neurtzeko bost etapa bereizten ditu.

Maila	Etapa	Prozesuaren egoera
I	Hasierakoa	Kaotikoa da, kontrol zaila du, errektiboa da
II	Kudeatuta	Proiektuen ezaugarriak ditu, askotan errektiboa da
III	Definituta	Erakundeen ezaugarriak ditu, proaktiboa da
IV	Kuantitatiboki kudeatuta	Erdi-mailako prozesua da, kontrolatuta dago
V	Optimizazioa	Etengabeko hobekuntzan dago

3.1. Taula. CMMI-ren 5 heldutasun-mailak.

*CMMI*ren 3. heldutasun-maila lortzea ezinbesteko zen merkatuan mantentzeko, hori dela eta, hainbat enpresa kexatu ziren merkatutik kanpo geratzeagatik, baina alferrikakoa izan zen. Horren adibidea, 2016an *OESIA NETWORKS, S.L.* enpresak jarritako erreklamazioa da, *Recurso 0006-2016*, honek *CMMI*ren 3. heldutasun-maila lortzean zegoela erreklamatzen zuen merkatuan mantentzeko, tramitazioren faltan zegoen. Hala ere, administrazioaren erantzuna errekurtsoa baistearen aurka egotea izan zen, lehen aipatutako 80. artikulua ez betetzeagatik. Arrazoia enpresak oraindik ez zuela eskatutako kalitate maila bermatzen zuen ziurtagirik edo ziurtagiriaren baliokiderik izan zen.

Evidentzia horiek direla eta, gaur egun proiektu informatiko baten bezeroek dute pisu handiena edo agintea, haien exijentzia gogorrak betetzea funtsezkoa da merkatuan mantentzeko. Bezeroen behar eta ametsetan fokatzea, 2018ko *CMMI 2.0* bertsioan eta estandar internazionalen sendoki indartuta geratu da.

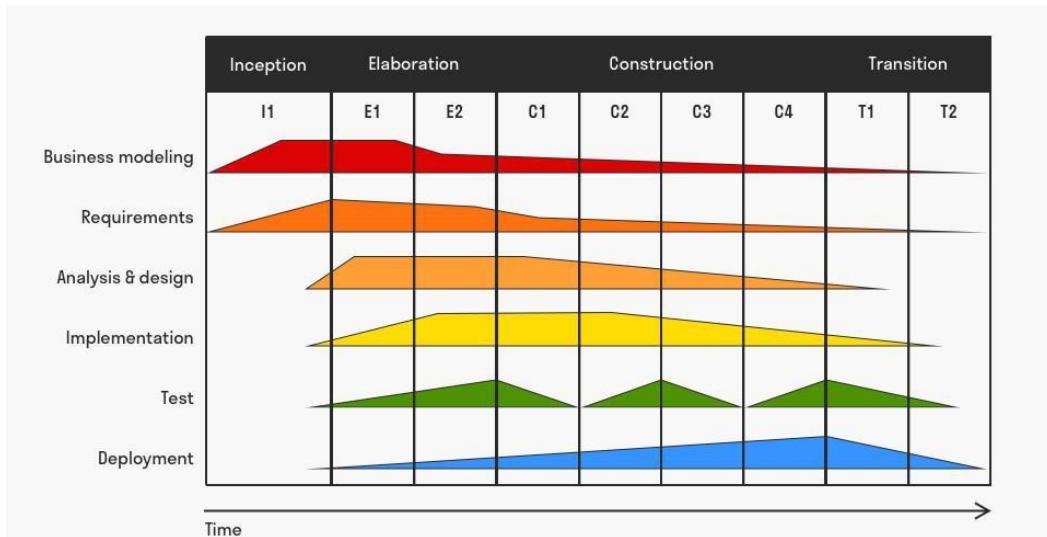
3.3 Metodologiak

Gaur egun pentsaezina bilakatu da software proiektu bat aurrera ateratzea metodologiarik jarraitu gabe. Software munduan artisautza lanak ez du etorkizunik, metodologia baten ezarpenak hori ekiditeko balio du. Software ingeniariek edozein momentutan jakin behar du zer egin, noiz eta nola, bestela arazoak eta galerak agertzenko probabilitatea handituz joango da proiektuaren garapena luzatzen doan heinean.

Gainera, proiektuak zeroztik egitea garestiegia da software garapen enpresentzat. Berrerabilpenean oinarritutako metodologia bat aukeratuz, enpresaren kostu ekonomikoak gutxituko dira epe laburrean.

Gehien erabiltzen direnak metodologia arinak dira. Arina kontzeptua softwarea sortzeko urratsak arindu behar direlako sortzen da. Giza interakzioetan zentratzen da, aldeen arteko elkarritzeta-fluxuari eutsi ahal izateko, garapen dinamikoagoa eta parte-hartzaleagoa ahalbidetzeko. Metodologia bizkorrek garapen-sistema egokitzaile bat erabiltzen dute, eta ez prediktiboa. Horrek esan nahi du lantaldeak buruan duela nahi duen emaitza, baina ez daki zehatz-mehatz zer produktu mota sor dezakeen.

Software garapenaren metodologia arinen artean RUP³ aurkitzen da. Softwarea garatzeko prozesu iteratibo bat da, Rational Software Corporation erakundeak sortua, IBM⁴ enpresaren dibisio bat. 3.1. Irudian ikus daiteke RUP prozesuaren egitura. Hala ere, RUP ez da zehatz-mehatz jarraitu behar den prozesua, baizik eta prozesu moldagarria da, garapen-erakundeek eta software-proiektuen taldeek egokitzeko asmoarekin, bakoitzaren premietarako egokiak diren elementuak hautatuz.

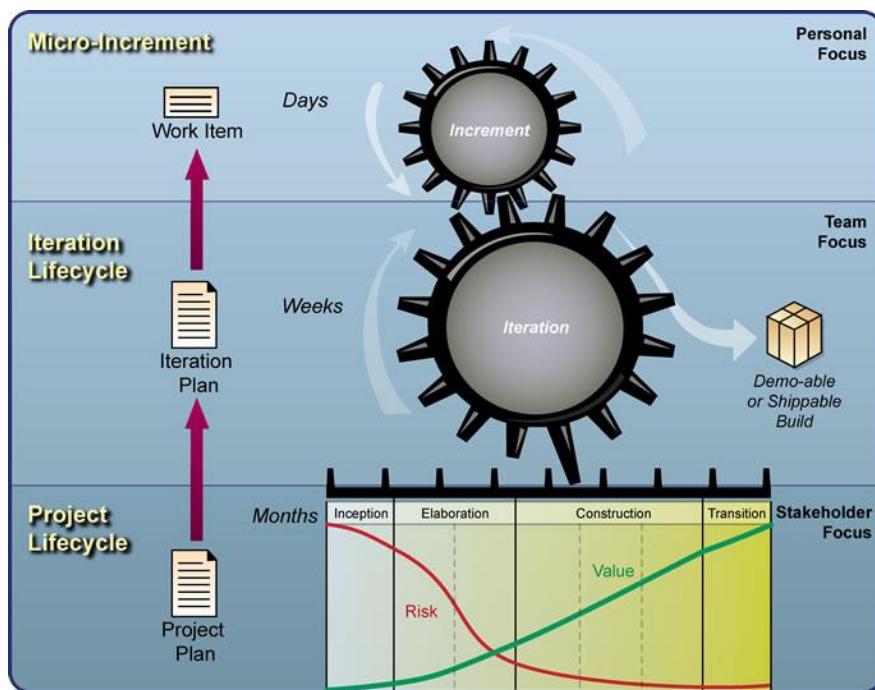


3.1. Irudia. RUPen prozesu iteratiboa, bizi-zikloaren faseak, iterazioak eta jarduerak.

RUPen oinarritutako metodologia simple eta erabiliena OpenUP da. Metodologia horrek RUPen funtsezko ezaugarriak gordetzen ditu, garapen iteratiboa, erabilpen-kasuak, arriskuen kudeaketa eta arkitekturan oinarritutako ikuspegia bultzatzen duten agertokiak barne. RUPen erabiltzen ez diren aukerako zati gehienak baztertu eta elementu asko bateratzen ditu. Emaitza prozesu askoz simpleagoa da, eta RUP printzipioekiko leiala izaten jarraitzen du. 3.2. Irudian ikus daiteke OpenUP prozesuaren laburpen bat.

³ RUP: Sigla(ingelesez), Rational Unified Process. Rational Software enpresak garatutako software-prozesu bat da. Objektuetara bideratutako sistemak aztertu, diseinatu, impletatatu eta dokumentatzeko erabilitzen den metodologia estandarra.

⁴ IBM: Sigla(ingelesez), International Business Machines. Informatikarekin lotutako tresnak, programak eta zerbitzuak ekoitzu eta merkaturatzen dituen enpreta multinazionala da.



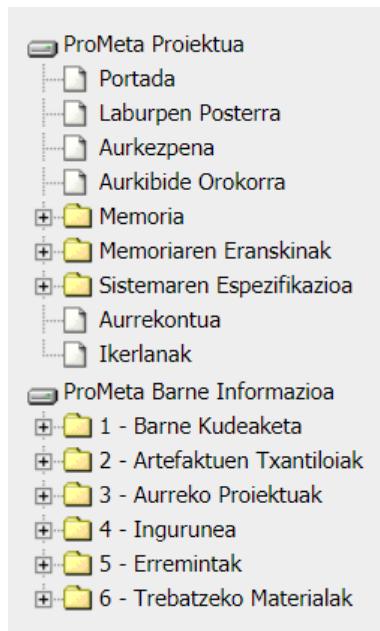
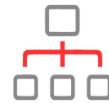
3.2. Irudia. OpenUP-en geruzak: mikro-gehikuntzak, iterazio bizi-zikloa eta proiektu bizi-zikloa.

3.4 Arauak

Proiekturen garapenean zehar sortzen diren dokumentazio multzoaren aurkezpenerako ezinbestekoa da ezarrita dagoen araudi ofiziala betetzen dela bermatzea, nazioarteko estandarren oinarritutakoa. Helburu nagusia proiektuan parte hartzen duten alderdi guztien aldeko dokumentazioa osoa eta gardena izatea izanik, bezeroaren gogobetetasuna handituz.

Espanian eta nazioartean araurik finkatuenak UNE⁵ eta CCII erakundeek aurkeztutakoak dira: **CCII-N2016-02. Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática**. 3.3. Irudian ikusten den moduan, zehatz-mehatz deskribatzen du nola egin behar den Informatika Ingeniaritzako Proiektu baten dokumentu-zehaztapena. Proiektuko dokumentazioa egiteko ereduak eta dokumentazioaren antolamendua deskribatzen du ere. Erreferentiazko nazioarteko esparru eta standarrak kontuan hartzen ditu, hala nola UNE 157801:2007 – "Informazio-sistemen proiektuak egiteko irizpide orokorrak"; UNE-ISO 21500:2013 – "Proiektua zuzentzeko eta kudeatzeko jarraibideak" eta PMBOK – "Proiektuen Zuzendaritzarako Oinarrien Gida".

⁵ UNE: Sigla(gaztelera), Una Norma Española. Comités Técnicos de Normalización (CTN) batzordeak sortutako arauen, arau esperimentalen eta txostenen (estandarrak) multzoak dira.



3.3. Irudia. CCII-2016N-02 araua betetzen duen proiektuaren webgunearen egitura.

Proiektuak aurkezteko araudien gabeziak proiektu informatiko batean gatazkak ekarri ditzake. Hau da, proiektuan esku hartzen duten aldeentzat nahi ez diren ondorioak eragin, bezero, hornitzairen zein interesdunen arteko gatazkak sortuz.

3.5 BPM

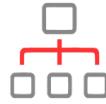
Business Process Management negozio-prozesuak definitzera eta gauzatzera bideratutako metodologia edo ikuspegি estrategiko gisa ikus daiteke. Negozio-prozesuak konplexuak eta dinamikoak dira. Gainera, malguak izan behar dute, negozioa nabarmen aldatzen delako eta etengabe eguneratu behar direlako.

Hona hemen BPMrekin lan egitean lortzen diren abantailak:

- Ataza errepikakorrik ezabatzea eta automatizatzea.
- Eraginkortasuna handitzea: prozesuetan akatsak minimizatuz, itxarote-denbora murriztuz, giza esku-hartzeak murriztuz eta lana berregitea saihestuz
- Negozio-arauak betetzen direla ziurtatzea.
- Zerbitzu-maila bermatzea, salbuespenak maneiatur, egoeren jarraipena eginez, gertakariak mailakatuz, prozesuen sendotasuna eta trazabilitatea bermatuz, etab.
- Lan egiteko modua aldatzeko aukera eskaintza, eragiten duen inpaktua murriztuz eta etengabe hobetuz.

Gartner eta *Forrester* aholkularitza-enpresek BPM merkatua ikertzeko, urtero, txosten bat argitaratzen dute. Txosten horiek patentatutako datu kualitatiboak aztertzeko metodoetan oinarritzen dira, merkatuaren joerak frogatzeko, hala nola zuzendaritza, heldutasuna eta parte-hartzaileak.

Gartnerren Koadrante Magikoa industria teknologikoaren azpisektore nagusiei buruz egiten duen merkatu-ikerketa bati erantzuten dio. Bertan, merkatu-joerak, tartean dauden enpresa nagusiak eta heldutasun teknologikoa besteak beste aztertzen dira. 3.4. Irudian lau koadrantetan banatutako grafiko gisa aurkezten da. X ardatzak exekutatzeko gaitasuna adierazten du, eta Y ardatzak, berriz, balio-proposamen osoa. Ezkerretik eskuinera eta goitik behera, sektore bakoitzeko enpresak honako koadrante hauetan kokatzen dira: *challengers*, *leaders*, *niche players* eta *visionaries*. Antzeko informazioa aurkitu dezakegu 3.5. Irudian agertzen den *Forrester Wave* txostenean. Bi txosten ezagun hauek, ondo kokatutako hornitzairek komertzialki erabiltzen dituzte.



3.4. Irudia. 2018ko BPM Suiten Koadrante Magikoa, Gartner.

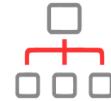


3.5. Irudia. 2019ko The Forrester Wave txostena, Forrester.

Bizagi bi txostenetan agertzen da, ondo kokatuta. *Bizagi* bezalako teknologia erabiliz arkitektura konplexuko web-aplikazioa implementatu daiteke, prozesu, erregela eta workflowentzako motorrarekin barne. Apustu hori garestia izan daiteke eta proiektu edo enpresaren menpekotasun teknologikoa handituko da. Beste apustu merkeago bat, metodologia eta estandarretara hurbiltzen joateko, teknologia propioa sortzea da, workflow-lengoia propioa sortuz eta lengoia horrekiko inferentzia motorra eraikiz.

3.6 Inferentzia Motorrak

ProWF projektuan teknologia propioa sortzea erabaki zen, workflow-lengoia propioa sortuz eta lengoia horrekiko inferentzia motorra eraikiz.



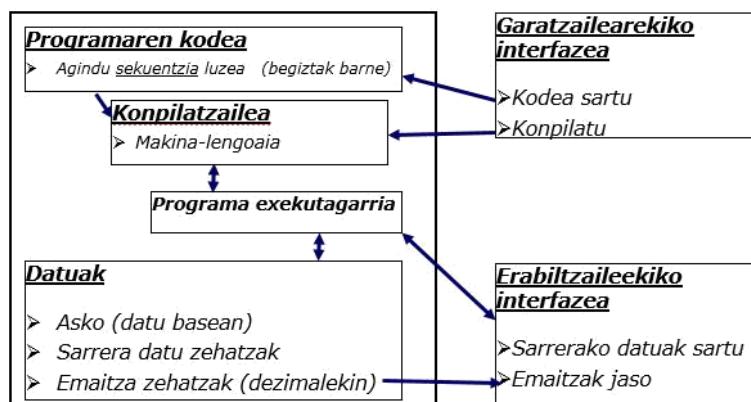
CLIPS (C Language Integration Production System) sistema adituak ekoizteko eta exekutatzeko garapen-ingurunea ematen duen tresna da, *NASA* sortu eta jabari publikoan utzi zuena. Bere lengoaiak erregelak, objektuetara bideratutako programazioa eta programazio prozedurala erabiltzen du jakintzak adierazteko. Lengoia simple baino oso ahalsua da, hurrengo ezaugarrietan nabarmenduz:

- **Garraigarritasuna.** C lengoaiak ematen diona.
- **Integrazioa edo zabalkortasuna.** Programazio prozeduralari esker funtzi berri ahalsuak sor daitezke. Service-oriented Architecture (SOA) estandarra erabilita BPM edo beste edozein sistemekin integragarria da.
- **Interakzio edo disziplinarteko garapena.** Formakuntza ezberdinako pertsonen ideiak azkar implementatzeko aukera lengoia simple eta ahaltsu bat erabilita, objektu eta erregeletan oinarrituta.
- **Egiaztapen edo balidazio errazagoa.** Horretarako, funtzi bereziak erabiliz.

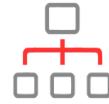
EHSIS, ordea, Euskal Herriko Unibertsitateko (EHU) *ERABAKI* taldeak hedatutako ingurunea da, *CLIPS 6.04*, *FuzzyCLIPS¹⁴ 6.04*, objektuetara eta gertaeretara bideratutako programazioa, interfazeen garapena, komunikazioa eta leihotan oinarritutako ingurunea integratzen duena. *EHSIS* inguruneak (ikus 3.6 irudia), softwarea garatzeko baliabide tradizionalen oinarritutako aplikazioak sortzeaz gain (ikus 3.7 irudia), *COOL¹⁵* lengoia erabilita, ezagutzan oinarritutakoak ere sor ditzae sistemaren arkitektura egoki batekin baliatuz (ikus 3.8 irudia).



3.6. Irudia. EHSISen garapen ingurunea.



3.7. Irudia. Programazio tradizionalaren eskema.



Erregelak (baldintzak + ekintzak):

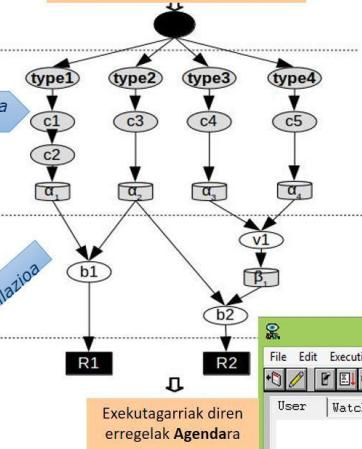
```
(defrule R1
  (type1 c1 c2)
  (type2 c3)
=>
  Ekintza)

(defrule R2
  (type2 c3)
  (type3 c4 v1)
  (type4 c5 v1)
=>
  Ekintza)
```

RETE zuhitzaren

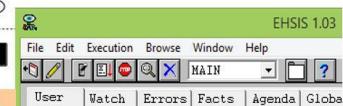
- Klaseak
- Objektuak
- Egitateak

Ebidentzia aldaketaren seinalea



Egitateak edo
ebidentziak:

```
(deffacts egitateak
  (type1 c1 c2)
  (type2 c3)
  )
```



3.8. Irudia. Erregeletan oinarritutako sistemen eskema: klaseak, objektuak, erregelak eta egitateak.

Erregelak idazteko lengoia simplea denez, klase eta objektuak erabilita ere, domeinuko ezagutza adierazteko ondo diseinatutako lengoia bat lortuz gero, ezagutza hori exekutatzeko kodea automatikoki sortzea ere posible da. Gainera, inferentzia motorrarekin abiadura handiko exekuzio eraginkorra lortu daiteke kode guztia RETE sare bezala gordeta, hau “konpilazio” mota bat bezala erabili daiteke. RETE sarearen egitura bereziak RETE algoritmoaren abiadura, egitate eta erregela askorekin ere, aprobetxatzeko aukera ematen du. Erreminta hau, bere eraginkortasuna eta jabari publikoa dela eta, konplexutasun handiko proiektuetan erabili daiteke produktu mantengarriak lortzeko.

EHSISen garapen inguruneak badu bere produkzio bertsioa, *EHSIS_RT* deitutakoa. Web-zerbitzuetarako bertsioa ere badu, Mairi deitutakoa.

3.7 MDE

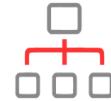
ProMeta proiectuan MDE erabili da ModelEditor azpisistema garatzerakoan.

Model Driven Engineering (MDE) edo eredu bidezko ingeniaritzako softwarea garatzeko metodologia da. Domeinu ereduak erabiltzen ditu, hau da, arazo zehatz batikotik lotutako gai guztien eredu kontzeptualak. Hori dela eta, aplikazioen domeinu jakin baten ezagutzaren eta jardueren irudikapen abstraktuak nabarmentzea du helburu, kontzeptu informatikoetan sartu gabe.

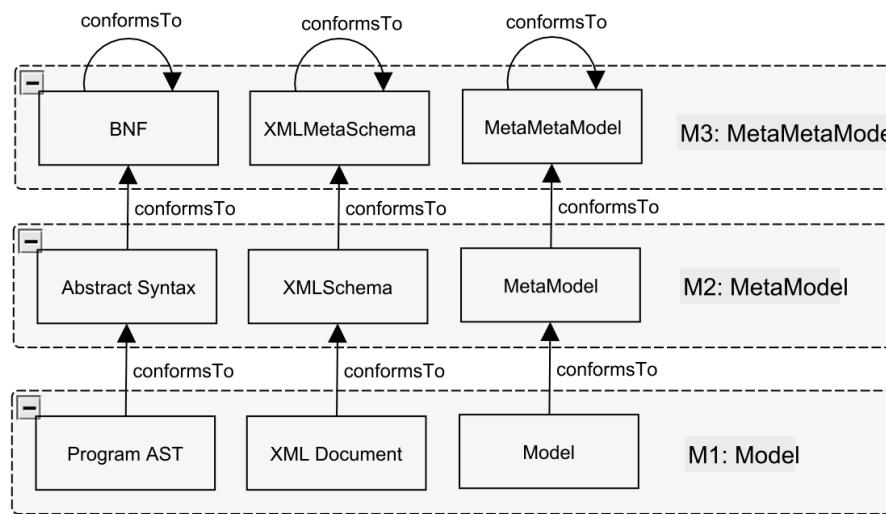
MDEren helburua produktibitatea handitu handitzea da. Horretarako, sistemen arteko bateragarritasuna maximizatzen du eredu estandarizatuak berrerabiliz, diseinu prozesua simplifikatu aplikazioaren domeinuko diseinu patroi erreplikakorren eredu bidez eta sisteman lan egiten duten pertsonen eta taldeen arteko komunikazioa sustatu praktika onen estandarizazioaren bidez.

MDEren modelatze paradigma bat eraginkorra dela deritzo bere ereduek domeinua ezagutzen duen erabiltzaile baten ikuspuntutik zentzua badute eta sistemak oinarri gisa balio badute. Eredua produktuen kudeatzaileen, diseinatzaileen, garatzaileen eta aplikazioaren domeinuko erabiltzaileen arteko kolaborazioarekin garatzen dira. Eredua amaitzen doazen heinean, softwarea eta sistemak garatzea ahalbidetzen dute.

MDEn 4 modelatze maila daudela esan dezakegu. Maila handiagoa den heinean abstrakzio-maila igotzen doa. 3.9. Irudian ikus daitezke 3 abstrakzio-maila altuenak: eredua, metaeredua eta meta-metaeredua.



- **M0. Instantziak.** Maila hau exekuzio sistemari dagokio. Maila honetan negozioko elementuak daude, edo mundu errealeko elementuen errepresentazioak (software errepresentazioak).
- **M1. Eredua.** Eredu honek software sistemaren itemak errepresentatzen dituzten kontzeptuak dauzka. M1 mailan dauden kontzeptuek M0 mailan dauden instantziak kategorizatu edo sailkatzen dituzte.
- **M2. Metaeredua.** M1 mailan dauden kontzeptuen inguruan arrazoitzeko beharrezkoak diren kontzeptuak dauzka. M2 mailako elementu batek M1-eko elementuak espezifikatzen ditu. Ereduen eredu honi metaeredu esan ohi zaio.
- **M3. Meta-metaeredua.** M2 mailan dauden kontzeptuen inguruan arrazoitzeko beharrezkoak diren kontzeptuak dauzka. M3 mailako elementuek M2-ko elementuak kategorizatzen ditu. Meta-metaeredu esan ohi zaio.



3.9. Irudia. MDEren 3 abstrakzio maila: eredu, metaeredua eta meta-metaeredua.

3.8 CMS

ProWF eta ProMeta projektuen IO-System sistema garatzeko Drupal CMSa erabili da.

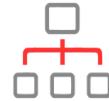
Content Management System dokumentuak eta bestelako edukiak antolatu eta kudeatzeko softwarea da. Askotan web aplikazio bat izaten da askotan, webguneak eta web edukiak kudeatzeko balio duena. Gaur egun sistema asko daude arlo honetan, bai kode irekikoak eta baita jabedunak ere.

Edukiak kudeatzeko sistema zerbitzarian kokaturik egon ohi da. CMS batera atzipena erabiltzaile motaren araberako maila ezberdinetan egin daiteke. Adibidez, edukiaren sortzaileek sistema osatuko duten dokumentuak sortuko dituzte. Argitalpen-teknikariek dokumentu hauek aipatu, onartu edo baztertuko dituzte. Editoreen burua gure web orrian eduki hori argitaratzeaz arduratuko da. Dena interfaze grafiko baten bidez kontrolatuko da, honen erabilera erraztu ahal izateko.

3.9 ProWF

Esan bezala, ezinbestekoa da ProWF sistema ulertzea proiektu hau horren jarraipena delako. Horretarako, bertan zehazten diren rolak, sistemaren osagaiak eta azpisistemak deskribatuko dira.

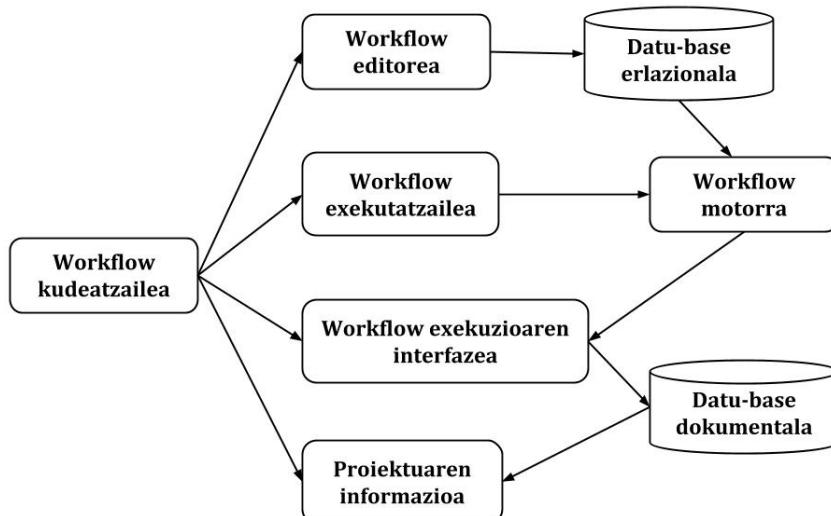
ProWF sistemaren erabiltzaileek hurrengo rol hauekin lan egin behar dute:



- **Prozesu Sortzailea:** sortutako workflow-lengoia baliatuz, softwareen garapenerako bizi-zikloa ezartzen duen metodologia definituko duena.
 - **Analista:** sortutako workflow-ereduak adierazten duen bizi-zikloa exekutatzeaz eta aplikatzeaz arduratuko da.
 - **Projektuko Zuzendaria:** projektuak sortu eta projektuko partaideen rolak esleituko ditu.
 - **Kalitate Arduraduna:** workflowaren exekuzioan zehar sortuko diren artefaktuen kalitatea bermatzea du helburu, balorazioak emanet.

ProWF sistemak gutxienez hurrengo osagaiak behar ditu proiektuaren helburu eta behar guztiak betetzeko. 3.10. Irudian osagai horien arteko erlazioak ikus daitezke.

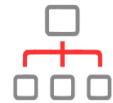
- **Workflow editorea:** sortutako workflow-lengoia erabiliz workflow-ereduak sortzeko balio du, ondoren, eredu horiek workflow motorrak exekutatzeko eraldatuko ditu.
 - **Workflow motorra:** bere lengoaiara eraldatutako workflow-ereduak goitik behera exekutatu eta datu-base erlazionalean workflowaren informazioa gordeko duen inferentzia motorra da.
 - **Workflow exekutzailea:** workflow motorrari aginduak eman eta bere irteerak jasoko ditu.
 - **Workflow exekuzioaren interfazea:** erabiltzaileak exekuzioan zehar ikusiko duena da, erabiltzailearen sarrera/irteerak maneiatzeko balio du.
 - **Datu-base erlazionala:** workflow-ereduaren informazioa biltegiratuko du.
 - **Datu-base dokumentala:** workflowaren exekuzioan zehar bete behar diren artefaktuak biltegiratuko ditu.
 - **Proiektuaren informazioa:** proiektuaren informazioa arau konkretu bat (CCII-N2016-02 edo berriago bat) betetzen duen webgune baten integratuko du, informazioa datu-base dokumentaletik eskuratuz.
 - **Workflow Kudeatzailea:** aurreko osagai guztiak integratuko ditu.



3.10. Irudia. ProWF sistemak behar dituen osagaia eta haien arteko erlazioak.

ProWF sistema bi azpisistema ezberdinetan bananduta egongo da:

- **Workflow Editor:** sortutako workflow-lengoia erabiliz workflow-ereduak sortzeko balio du, ondoren, eredu horiek workflow motorrak exekutatzeko eraldatuko ditu. Testu-editore bat izango da. Rolak: Prozesu sortzailea.
 - **IO-System:** workflow kudeatzaileraren sarrera/irteerak maneiatzea du helburu. CMS baten bitartez kudeatutako web-aplikazioa izango da. Rolak: Projektuko zuzendaria, analista eta kalitate arduraduna.



4 Egungo Egoera

Kapitulu honetan, proiektuaren egungo egoera deskribatuko da, ProWF aurreko proiektuaren egoera eta egilearen prestakuntza azalduz.

4.1 Deskribapena

Egungo egoera ulertzeko, ezinbestekoa da proiektu honen aurrekaria den ProWF proiektuaren egoera ulertzea. Proiektuaren izenburuak dioen moduan ProWF sistemak *software projektuen elaboraziorako workflowetan oinarritutako sistemaren sorkuntza eta bizi-zikloa definitzeko metodologia baten ezarpena* ahalbidetzen du. Sistemak bi osagai nagusi ditu, workflow editorea eta sarrera irteera sistema.

Horretarako, proiektuan zehazten diren abantailak eta desabantailak aztertu eta osatuko ditugu. Gainera, proposatzen diren hobekuntzak ere zehatzuko ditugu, eta gure ideiekin osatu. Izan ere, hobekuntza horietako batzuk aurrera eramango dira ProMeta proiektuan.

4.2 Abantailak eta Desabantailak

ProWF proiekuan sistemaren abantaila eta desabantaila batzuk identifikatu ziren. Osatu egin dira proiektua aztertzerakoan identifikatu diren beste batzuekin. Garrantzitsua da hauek kontuan hartzea ProMeta proiektuaren planteamendua egiteko.

4.2.1 Abantailak

Sistemaren abantailen artean bi mota aurki ditzakegu, workflow-lengoaiari lotutakoak eta workflowetan oinarritutako sistemari lotutakoak.

Workflow-lengoaiari lotutakoak:

- Sortutako workflow-ereduen irudien nabigagarritasunak garbitasuna eta ulergarritasuna ematen dio prozesuari. Gainera, OpenUP metodologiaren webgunean agertzen diren formak eta koloreak erabiltzen ditu.
- Workflow-eredua aldagarria da, baldin eta sortutako lengoia grafikoa errespetatzen bada.
- Lengoaiak softwarearen bizi-zikloaren ezaugarri esanguratsuenak harrapatzen ditu.

Workflowetan oinarritutako sistemari lotutakoak:

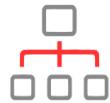
- Interfaze simple eta intuitiboa du, itxura profesionalarekin.
- Drupal CMSari esker, erabiltzaileen erregistro eta kudeaketa erraza du.
- Workflow motorra, workflow-lengoia erabiliz sortutako edozein prozesu exekutatu dezake, prozesuaren objektuak automatikoki sortzen dira eta prozesuak exekutatzeko erregelak berrabili daitezke.
- Workflow motorraren prozesaketa-denbora asko murrizten da, erabiltzen dituen instantzia eta erregelak RETE sarean “konpilatu” izanari esker.

4.2.2 Desabantailak

Sistemaren desabantailak ere multzo berdinatan sailka daitezke.

Workflow-lengoaiari lotutakoak:

- Workflow-eredua nabigagarria denez, hainbat fitxategi eraldatu behar dira CLIPS lengoaiako klase eta instantziak sortzeko. Prozesu errepikakor eta neketsua da.



- OpenUP bizi-zikloko workflow eredua eskuz definitu beharra. Horrek prozesua definitzeko denbora asko behar izatea eragiten du. Hobe izango litzateke webguneko informazioa erabiliz automatikoki sortzea.

Workflowetan oinarritutako sistemari lotutakoak:

- Web-aplikazioak lokaleko instalazioa behar du. Zerbitzari batera eraman daiteke eta horrela instalazio prozesua asko murriztuko litzateke, bakarrik Workflow Editor azpisistemaren osagaia instalatuz.
- Drupalen bidez sortutako web-aplikazioak ez ditu erantzun azkarrak ematen. Gunearen orrialdez aldatzean kargatu behar diren modulu eta beste aspektuek errrendimendua murrizten diote.
- Webguneko editorea testu soilean dagoenez, ez du aukerarik ematen formatua emateko. Adibidez, ezin da letra Iodia erabili eta ez dago taulak betetzeko aukerarik.
- Webgunean artefaktuen informazioa betez ez dira lortzen artefaktuen txantiloiaaren formatua betetzen duten dokumentuak.
- Drupaletik sortzen diren datuak kanpoko datu-base baten daude. Horrek Drupalek eskaintzen dituen aukerak murrizten ditu, datuak bistaratzeko eta editatzeko aukerak, adibidez.

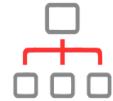
4.3 Proposatutako Hobekuntzak

ProWF proiektuan hobekuntza interesgarri asko proposatzen ziren. Garrantzitsua da horiek kontuan hartzea ProMeta proiektua haren jarraipena baita. Gainera, hobekuntza posible gehiago ere gehitu ditugu aurrekoak osatzeko.

ProWF proiektuan hurrengo hobekuntzak proposatzen dira etorkizunerako:

- OpenUP bizi-zikloko workflow eredua amaitu eta ahal bada, hobetu. Prozesuan gelditzen diren faseak gehitu eta bigarren fasea (*elaboration*) guztiz definitu. Horretarako, “*Workflow-lengoaiaren Eskuliburua*” eta “*Workflow Editor – Eskuliburua*” dokumentuak jarraituz.
- *Workflow*-ereduen eraldaketa-prozesu errepikakorra ekiditeko metaeredu bat definitzea, DOT lengoia deskriptiboaren eta COOL lengoiaaren arteko eredu bat sortuz, urrats bakar baten bidez eraldaketa eginez eta kanpoko softwareak (Gephi, Protégé) erabiltzea ekidituz. Produktibilitatea, azkartasuna eta mantenugarritasuna bilatzu.
- *Workflowak* kudeatzeko sistema zerbitzari batean jartzea. Zerbitzari batean egonda, erabiltzaileak ez du instalaziorik beharko.
- *Workflowak* kudeatzeko sisteman, *IO-System* azpisisteman, artefaktuen sekzioak idazterako orduan *HTML* edo *WYSIWYG*⁶ motako testu-editore bat implementatzea. Softwarearen bizi-zikloa definitzen duten metodologia askotan taulak eta Excel orriak bete behar dira, prototipo honetan, ordea, ez dago taulak txertatzeko aukerarik.

⁶ WYSIWYG: sigla (ingelessez), What You See Is What You Get. Testu-prozesadoreei eta beste testu-editore batzuei aplikatutako esaldi bat da, azkenengo emaitza zuzenean erakutsiz dokumentu bat idazteko aukera ematen duena.



- Bezero ezberdinen eskakizunak asetzeko gaitasuna izateko asmotan, metodologia ezberdinak integratzen dituen garapen-prozesuak definitzea.
- Gure enpresak ondo egiten duena garapen-prozesuan sartzea. Hori CMMI 2.0 kalitate-ereduak eskatzen du. Prozesu berriekin integratzeko lanak konplexuak izan daitezke.
- Garapen-prozesua grafikoki adieraztea xehetasun maila handiagorekin eta funtzionalitate gehiagorekin. Lehen fase batean, lan-fluxuen eredua erabiliz eta, bigarren fase batean, BPMN estandarrak definitzen duen lengoia grafikoa erabiliz, partekatze eta adoste lanak erraztu eta azkartzeko.
- Garapen-prozesua beste metodologia batzuen baliabideekin edo adostasun-maila handiko artefaktuen txantiloiekin aberastea, adibidez, *RUP* metodologia arina.
- CMMI 2.0 kalitate-ereduaren 2. maila lortzeko garapen-prozesua osatzea.
- CMMI 2.0 kalitate-ereduaren 3. maila lortzeko garapen-prozesua osatzea.

Hobekuntza posible gehiago ere identifikatu dira proiektua aztertzerakoan:

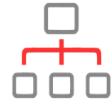
- Drupaletik zuzenean inferentzia motorrari deitu ahal izatea, tarteko fitxategirik erabili gabe. Orain sistemak erabiltzaileari esaten dio fitxategi bat ejekutatu behar duela.
- OpenUP bizi-zikloko workflow eredua eskuz definitu beharrik ez edukitzea. Horrek prozesua definitzeko denbora asko behar izatea eragiten du. Hobe izango litzateke webguneko informazioa erabiliz automatikoki sortzea.
- Prozesuaren informazioa gordetzeko metaeredua eta ereduak erabiltzea. Metaereduak erabiltzeak flexibilitatea ematen du etorkizunean eraldaketak egiteko komeni den formatura.
- Drupaletik sortzen diren datuak kanpoko datu-base baten gorde ordez Drupalen datu-basean gordetzea. Horrek Drupalek eskaintzen dituen aukerak aprobetxatzen ditu, datuak bistaratzeko eta editatzeko aukerak, adibidez.
- Drupal webguneararen itxura hobetu, defektuzko itxura aldatuz. Itxura egoki bat aurkitu webgunerako, dropdown menuak onartzen dituena.

4.4 Prestakuntza

Proiektu honen egileak bazituen proiektu honetarako erabilgarriak diren hainbat ezagutza, Informatika Ingeniaritzako Graduko hainbat irakasgaitan ikasitakoak. Esaterako, software projektuen, softwarearen bizi-zikloaren, metodologia zein estandarren oinarritzko ezagutzak.

Softwarearen Kalitatea irakasgaien proiektu honetan interesa duten hurrengo ekintzak jorratu ziren:

- *BPMn* oinarritutako software bat probatu, *Bizagi*. Software horren bidez, prozesuetan oinarritutako web-aplikazioa bat sortu zen. Lehenengo, *Bizagi Modeler* softwarearen bitartez prozesua modelatu, eta ondoren, prozesu horretan oinarritutako web-aplikazioa eraiki zen *Bizagi Studio* softwarearekin.



- *OpenUP* metodologia jarraitzen zuen proiektu bat osatu, softwarearen bizi-zikloa definituz. Ez ziren metodologiako artefaktu guztiak bete, baina bai hasierako fasekoak, betekizunen ingeniaritzari buruzkoak.
- Proiektu bat aurkezteko webgunea sortu eta antolatu *CCII-2016N-02* araua jarraituz.

Softwarearen Garapen Industriala irakasgaian beste gai hauetan landu ziren:

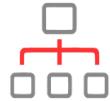
- Model Driven Engineering (MDE) edo ereduek bideratutako ingeniaritzaren oinarrizko kontzeptuak: metaeredua eta eredu.
- Domain Specific Language (DSL) edo domeinu zehatzeko lengoaien sorrera.
- ATLAS Transformation Language (ATL) erabilera ereduen arteko eraldaketak egiteko.
- Eclipse Modelling Framework (EMF) tresnen erabilera MDE-rako.

Web Sistemak irakasgaian ikasitakoa:

- XAMPP-en erabilera webguneak ordenagailu lokalean garatzeko.
- PHP programazio lengoaiaren erabilera web garapenerako.
- MYSQL-ren erabilera web garapenerako.

Bestetik, egileak prestakuntza zuen erabilgarriak izan diren beste gai batzuetan:

- Git eta GitHub-en erabilera bertsio kontrolerako.
- GitHub Pages-en erabilera webgune estatikoak sortzeko.
- Java eta Eclipse tresnen erabilera.



5 Arauak eta Erreferentziak

Kapitulu honetan, proiektuan zehar erabilitako araudia, bibliografia, metodoak, tresnak , ereduak, metrikak eta prototipoak deskribatuko dira.

5.1 Aplikatutako Legedia eta Araudia

Hainbat esparrutako legedia eta araudia aplikatzen da. Alde batetik, GrALari eta dokumentazioari buruzkoak eta bestetik erlazionatuta dauden administrazio publikoaren legeak.

Gradu Amaierako Lanen inguruko bete beharreko arautegia:

- [UPV/EHUko gradu amaierako lanen araudia](#). Euskal Herriko Unibertsitatean gradu amaierako lana egin eta defendatzeari buruzko arautegia.
 - [Informatika Fakultateko gradu amaierako lanen araudia](#). Informatika Fakultateko Gradu Amaierako Lanari buruzko arautegia.
 - [BOE-A-2009-12977](#). Informatika Ingeniaritzako Graduko edo Ingeniaritza Teknikoko titulazioak bete beharreko konpetentzia profesionalak eta Gradu Amaierako Lanen izaera profesionala ezartzen duen Errege Dekretua.

Administrazioa Publikoak ezarritako Sektore Publikoko Kontratuengen legeak eta aurkeztutako kexak:

- [BOE-261-2007-18874](#). 30/2007 Legea, urriaren 30ekoak, Sektore Publikoko Kontratuak buruzkoak.
 - [BOE-A-2011-17887](#). 3/2011 Legegintzako Errege Dekretua, azaroaren 14koak, Sektore Publikoko Kontratuak buruzko Legearren testu bategina onartzen duena.
 - [BOE-A-2017-12902](#). 9/2017 Legea, azaroaren 8koak, Sektore Publikoko Kontratuena, Europako Parlamentuaren eta Kontseiluaren 2014ko otsailaren 26ko 2014/23/EB eta 2014/24/EB zuzentarauen transposizioa egiten duena Espainiako ordenamendu juridikora. 93. Artikulua: Kalitatea bermatzeko arauak betetzen direla egiaztatzea.
 - [Recurso nº 6/2016 Resolución nº 100/2016](#). Kontratu baliabideen administrazio zentralaren erresoluzioa OESIA NETWORKS, S.L enpresaren errekursoari.

Kalitate-eredu eta giden inguruko informazio eta baliabideak:

- [CMMI-DEV, V1.3.](#) Improving processes for developing better products and services.
 - [CMMI 2.0.](#) Capability and performance model.
 - [PMBOK](#). Project Management Body of Knowledge.

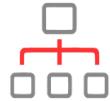
Proiektu honen dokumentazioen antolaketarako eta proiektuaren aurkezpenerako aplikatu den araua CCII-N2016 estandarra da.

- [CCII-N2016-01](#). Ingeniaritzako informatikoko proiektuen ikuskaritza edo bisa egiteko araua CCII N2016-01 estandarra da. Estandarrak dokumentuen osotasuna berrikusteko zerbitzuen prozesua deskribatzen du.
 - [CCII-N2016-02](#). Estandar honen ingeniaritzako informatikoko proiektuen dokumentazioaren antolaketa eta bere aurkezpena zehazten du. Memoria eta bere eranskinak estandar honen arabera antolatu dira, baita memoriarekin batera entregatu den webgunearen antolaketa.

5.2 Bibliografia

Jarraian, proiektuan zehar informazioa bilatzeko eta datuak lortzeko erabili diren erreferentzia bibliografikoak zerrendatuko dira ordena alfabetikoan.

CCII. (2016). *Norma CCII-N2016-01: Norma de Visado de Proyectos y Actuaciones Profesionales en Ingeniería Informática*.



<https://juletx.github.io/ProMeta/Proiekta/Memoriaren%20Eranskinak/A1%20-%20Sarrerako%20dokumentazioa/CCII-N2016-01.pdf> helbidetik eskuratua

CCII. (2016). *Norma CCII-N2016-02: Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática*.

<https://juletx.github.io/ProMeta/Proiekta/Memoriaren%20Eranskinak/A1%20-%20Sarrerako%20dokumentazioa/CCII-N2016-02.pdf> helbidetik eskuratua

Drupal. (2021). *Drupal Documentation*. <https://www.drupal.org/documentation> helbidetik eskuratua

Eclipse Foundation. (2012). *ABRD: Agile Business Rules Development*. <https://420-gelhy.github.io/EPF/ARBD/index.htm> helbidetik eskuratua

Eclipse Foundation. (2012). *OpenUP: Open Unified Process*. <https://420-gelhy.github.io/EPF/openup/index.htm> helbidetik eskuratua

Gonzalez, J. L. (2019). *BETRADOK proiekta: Betezunen trazabilitate inpaktu-analisi automatikoa eta dokumentazio formalaren sorkunta automatikoa modeloetan oinarritutako ekosistemetan. Gradu Amaierako Lana*. <https://juletx.github.io/BETRADOK/> helbidetik eskuratua

Pantheon. (2021). *Pantheon Documentation*. <https://pantheon.io/docs/> helbidetik eskuratua

Raño, J. R. (2020). *ProWF proiekta: Software proiekturen elaboraziorako workflowetan oinarritutako sistemaren sorkunta eta bizi-zikloa definitzeko metodologia baten ezarpena. Gradu Amaierako Lana*. <https://juletx.github.io/ProWF/> helbidetik eskuratua

5.3 Metodoak

Proiektuan hainbat metodo erabili dira, garapena baldintzatu dutenak.

5.3.1 OpenUP

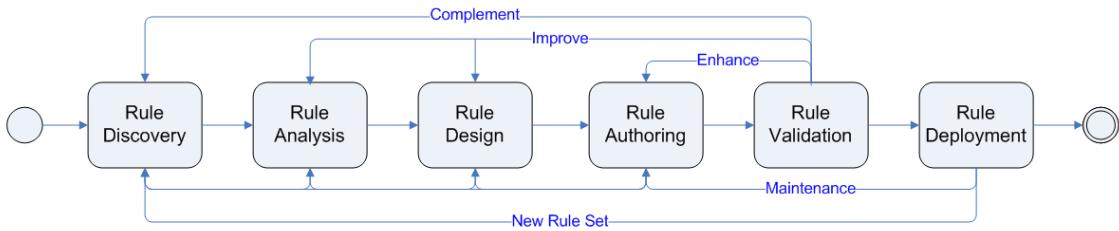
OpenUP softwarea garatzeko metodo eta prozesu bat da, teknologien sektoreko empresa multzo batek proposatutakoa, zeintzuk 2007an Eclipse Fundazioari dohaintzan eman zioten. Fundazioak lizentzia libre bezala argitaratu du eta eredu gisa mantentzen du Eclipse Process Framework (EPF) proiektuaren barruan.

Metodologia honek garrantzi handia izan du proiektu osoan zehar. Batetik, proiektuaren helburuetako bat metodologia baten definizioa eta ezarpena izan da eta OpenUP izan da aukeratutako metodologia. Bestetik, proiektuaren elaborazio prozesurako OpenUP metodologia jarraitu da, dokumentazioa bilduz eta proiektuaren kontrola eramanez.

5.3.2 ABRD

Agile Business Rules Development metodologiaren eredua ere erabili da. Horrela, bi metodologia erabiliz ziurtatzen da sistemaren egitura egokia dela. ABRD negozio erregela aplikazioa garatzeko metodologia da, erregela motorra eta BRMS⁷ erabiliz. Metodologia honek erregelen prozesua eta sistema hedatzeko beharrezkoak diren arkitektura diciplinak zehazten ditu. 5.1. Irudian metodologiaren prozesua ikus daiteke, fase eta jarduerekin.

⁷ BRMS: Business Rules Management System: Negozio logika aplikazioarekiko independenteki kudeatzeko sistema. Erabakiak automatikoki hartzea eta exekutatzea ahalbidetzen du.



5.1. Irudia. ABRD metodologiaren prozesuko faseak.

5.3.3 MDE

Model Driven Engineering (MDE) edo eredu bidezko ingeniaritza softwarea garatzeko metodologia da. Domeinu ereduak erabiltzen ditu, hau da, arazo zehatz bat lotutako gai guztien eredu kontzeptualak. Hori dela eta, aplikazioen domeinu jakin baten ezagutzaren eta jardueren irudikapen abstraktuak nabarmentzea du helburu, kontzeptu informatikoetan sartu gabe.

5.4 Tresnak

Atal honetan erabili diren tresna nagusien deskribapen labur bat egingo da.

5.4.1 Git

Linus Torvaldsek garatutako bertsio-kontrol software bat da. Hain zuzen, produktu edo konfigurazio batean egin daitezkeen aldaketak kudeatzeko programa da. Proiektuaren fitxategi guztiengarri bertsio-kontrolerako erabili da, hainbat errepositoriotan banatuta.

5.4.2 GitHub

Bertsio kontrolerako web-ostatu zerbitzua da, Git erabiltzen duena. Gehienbat iturburu koderako erabiltzen da. Git-en bertsio-kontrol banatu eta iturburu-kode kudeatzaile funtzionalitate guztiek eskaintzeaz gain bere ezaugarri propioak gehitzen ditu. Projektuaren Git errepositorio guztiek igozten erabili da.

5.4.3 GitHub Pages

GitHub-ekin integratuta, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan. Konfigurazio oso erraza, errepositoriotik bertatik egiten da. Webgune estatikoak bakarrik onartzen ditu, beraz, drupal-erako ez du balio, PHP erabiltzen baitu. Webguneak HTML, CSS, eta JavaScript-en idatzita egon daitezke edo Jekyll webgune estatiko sortzailea erabiliz.

5.4.4 Toggle Track

Proiektuko denbora neurtzeko erabili den tresna. Temporizadore baten bidez ataza bakoitzean pasatako denbora neurtu daiteke. Eskuz ere aldatu daiteke denbora ahaztu egin bazaigu. Abisatu egiten du inaktibo bagaude eta denbora neurtzen ari bagara. Aspaldian denborarik ez badugu neurtu ere abisatzen du, ez ahazteko. Denbora proiektutan, atazatan eta kategoriatan sailkatu daiteke. Edozein momentuko denboraren estatistikak ikusi eta deskargatu daitezke.

5.4.5 Java

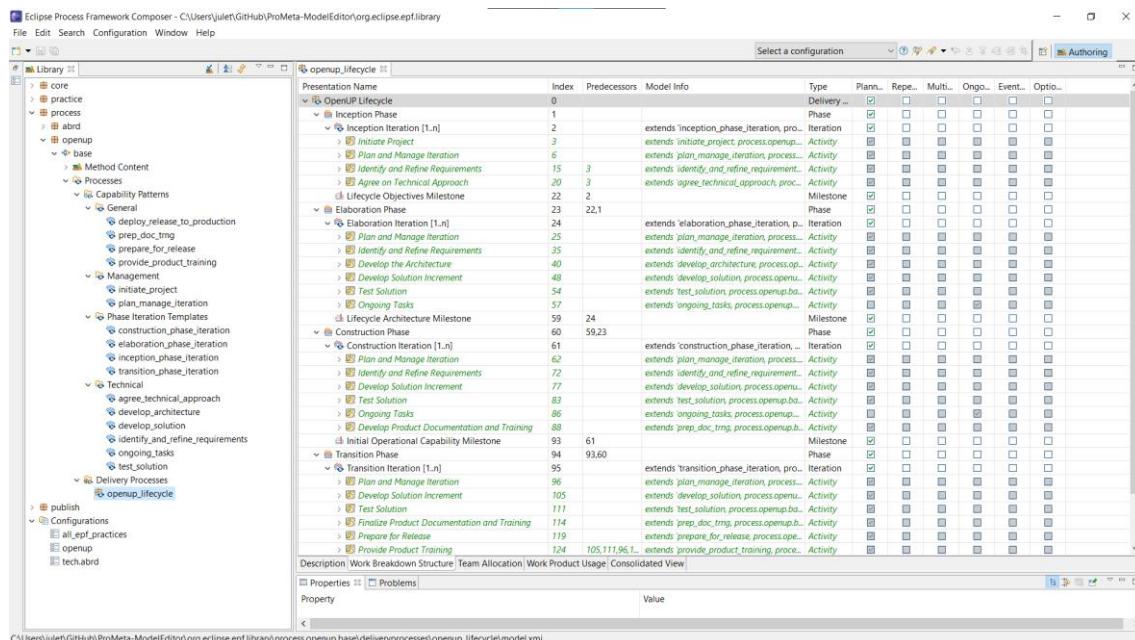
Sun Microsystems-ek garatutako programazio lengoia eta plataforma informatikoa da. Plataforma makina birtual bat da eta Java programazio lengoia eta garapen tresnak erabiliz garatutako aplikazioak exekutatzeko gai da. Proiektuan Java 8 bertsioa erabili da EPF Composer-erako eta Java 11 eta 15 Eclipseko garapenerako. Java programazio lengoia erabili da ModelEditor aspisistemaren garapenerako.

5.4.6 Eclipse IDE

Kode irekiko software plataforma bat da. Ematen zaion erabilera nagusia Java programatzeko garapen ingurune integratuarena (IDE) da. Proiektuan zehar ModelEditor azpisistemaren garapenerako erabili da.

5.4.7 EPF Composer

5.2. Irudian ikus daiteke EPF Composer tresnaren erabilera adibide bat.



The screenshot shows the Eclipse Process Framework Composer interface. On the left, there's a tree view of the 'Library' containing various process models like 'core', 'practice', 'abrd', 'openup', 'base', and 'openup_lifecycle'. The 'openup_lifecycle' model is selected. The main area displays a table titled 'openup_lifecycle' with columns for 'Presentation Name', 'Index', 'Predecessors', 'Model Info', 'Type', 'Plan...', 'Repe...', 'Multi...', 'Ongo...', 'Event...', and 'Optio...'. The table lists numerous activities and milestones from the OpenUP lifecycle, such as 'Inception Phase', 'Elaboration Phase', 'Construction Phase', and 'Transition Phase', along with their dependencies and properties. A status bar at the bottom indicates the file path: C:\Users\julen\GitHub\ProMeta-ModelEditor\org.eclipse.epf.library\processes\openup\base\deliveryprocesses\openup_lifecycle\model.xml.

5.2. Irudia. OpenUP bizi-zikloa EPF Composer tresnan.

5.4.8 Xtext

Programazio lengoaiak eta domeinu espezifikoko lengoaiak (DSL) garatzeko kode irekiko ingurunea da. Xtext-ek analizatzaile bat, sintaxi zuhaitz abstraktuaren klase eredua eta Eclipse-n oinarritutako IDE pertsonalizagarria sortzen ditu. Proiektuan testu editorea sortzeko eta SQL kodea sortzeko erabili da.

5.4.9 XSLT

eXtensible Stylesheet Language Transformations (XSLT) XMLn oinarritutako lengoaiak, XML fitxategiak transformatzeko balio duena. Proiektuan ModelEditor azpisisteman XML eredua UMAra transformatzeko erabili da.

5.4.10 Visual Studio Code

Microsoftek garatutako kode editorea. Bertsio kontrola, sintaxi nabamentzea eta kode osatze automatikoa bezalako aukerak eskaintzen ditu. Software librea eta doakoa da. Proiektuan zehar hainbat fitxategi mota editatzeko erabili da. IO-System sistemaren garapenena editore nagusia izan da. Gainera, bertsio kontrolerako ere erabili da, commit eta push guztiak bertatik eginez.

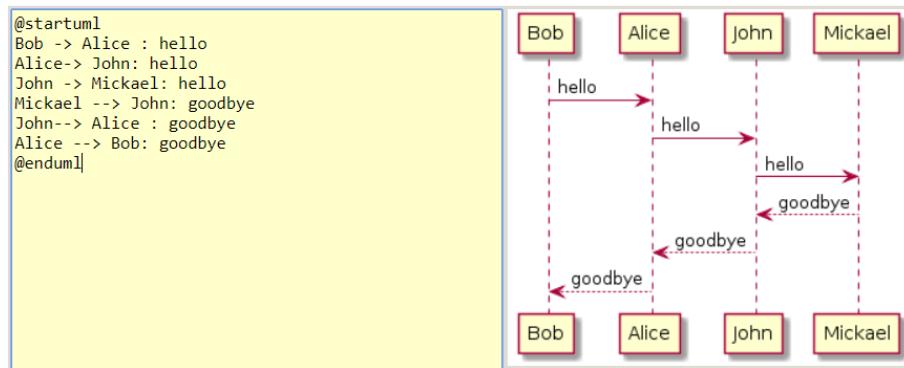
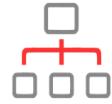
5.4.11 Microsoft Office

Microsoftek garatutako ofimatiika aplikazioa suitea. Ordainpekoa da Office 365 programaren harpidetza bitartez. Proiektuan zehar Word eta Excel erabili dira memoria eta eranskinak idazteko. PowerPoint erabili da aurkezpena egiteko.

5.4.12 PlantUML

Kode irekiko tresna. Testu planotik abiatuta eta etiketa bidezko lengoaiak definitu bat erabiliz, UML diagramak sortzeko balio du. Adibidez, 5.3. Irudia lengoaiak deskribitza erakusten da eta horretatik sortutako diagrama simple baten eredu.

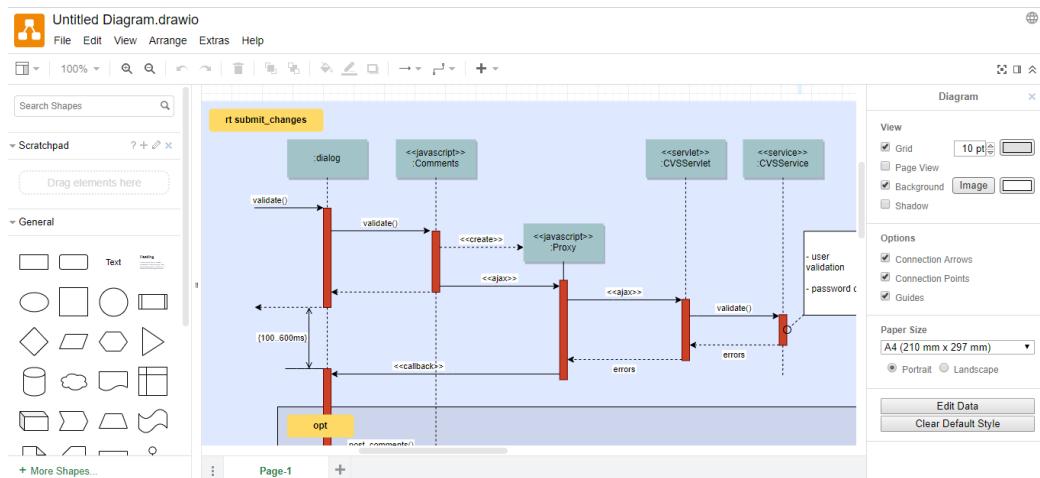
Proiektu honetan, OpenUP metodologiak eskatzen dituen erabilpen kasuen ereduak sortzeko erabili da.



5.3. Irudia. PlantUMLren lengoain idatzitako sekuentzia-diagrama.

5.4.13 Draw.io

Hainbat motatako diagramak sortzeko aukera ematen duen softwarea da. Eskuz sortzen dira diagramak eta bi aukera daude aplikazioa erabiltzeko: online edo offline. Proiektuan zehar aplikazio hau erabili izan da dokumentazioan agertzen diren diagrama batzuk egiteko. 5.4. Irudian driagrama adibide bat ikus daiteke.



5.4. Irudia. Draw.io diagrama adibidea.

5.4.14 Drupal

Drupal edukiak kudeatzeko sistema edo CMS librea, modularra eta oso konfiguragarria da. Sistema dinamikoa da, hau da, zerbitzariaren edukiak modu finkoa biltegiratu beharrean, orrien testu-edukia eta beste konfigurazio batzuk datu-base batean biltegiratzen dira eta web-ingurune bat erabiliz editatzen dira. 5.5. Irudian proiektuan sortutako Drupal webgunea ikus daiteke.

ProMeta IO-System sistemaren interfaze moduan erabili da, datuen sarrerra/irteerak kudeatzeko.

The screenshot shows a Drupal administrative interface. At the top, there's a navigation bar with links like 'Manage', 'Shortcuts', 'Juletxara', 'Content', 'Structure', 'Appearance', 'Extend', 'Configuration', 'People', 'Reports', 'Help', 'Edit', 'My account', and 'Log out'. Below the navigation is a header with the logo 'ProMeta' and a 'RSS Feed' link. The main content area has a title 'Methodologies' with a star icon. On the left, there's a sidebar with a 'Title' input field and an 'Apply' button. The main content area lists two configurations:

Title	Description	Version
Agile Business Rules Development	This configuration publishes the Agile Business Rules Development practice.	7.5.0
OpenUP	OpenUP is a lean Unified Process that applies iterative and incremental approaches within a structured lifecycle.	1.0.0

5.5. Irudia. Proiektuan sortutako Drupal webgunea.

5.4.15 XAMPP

XAMPP software libreko pakete bat da, nagusiki MySQL datu-baseen kudeaketa sistema eta Apache web zerbitzaria integratzen duena. 5.6. Irudian ikus daiteke XAMPPen erabileraen adibide bat.

Proiektuan honetan Drupal webgunea lokalean garatzeko erabili da.

The screenshot shows the XAMPP Control Panel version 3.2.4. It displays a table of services with columns: Service, Module, PID(s), Port(s), and Actions (Stop, Admin, Config, Logs). The table shows the following entries:

Service	Module	PID(s)	Port(s)	Actions
Apache	Apache	2404 12372	80, 443	Stop Admin Config Logs
MySQL	MySQL	7176	3306	Stop Admin Config Logs
FileZilla				Start Admin Config Logs
Mercury				Start Admin Config Logs
Tomcat				Start Admin Config Logs

To the right of the table is a vertical toolbar with icons for Config, Netstat, Shell, Explorer, Services, Help, and Quit. Below the table is a log window showing status messages:

```

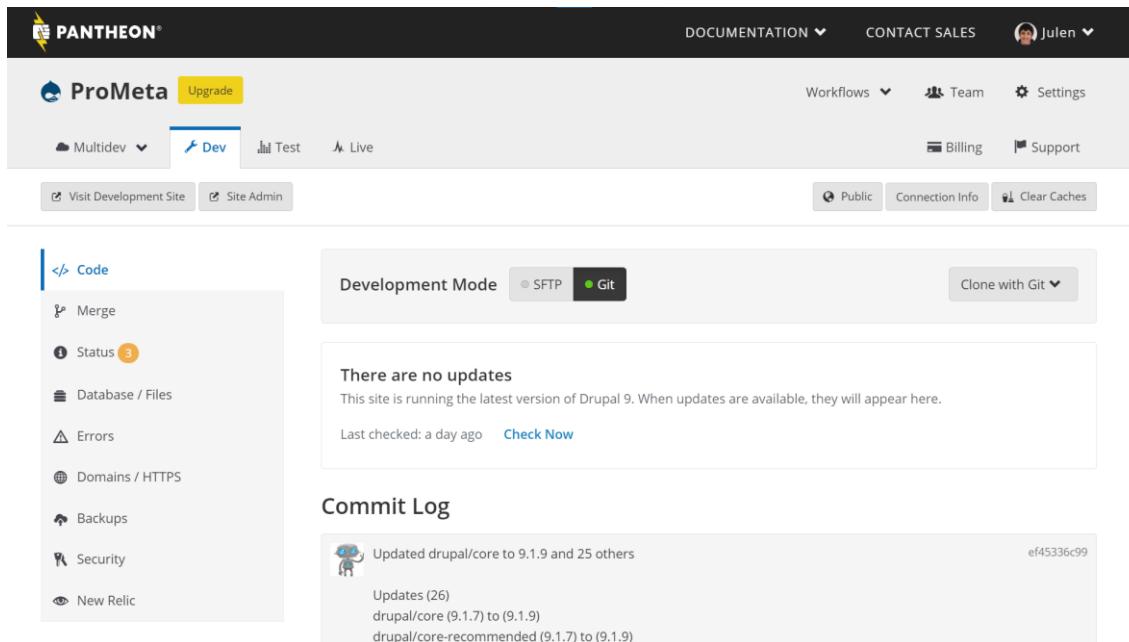
16:32:55 [Apache] Status change detected: running
16:32:56 [mysql] Attempting to start MySQL app...
16:32:56 [mysql] Status change detected: running
16:44:31 [Apache] Attempting to stop Apache (PID: 12644)
16:44:31 [Apache] Attempting to stop Apache (PID: 10868)
16:44:31 [Apache] Status change detected: stopped
16:45:12 [Apache] Attempting to start Apache app...
16:45:12 [Apache] Status change detected: running

```

5.6. Irudia. XAMPP kontrol panela Apache eta MySQL hasieratzeko.

5.4.16 Pantheon

Pantheon kode irekiko Drupal eta WordPress webguneetarako hosting plataforma da. 5.7. Irudian Pantheon kontrol panela ikus daiteke.



5.7. Irudia. ProMeta webguneko Pantheon kontrol panela.

5.4.17 MariaDB

MariaDB datu-baseak kudeatzeko sistema erlazional libre bat da. Oracle Corporationek MySQLren jabea zen Sun Microsystems erostean MySQLren jatorrizko garatzaileetako batzuk GNU Lizentzia Publiko Orokorra lizenziapenera mantentzen asmoz sortutako fork bat da. Datu-base erlazional moduan erabili da proiektau.

5.4.18 phpMyAdmin

phpMyAdmin PHP-n idatzitako tresna da MySQL administrazioa web orrien bidez kudeatzeko. Proiektaun datu-base erlazionalak kudeatzeko erabili da.

5.4.19 PHP

PHP (PHP: Hypertext Preprocessor) interpretatutako programazio lengoia bat da, batez ere webgune dinamikoak sortzeko erabili ohi dena. Datu-base sistema ugarirekin funtzionatzeko aukera izatea eta sistema eragile gehienetarako eskuragarri izatea dira beronen abantaila nagusiak. Proiektaun Drupal-erako programazio lengoia moduan erabili da.

5.5 Ereduak

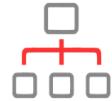
Proiektaun hainbat eredu erabili dira, batzuek dokumentazioarekin erlazionatuak eta beste batzuk implemenatzioarekin.

5.5.1 CCII-N2016-02

Arauen atalean aipatu den moduan, estandar honetan oinarrituta antolatu da memoria eta proiektauren webgunea. CCII erakundeak sortutako araua da, ingeniaritzaren informatikoko proiektautarako dokumentazioaren egitura eta beharrezkoak diren dokumentu eta sekzioak definitzen dituena.

5.5.2 OpenUP

OpenUP metodologiaren eredua erabili da. Eedu horrek metodologiaren informazio guztia du. Metodologia hau jarraitzeko, bere webgunean artefaktu batzuen txantiloia daude eskuragarri. Txantiloioriek jarraituz OpenUP metodologiaren bitartez sortutako artefaktuak idatzi dira.

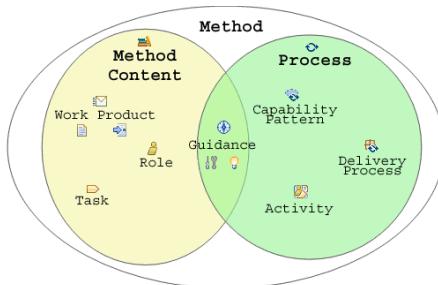


5.5.3 ABRD

Agile Business Rules Development metodologiaren eredua ere erabili da. Eredu horrek metodologiaren informazio guzta du. Metodologia hau jarraitzeko, bere webgunean artefakto batzuen txantiloia daude eskuragarri.

5.5.4 UMA

Unified Method Architecture metaeredua erabili da OpenUP eta ABRD ereduak definitzeko. Metaeredu honen helburua edozein metodologia modelatu ahal izatea da. 5.8. Irudian ikusten den moduan, UMA metodoaren edukia prozesutik bereizten da. Horrela, metodoaren edukia hainbat prozesutan berrerabil daiteke.



5.8. Irudia. UMA metodologiako metodoaren eta prozesuaren kontzeptuak.

5.5.5 Ecore

EMF-ko Ecore meta-metaeredua erabili da UMA metaeredua definitzeko. Meta-metaeredu honen helburua edozein metaeredu modelatu ahal izatea da. Ecore oinarri moduan hartuta definitzen dira metaeredu guztiak, baita Ecore bera ere.

5.6 Metrikak

Proiektuaren helburuekin erlazionatutako 4 metrika nagusi daude: irismena, denbora, kostua eta kalitatea.

5.6.1 Irismena

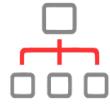
Proiektuaren irismena neurtzeko atazak definitu dira eta bakoitzaren denbora estimazioa egin da. Kontuan hartuta proiektua eta proiekutuko taldea txikiak direla, OpenUP metodologia erabiltzea nahikoa da. Gainera, metodologiako lehenengo bi edo hiru fase egitearekin nahikoa izango da. Talde handiagoa edo proiektu konplexuagoa izango balitz agian RUP bezalako metodologia konplexuagoa bat beharko genuke.

5.6.2 Denbora

Denboraren kontrola egiteko ataza bakoitzean pasatako denbora neurtu da, Toggle Track aplikazioaren kronometroa erabiliz. Ondoren, neurtutako denbora estimatutakoarekin konparatu da, eta horren arabera erabakiak hartu dira.

5.6.3 Kostua

Proiektuaren kostua neurtzeko aurrekontua egin da. Bertan, giza baliabideen eta erreminten kostuak zehaztu dira. Erreminta guztiak doakoak izatea espero denez, kostua kudeatzea erraza da. Giza baliabideen kostua jakiteko proiekutuko kide bakoitzaren ordu kopurua eta orduko kostua biderkatzen dira.



5.6.4 Kalitatea

Proiektuaren kalitatea neurtea garrantzitsua da onargarria dela ziurtatzeko. Kudeaketa planean zehazten den moduan zaintzeko hiru aspektu kontrolatu behar dira: implementazioa, funtzionalitateak eta dokumentazioa.

5.7 Prototipoak

Proiektuan bi prototipo nagusi garatu dira, azpisistema bakoitzari dagozkionak, ModelEditor eta IO-System.

5.7.1 ProMeta ModelEditor

Sortutako editore grafikoa eta testu editorea erabiliz prozesuaren eredua sortzeko eta editatzeko aukera emango du. Sistema honen ardura prozesu ingeniarri rolak izango du.

5.7.2 ProMeta IO-System

CMS baten bitartez kudeatutako web-aplikazioa izango da. Helburua metodologia jarraitzen duten proiektuen informazioa gordetzea da. Rol bakoitzak metodologian dituen ataza berdinak bete beharko ditu.

6 Definizioak eta Laburdurak

Atal honetan memorian zehar agertu diren termino definizioak eta laburduren esanahiak azalduko dira.

6.1 Bizagi

Bi produkto osagarri dituen softwarea da, prozesuen modelatzalea (Bizagi Modeler) bat eta BPMren suite ofimatiko bat (Bizagi Studio).

6.2 BPM

Sigla(ingeles), Business Process Management. Enpresei prozesuak automatikoki modelatzeko, implementatzeko eta exekutatzeko aukera ematen dien software teknologia da.

6.3 CCII

Sigla(gaztelera), Consejo de Colegios de Ingeniería Informática. Estatu-mailan informatika ingeniariz guztiak errepresentatu eta bateratzen dituen antolakunde da. Ikus, gainera: [CCII, webgunea](#).

6.4 CCII-N2016-02

CCII erakundeak sortutako araua, ingeniaritzako informatikoko proiektuetarako dokumentazioaren estruktura eta beharrezkoak diren dokumentu eta sekzioak definitzen dituena. Informazio gehiago: <https://www.ccii.es/norma>

6.5 CMMI

Capability Maturity Model Integration. Software-sistemak garatzeko, mantentzeko eta erabiltzeko, prozesuak hobetzeko eta ebalutzeko eredu da, CMMI Institutuak administratutakoa.

6.6 CMS

Sigla(ingeles), Content Management System. Dokumentuak eta bestelako edukiak antolatu eta kudeatzeko softwarea da, normalean web-aplikazioa.

6.7 DOT

Testu lauan idatzitako lengoia deskriptiboa da. Grafoak deskribatzeko modu simple bat eskaintzen du, gizakiek eta konputagailuek ulertzeko moduko.

6.8 Drupal

Edukiak kudeatzeko sistema edo CMS librea, modularra eta oso konfiguragarria. Ikus, gainera: [Drupal, webgunea](#).

6.9 IBM

Sigla(ingeles), International Business Machines. Informatikarekin lotutako tresnak, programak eta zerbitzuak ekoitztu eta merkaturatzen dituen enpresa multinacionala da.

6.10 OpenUP

Open Unified Process softwarea garatzeko metodo eta prozesu bat da, Eclipse Fundazioak garatua. Rational Unified Process (RUP) metodologiaren azpimultzo minimoa da. Proiektua iteraziotan banatzen du eta eta proketuaren bizi-zikloaren lau fasetan banatzen du: Hasiera, Elaborazioa, Eraikuntza eta Trantsizioa. Informazio gehiago: [OpenUp 1.0](#) edo [OpenUp 1.5](#).

6.11 PlantUML

Testu laua UML diagrametan bihurtzeko balio duen software irekia. Ikus, gainera: [PlantUML, webgunea](#).

6.12 ProMeta

Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema. Proiektu honen izena ingeleseko hitzetatik eratutako hitz-jokoa da. “Pro” profesional edo process hitzetik dator. “Meta” metamodel hitzaren laburdura da eta metaeredu esan nahi du.

6.13 ProWF

Software proiekturen elaboraziorako workflowetan oinarritutako sistemaren sorkuntza eta bizi-zikloa definitzeko metodologia baten ezarpena. Proiektu honen aurrekariaren izena ingeleseko hitzetatik eratutako hitz-jokoa da. “Pro” professional hitzetik dator eta profesionala esan nahi du, “WF” workflow hitzetik datorren laburdura da eta lan-fluxu esan nahi du.

6.14 RUP

Sigla(ingelesez), Rational Unified Process. Rational Software enpresak garatutako software-prozesu bat da. Objektuetara bideratutako sistemak aztertu, diseinatu, implementatu eta dokumentatzeko erabiltzen den metodologia estandarra.

6.15 SPICE

Software Process Improvement and Capability dEtermination. ISO/IEC 15504. Garapen-prozesuak hobetzeko, ebalutzeko, informazio-sistemak eta software-produktuak mantentzeko eredu da.

6.16 UML

Unified Modeling Language (Modelaketarako lengoia bateratua) sistemak zehaztu, diseinatu eta eraikitzea lengoia da, printzipioz objektuei orientatutako programaziorako prestatuta dagoena. UML aplikazio baten garapen fase guztiek modelatzeko lengoia homogeneo bat definitzen saiatzen da, bezeroaren zehaztapenatik hasita programatzialearen diseinu xehera arte.

6.17 UNE

Sigla(gazteleraz), Una Norma Española. Comités Técnicos de Normalización (CTN) batzordeak sortutako arauen, arau experimentalen eta txostenen (estandarrak) multzoak dira.

6.18 URPS

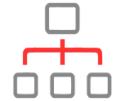
Sigla (Ingelesez). Usability, Reliability, Performance and Supportability. Softwarearen kalitate ezaugarriak dira: Erabilgarritasuna, Fidagarritasuna, Errrendimendua eta Mantenigarritasuna.

6.19 Workflow

Aspektu operacionalekin lan-aktibitate bat deskribatzeko egiten den irudikapena. Irudikapen horretan atazak nola egituratzen diren, zein den atazen arteko ordena eta nola sinkronizatzen diren, nolakoa den atazen informazio-fluxua eta atazen betetzearen jarraipena nola egiten den grafikoki deskribatzen da.

6.20 WYSIWYG

Sigla (ingelesez), What You See Is What You Get. Testu-prozesadoreei eta beste testu-editore batzuei aplikatutako esaldi bat da, azkenengo emaitza zuzenean erakutsiz dokumentu bat idazteko aukera ematen duena.



7 Hasierako Betekizunak

Kapitulu honetan proiektuaren hasierako betekizun funtzionalak eta ez-funtzionalak azaltzen dira. Gainera, sistemak izan behar dituen kalitate ezaugarriak eta sistemaren interfazeen ezaugarriak deskribatzen dira.

7.1 Betekizun Funtzionalak

Betekizun funtzionalek sistemaren beharrak eta horiek betetzeko ezaugarriak finkatzen dituzte. 7.1. Taulan datu horiek ikus daitezke lehentasunarekin eta entrega datarekin batera. Betekizun hauek goi-mailan definitu dira, aurrerago zehaztuko da gehiago bakoitzaren inguruan.

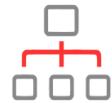
Beharra	Ezaugarriak	Lehentasuna	Entrega Data
Software garapeneko prozesuaren definizioa	Software garapeneko prozesuen metaeredua definitu. Gutxienez OpenUP metodologiaren eredua definitu.	Altua	2021/06/20
Editore grafikoa eta testuala	Metaeredua erabiliz ereduak aldatzeko editoreak sortu. Editoreen arteko bateragarritasuna bermatu.	Altua	2021/06/20
Prozesuaren datu-basea	Prozesuen informazioa gordeko duen datu-basea definitu eta datuak gorde.	Altua	2021/06/20
Prozesuaren webgunea	Web interfazea garatu. Web kodea garatu. Webgunea zerbitzari batean jarri.	Altua	2021/06/20

7.1. Taula. Hasierako betekizun funtzionalak.

7.2 Betekizun Ez-Funtzionalak

Betekizun ez-funtzionalek sistemaren funtzionalitateekin zuzenean erlazionatuta ez dauden betebeharrok deskribatzen dituzte. 7.2. Taulan betekizun ez-funtzionalak eta horien lehentasuna eta entrega data ikus daitezke. Orokorrean, betekizun funtzionalek baina lehentasun txikiagoa dute, sistemaren funtzionamendurako ez baitira ezinbestekoak.

Betekizuna	Lehentasuna	Entrega Data
Dokumentazioa eta proiektuaren webgunea CCII estandarraren arabera.	Altua	2021/06/20
Proiektuan OpenUp metodologia jarraitzea.	Altua	2021/06/20
Garapen prozesua metodologia eta estandarrek esaten duten moduan eratzea.	Altua	2021/06/20
Erreminta, metodologia eta ezagutzaren aldetik eman daitezkeen aldaketen aurrean, soluzioa malgua eta egokigarria izatea.	Ertaina	2021/06/20



Proiektuaren garapenerako doakoak eta libreak diren tresnak erabiltzea.	Ertaina	2021/06/20
Proiektuko osagaien dokumentazioa eta eskuliburuak	Ertaina	2021/06/20

7.2. Taula. Hasierako betekizun ez-funtzionalak.

7.3 Sistemaren Ezaugarriak

Software kalitatearekin erlazionatutako URPS⁸ ezaugarriak deskribatuko dira. Garrantzitsua da argi edukitzea sistemak izan behar dituen kalitate ezaugarriak.

7.3.1 Erabilgarritasuna

Sistemak erabilgarritasun altua izango du. Honek esan nahi du erabiltzeko, ikasteko eta memorizatzeko erraza izango dela. Ez da ikastarorik beharko aplikazioa erabili ahal izateko, intuitiboa denez erabiltzaileek erraz ikasiko baitute. Sistemak erabiltzailea laguntzeko eskuliburuak eta oharrak izango ditu, sistemako orri guztietan argi izango du zer egin behar den. Sisteman erabiltzen ikastea prozesu azkarra izango da, funtzionalitate sinpleak izango ditu eta ezagunak software talde baten partaideentzat.

7.3.2 Fidagarritasuna

Sistemak fidagarritasun altua izango du. Honek esan nahi du ia beti eskuragarri egon behar duela, hutsegiteek eragin txikia izan behar dutela eta hauetatik azkar berreskuratuko dela. Fidagarritasuna bermatzeko, sistema monitorizatuko da arazo potentzialak azkar identifikatu eta ekiditeko.

7.3.3 Errendimendua

Sistemak errendimendu altua izango du. Honek esan nahi du erantzun denbora azkarra edukiko duela eta aldi berean hainbat konexio onartuko dituela. Hasieratze eta amaitze denbora ere azkarra izango da. Horretarako, garrantzitsua izango da zerbitzariak ahalmen nahikoa izatea.

7.3.4 Mantenugarritasuna

Sistemak mantenugarritasun altua izango du. Honek esan nahi du instalatzeko, konfiguratzeko, eguneratzeko eta mantentzeko erraza izango dela. Erabiltzaileek web bidez erabiliko dute sistema, eta beraz ez dute ezer instalatu beharrik izango. Instalazioa, eguneratzeak eta mantenua zerbitzarian egingo dira.

7.4 Sistemaren Interfazeak

Sistemaren erabiltzaile interfazeen ezaugarri garrantzitsuak deskribatuko dira. Sistemaren interfazeak egokiak izatea ezinbestekoa da erabiltzailearen esperientzia ona izateko.

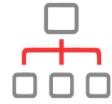
7.4.1 Itxura eta Sentsazioa

Orrialdean erabilitako kolore eta itxurari dagokionez, koloreak kontuz aukeratuko dira irakurgarritasuna bermatzeko. Gainera, kontzeptuak OpenUp prozesuan erabiltzen diren antzeko koloreekin adieraztea izango da helburua, lortura errazagoa izan dadin. Erabiliko diren menuak eta aukerak software proiektyuetan aritzen direnentzat ulerterrazak eta ezagunak izan behar dira.

7.4.2 Diseinu eta Nabigazio Betekizunak

Nabigazio menuan funtzionalitate nagusiak bilduko dira, eskuragarri egon daitezten. Menuan agertzen diren aukerak erabiltzaile motaren araberakoak izango dira, eta menua atzigarria izan behar du edozein

⁸ URPS: sigla (Ingelesez). Usability, Reliability, Performance and Supportability. Softwarearen kalitate ezaugarriak dira: Erabilgarritasuna, Fidagarritasuna, Errendimendua eta Mantenigarritasuna.



momentutan. Sistemaren atal desberdinak modu egokian antolatuta egongo dira. Sistema responsive izango da, pantailaren tamainaren arabera itxura automatikoki aldatuko da.

7.4.3 Konsistentzia

Interfazeak konsistentzia mantendu beharko du, itxuraz berdinak diren elementuek funtzionalitate antzekoa izan behar dute. Gainera, interfazeetako elementuek ohiko portaera izango dute, erabiltzaile gehienentzat beste sistemetatik ezaguna dena.

7.4.4 Erabiltzailearen Pertsonalizazio Betekizunak

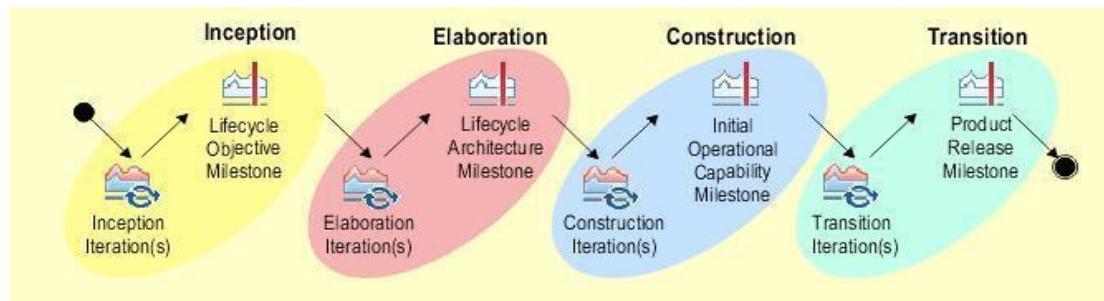
Ez da aurreikusten erabiltzailean pertsonalizazio aukerarik izatea. Baliteke, interfazearen itxura edo hizkuntza aldatzeko aukeraren bat gehitzea.

8 Irismena

Kapitulu honek proiektuaren irismena definitzea eta proiektuak sortutako entregagarriak zerrendatzea du helburu.

8.1 Bizi-zikloa

Proiektu honen irismena finkatzeko, OpenUP metodologiaren bizi-zikloa jarraitu da. 8.1. Irudian ikusten den moduan bizi-ziklo hori lau fasez osatuta dago: hasiera, elaborazioa, eraikuntza eta trantsizioa.



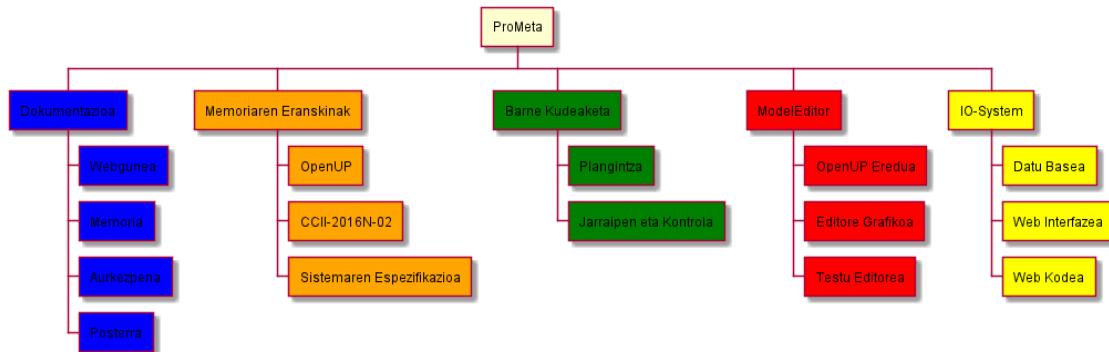
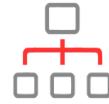
8.1. Irudia. OpenUP metodologiako bizi-zikloaren faseak.

Proiektuaren kasuan eraikuntza faseraino iritsiko da, produktuaren lehenengo prototipoa ateraz eta dokumentazioa sortuz. Denbora mugatuko proiektua izanik, trantsizio fasa kanpoan geratu da. Hurrengo zerrendan deskribatzen dira irismena definitzen duten ezaugarriak:

- OpenUP metodologiak eskatutako dokumentuak betetzea. Horretarako OpenUP metodologiak bere webgunean eskaizten dituen txantiloia jarraituz.
- CCII-N2016-02 arauak eskatzen dituen dokumentuak betetzea. Ingeniaritza informatikako proiektu profesional baten dokumentazioa ere profesionala izan dadin, arau estandar bat erabiltzea oso garrantzitsua da.
- Proiektuaren webgunea osatu. Webgune honetan jarritako dokumentuak CCII-N2016-02 arauak eskatzen duen dokumentazio egitura jarraituz. Bertan, proiektuaren memoria, memoriaren eranskinak, OpenUP metodologiarekin sortutako dokumentu guztiak eta proiektuarekin zerikusia duten hainbat aspektu agertuko dira.
- Proiektuaren memoria idaztea.
- Proiektuaren defentsa prestatzea. Horretarako, memorian idatzitako aspektu guztiak laburbiltzen dituen aurkezpen bat prestatuz.
- Proiektuaren posterra egitea.

8.2 LDE Diagrama

LDE diagramaren bitartez lan-pakete eta ataza nagusiak modu argian adieraz daitezke. 8.2. Irudian ikus daiteke LDE diagrama.



8.2. Irudia. LDE diagrama.

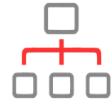
Hurrengo zerrendan proiektuko lan-pakete eta ataza nagusiak deskribatzen dira.

- **Dokumentazioa**
 - **Webgunea:** proiektuaren dokumentu guztiak biltzen dituen eta CCII-2016N-02 araua betetzen duen webgunea.
 - **Memoria:** proiektuaren memoriaren txantiloia prestatu eta memoria idatzi.
 - **Aurkezpena:** proiektuaren aurkezpena prestatu.
 - **Posterra:** proiektuaren posterra egin.
- **Memoriaren Eranskinak**
 - **OpenUP:** metodologiak zehazten dituen artefaktuak, hurrengo atalean zerrendatzen direnak.
 - **CCII-2016N-02:** arauak zehazten dituen memoriaren eranskinak.
 - **Sistemaren Espezifikazioa:** OpenUP metodologiak zehazten dituen artefaktuak.
- **Barne Kudeaketa**
 - **Plangintza:** OpenUP metodologiaren plangintza dokumentuak.
 - **Jarraipen eta Kontrola:** OpenUP metodologiaren jarraipen eta kontroleko artefaktuak.
- **ModelEditor**
 - **OpenUP Eredua:** OpenUP metodologiaren informazio guztia duen eredua.
 - **Editore Grafikoa:** metodologien ereduak grafikoki aldatzeko editorea.
 - **Testu Editorea:** metodologien ereduak testu bidez aldatzeko editorea.
- **IO-System**
 - **Datu Basea:** metodologiaren informazioa gordetzen duen datu-basea sortu eredutik.
 - **Web Interfazea:** webgunearen interfazea aldatu.
 - **Web Kodea:** webgunearen kodea programatu.

8.3 Artefaktuak

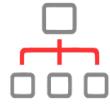
OpenUP metodologiari dagokionez, 8.1. Taulan ikusten dira bete diren dokumentuak, domeinuaren arabera sailkatuta. Esan bezala, hauek hasiera, elaborazio eta eraikuntza faseei dagozkien dokumentuak dira.

Domeinua	Artefaktua
Arkitektura	Arkitektura Kuadernoa
Hedapena	Produktuaren Dokumentazioa Laguntza Dokumentazioa Erabiltzaile Dokumentazioa Trebatzeko Materialak
Garapena	Implementazioa Eraikuntza



	Diseinua
Ingurunea	Garapen Kasua Tresnak
Proiektu Kudeaketa	Iterazio Plana Proiektu Plana Lan-atazen zerrenda Arriskuen zerrenda
Betekizunak	Glosategia Ikuspegia Betebeharren Espezifikazioa Erabilpen Kasuak Erabilpen Kasuen Eredua
Proba	Proba Kasuak Proba Log-ak Proba Script-ak

8.1. Taula. OpenUP metodologiako bete diren artefaktuak domeinutan sailkatuta.



9 Hipotesiak eta Murritzapenak

Kapitulu honetan, proiektuaren hasierako hipotesiak eta proiektuaren garapenerako ezarritako murriztapenak deskribatuko dira.

9.1 Hipotesiak

Proiektuarekin lanean hasteko hipotesi batzuk ateratzen ditugu aurreko lan eta datuetatik. Hurrengoak dira proiektuaren hasierako hipotesiak:

- Metaereduen erabilerak malgutasuna ematen du proiektuaren hurbilpena aldatzeko edo etorkizunean beste hurbilpen batzuk probatzeko.
 - Definitutako garapen prozesuak software proiektuen elaborazioa gidatu eta kontrolatuko du. Prozesua aldatzeak sistemaren portaera eta datu-basea automatikoki aldatzea ekarriko du.
 - CMS baten erabilera datuen sarrera/irteerarako irtenbide egokiena da. Webgune bat sortzeko aukera ematen duen tresna erabilerraza izateaz aparte, ez da baliabide tekniko aurreratuetara etengabe jo behar. Kudeaketa, administrazioa eta mantentze-lanak egiteko laguntza ematen du kanpoko baliabiderik erabili gabe.
 - Datu-base erlazionalak prozesu baten ezagutza gordetzeko modurik egokiena da, datuen independentzia, emaitzen koherenzia eta datu-basearen produktibilitatea handitzea lortuz.
 - Sistema iteratiboki hobetzen joango denez, estandarretan oinarritutako garapenak bere mantenua eta hedapena errazten ditu.

9.2 Murriztapenak

Proiektu informatikoen bezeroen eskakizunen ondorioz, neurri batean mugatu egin dira proiektuarekin lotutako elementu batzuk egiteko kontuan hartu beharreko aukerak. Hurrengoa dira proiektuaren hasierako murritzapenak:

- Kostu ekonomikoa ez da batere aldatu hasierako planteamendutik. Izan ere, proiektuan zehar ez da kostu gehigarrik sortu, erabilitako teknologia guztiak doakoak izan direlako.
 - Denborari dagokionez, ekaínaren 20rako proiektua bukatzeko murriztapena bete behar da.
 - Kalitateari dagokionez, proiektuak denbora eta kontu murriztapenen barruan kalitate onargarria izan beharko du.
 - Betezizunen ingeniaritza eta bizi-zikloa definitzen duen metodologia bat jarraitzea proiektuaren elaborazio eta garapenerako: OpenUP. Bezeroak egiten duen exijentzia da metodologia bat erabiltzea.
 - Proiektuaren dokumentuen antolaketarako CCII-2016N-02 estandarra erabiltzea.
 - Sortuko den software proiektuen elaboraziorako sistema web bidez atzigarrria izan behar du.

10 Aukeren Azterketa eta Egingarritasuna

Kapitulu honetan proiekutako soluzioaren alternatibak, balorazio-irizpideak eta hautatutako aukeren justifikazioak agertzen dira.

10.1 Arkitektura

Soluzioaren arkitekturari dagokionez, ProWF proiektuan bi aukera aztertu ziren: *Bizagi* bezalako softwarearen bitarbez prozesuetan (*BPM*) oinarritutako web-aplikazioa eraikitzea edo arkitektura propioa sortzea. Proiektu honetan aukera berri bat gehitu da.

10.1.1 Bizagi

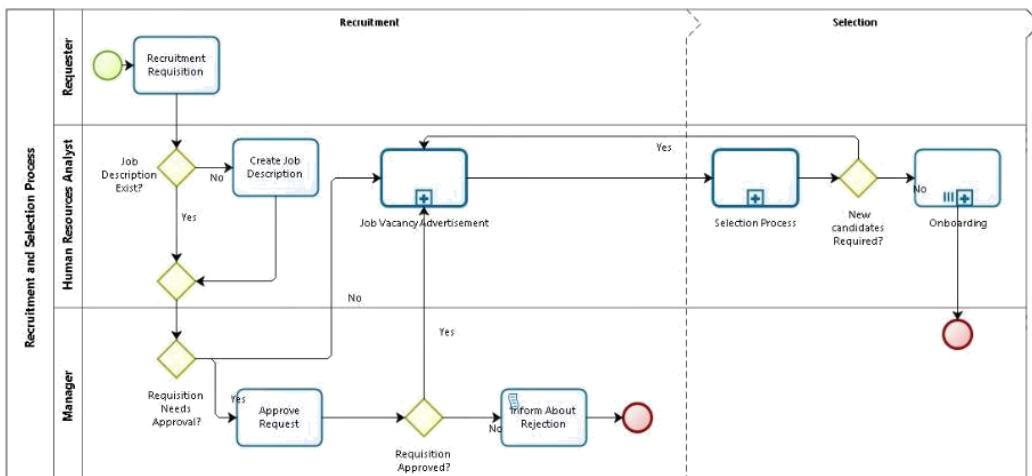
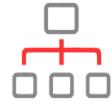
Bizagi Softwarearen Kalitatea irakasgaian erabilitako softwarea da. Bertan, *Bizagi Modeler* editorearen bitartez prozesuak sortu eta ondoren, *Bizagi Studio* tresnan negozio-erregelak adieraziz, datu-basea konfiguratz, formularioak definituz, web-zerbitzuak integratuz eta beste hainbat aspektu ukituz, sortutako prozesuan oinarritutako web-aplikazio bat sortu zen. Sortutako web-aplikazioak itxura oso profesionala zuen eta bere sorkuntza ez zen izan batere zaila.

Proiektu honetarako bideragarria izango litzateke software hau erabiltzea, hurrengo bi arrazoiengatik:

- Lan-fluxua definitzeko *Bizagi Modeler* editorea erabili daiteke, bizi-zikloa definitzen duten metodologiek faseak eta rolak erabiltzen dituztelako, 10.1. Irudian ikus daitekeen bezala. Beraz, *Bizagi Modeler* erabiliz, bere lengoiaaren (*BPMN*) arauak errespetatuz, *workflow* azpi-lengoia bat sortu izango litzateke.
 - Soluzioaren sarrera/irteerak kudeatzeko *Bizagi Studioren* bitartez sortutako web-aplikazioa erabili daiteke, bere datu-baseen kudeaketa, negozio-erregelak, formularioak eta web-zerbitzuak baliatuz.

Arkitekturaren soluzio honek, ostera, bi desabantaila nagusi ditu:

- Arkitektura guztia kanpoko tresna baten bitartez eraikitzeak izugarrizko menpekotasun teknologikoa sortuko du. Garapen-prozesuaren konplexutasuna gorakorra izango denez, sistema osoa, hasieratik, *Bizagiren* menpe jartzea arriskutsua izan daiteke eta ez da batere komenigarria produktuaren aldaketa eta hobekuntzarako.
 - *Bizagi Modeler* freemium bat da, hau da, zerbitzu basikoak dohain eskaintzen ditu eta zerbitzu aurreratuago edo bereziengatik zerbait ordaindu behar da. *Bizagi Studio*, ordea, suite ofimatkia da, enpresen erabilpenerako paketeak eskaintzen ditu bere produktua erosiz. Hori dela eta, *Bizagirekin* arkitekturaren kostu ekonomikoa handia izango litzateke.



10.1. Irudia. Bizagi Modeler erabiliz sortutako prozesua.

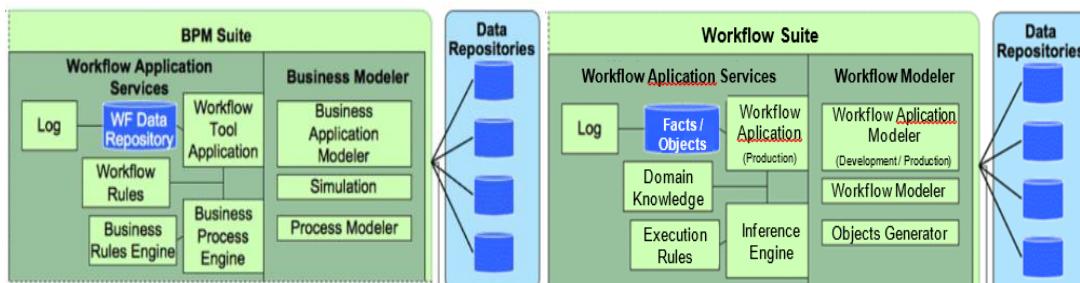
10.1.2 ProWF

Esan bezala, ProWF proiektuan erabakia **arkitektura zerotik eraikitzea** izan zen. Hurrengo kapituluan sartuko gara soluzioaren deskribapenean.

Workflowetan oinarritutako web-aplikazio bat zerotik eraikitzea apustu handia zen, denbora gehiago eman beharko litzateke arkitekturaren garapenean, baina, ordea, pisuzko arrazoia zituen:

- *Bizagiren desabantailak oso kaltegarriak ziran. Honela, menpekotasun teknologikoa saihestu eta kostu ekonomikoa asko gutxitzen da.*
- *Workflow-lengoia guztiz propioa sortzea komenigarria zen, lengoaia hori exekutatzeko motor propioa ere, bere erregela eta berezitasunekin.*
- *Gaur egun, sarrera/irteeren kudeaketarako, CMS baten erabilerarekin web-aplikazioen sormena ez da hain zaila eta soluzio profesionalak lortu daitezke.*

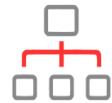
Hala ere, workflowetan oinarritutako arkitektura hau zerotik eraikitzea ez da lan erraza eta sistema konplexua bat sortu behar da, gainera, *Bizagik* eta *BPMN¹⁸* estandarrak eskaintzen dituzten funtzionalitate asko galduko ditugu. 10.2. Irudian BPMN eta workflowetan oinarritutako arkitekturak ikus daitezke.



10.2. Irudia. BPMN eta workflowetan oinarritutako arkitekturak.

10.1.3 ProMeta

ProMeta proiektuan sistemaren abstrakzio-maila igotzea erabaki da. Metodologien informazioa workflow lengoaian edo BPMNn eskuz definitu ordez, metaeredu eta ereduen bidez gordetzen da. Horrela,



ereduaren informazioa berrerabil daiteke eta beste ereduetara eraldaketa egin. Eeduaren informazioa zuzenean kodea sortzeko ere erabil daiteke, beste metaereduren beharrik gabe.

Hurbilpen honek ziurtatzen du etorkizunean aurreko aukeretako edozein edo beste batzuk aukeratu ahal izatea. Iza ere, BPMNren eta workflow-lengoaiaren metaereduak izanik, eraldaketa egitea ez litzateke horrenbeste lan izango. Gainera, eredutik datu-baserako SQL kodea edo webgenerako HTML kodea sortzeko aukera ere egongo litzateke.

Proiektu honetan saiakera bat egin da BPM eta workflow-lengoaiarik erabili gabe. Iza ere, eredutik sortutako datu-basea eta Drupal-eko moduluak bakarrik erabili dira.

10.2 CMS

ProWF proiektun, soluzioaren datu zein informazioaren sarrera/irteerak kudeatzeko web-aplikazio bat sortzea erabaki zenez, CMS bat erabiltzea adostu zen. CMS baten bitartez web-aplikazioaren administrazioa eta kudeaketa ahalbidetzen da eta itxura profesionala duen emaitza lortu daiteke.

Hasieratik *Drupal* erabiltzea gomendatu zuen proiektuaren tutoreak, Juan Manuel Pikatzak, baina Drupal erabiltzen hasi baino lehen merkatuan zeuden beste CMSak aztertu behar ziren ere. Hiru CMS aztertu ziren nagusiki: *Wordpress*, *Joomla* eta *Drupal*.

Taulan ProWF proiektuan egindako konparaketa bat ikus daiteke erabakia hartzeko gehien nabarmenzen diren puntuekin.

Ezaugarria	Wordpress	Joomla	Drupal	Oharrak
Kode irekia	✓	✓	✓	-
Dokumentazio simple eta ondo egituratuta	✓	✓	✓	-
Komunitate aktiboa eta foroak	✓	X	✓	Hemen Wordpress da nagusiena.
Estentsio gehigarri eta moduluak hedapena	✓	X	✓	<i>Joomla</i> k estentsio gehigarriak ditu ere, baina ez askorik.
Beginner-friendly (erabiltzaile berrientzako erabilerraza)	✓	X	X	<i>Joomla</i> eta <i>Drupal</i> ekin zaila izan daiteke hasieran bere konfigurazioa edo gunearren itxura aldatzen jakitea edo
Erabiltzaileen kudeaketa erraza	X	X	✓	<i>Drupal</i> en bitartez rolak sortu/esleitu eta baimen espezifikoak eman daitezke
Programazio-lengoaia	PHP	PHP	PHP	-

10.1.Taula. CMS ezberdinen ezaugarrien konparaketa.

Azkenik, azterketa sakon bat egin eta aukera bakoitza ebaluatu ostean, *Drupal* CMSa erabiltzea izan zen erabakia, hurrengo arrazoiengatik:

- Drapalen erraza da edukia gehitzea/sortzea. Eduki pertsonalizatu motak malguak dira eta aukera asko eskaintzen dituzte.
- Guneari gehitzeko hainbat modulu eskuragarri daude bere webgunean eta proiektu honetarako oso erabilgarriak diren moduluak aurkitu ziren.
- Erabiltzaileak administratzea erraza da, rol berriak sortu eta baimenak zehaztu ditzakeen sistema integratu batekin. Funtzionalitate hori oso komenigarria zen proiektu honentzat.
- Mundu mailan garrantzitsuenak diren teknologia saltzaileen sailkapenak argitaratzen dituzten Gartner eta Forrester erakundeen txostenetan, CMS atalean, liderra den Acquia enpresak Drupal erabiltzen du oinarri bezala.

ProMeta proiektuak CMSari dagokionez antzeko helburuak dituenez, aurreko arrazoi guztiak mantentzen dira. Gainera, *Drupal* erabiltzeak orain beste abantaila bat du, aurreko proiektuaren zati batzuk berrerabilzeko aukera.

10.2.1 Wordpress

<https://wordpress.com/>

<https://wordpress.org/>

<https://es.wikipedia.org/wiki/WordPress>

2003ko maiatzaren 27an jarri zen abian, edozein motatako web orrialdeak sortzera bideratuta. Jatorrian blogen sorkuntzan arrakasta handia lortu zuen, baina geroago web orrialde komertzialak sortzeko tresna nagusietako bat bilakatu zen.

WordPress PHP hizkuntzan garatzen da MySQL eta Apache exekutatzen duten inguruneetarako, GPL lizenziapean eta software librea da.

Helburu orokorreko CMS ezagunena da. 2019ko martxoan Interneteko gune guztien % 33,4k eta eduki kudeatzaleetan oinarritutako gune guztien %60,3k erabiltzen zuten.

Arrakastaren arrazoieta bat garatzaile eta diseinatzaileen komunitate izugarria da, bere muinean programatzeaz edo komunitatearentzako pluginak eta txantiloia sortzeaz arduratzen dena.

10.2.2 Drupal

<https://www.drupal.org/>

<https://es.wikipedia.org/wiki/Drupal>

Doakoa, modularra, erabilera anitzekoa eta oso konfiguragarria da. Artikuluak, irudiak, fitxategiak argitaratzea ahalbidetzen du eta beste zerbitzu gehigarri batzuk ere eskaintzen ditu, hala nola foroak, inkestak, bozketak, blogak, erabiltzaileen administrazioa eta baimenak.

Drupal sistema dinamikoa da: bere edukia zerbitzariaren fitxategi estatikoetan gorde beharrean, orrien testu edukia eta bestelako ezarpenak datu base batean gordetzen dira eta web ingurunea erabiliz editatzen dira.

Doako programa da, GNU/GPL lizenziarekin, PHP-n idatzia eta MySQL-rekin bateragarria. Erabiltzaileen komunitate aktibo batek garatu eta mantentzen du. Aipagarria da kodearen eta sortutako orrien kalitatea, web estandarrak errespetatzea eta sistema osoaren erabilgarritasuna eta koherenzia.

Drupal-en diseinua bereziki egokia da Interneteko komunitateak eraikitzeko eta kudeatzeko. Malgutasun eta moldagarritasunagatik nabarmentzen da, baita eskuragarri dauden modulu osagarrien kopuru handiagatik ere, webgune mota ugari egiteko egokia da.

10.2.3 Joomla

<https://www.joomla.org/>

<https://es.wikipedia.org/wiki/Joomla>

Webgune dinamiko eta interaktiboak garatzeko aukera ematen du. Webgune bateko edukia modu errazean sortu, aldatu edo ezabatzeko aukera ematen du administrazio panel baten bidez. Kode irekiko softwarea da, PHP-n programatua edo garatua eta GNU General Public License (GPL) lizenziapean argitaratua.

Bere funtzionamendurako datu-base kudeatzailearekin sortutako datu-basea behar du (MySQL da ohikoenai), baita Apache HTTP zerbitzaria ere.

Ondo eratutako HTML kodea sortzea, blogen kudeaketa, artikuluak inprimatzeko ikuspegiak, albisten flash-a, foroak, inkestak (inkestak), egutegiak, gune bilaketak integratuak eta hizkuntza anitzeko lagunza dira Joomla-rekin sor daitezkeen tresnetako batzuk. Gaur egungo joerak direla eta apustu handia egiten ari da merkataritza elektronikoaren alde.

10.3 Dokumentazioa Hostinga

Proiektuaren webgunea eta sortutako produktuaren webgunea interneten publikoki eskuragarri egon daitezen hosting zerbitzu bat erabili behar da. Hauek batera publikatu daitezke edo aparteko webgune moduan. Doako aukerak bakarrik aztertu dira.

10.3.1 GitHub Pages

<https://pages.github.com/>

GitHubekin integratuta, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan. Konfigurazio oso erraza, errepositoriotik bertatik egiten da. Webgune estatikoak bakarrik onartzen ditu, beraz, drupal-erako ez du balio, PHP erabiltzen baitu. Webguneak HTML, CSS, eta JavaScript-en idatzita egon daitezke edo Jekyll webgune estatiko sortzailea erabiliz.

10.3.2 GitLab Pages

<https://docs.gitlab.com/ee/user/project/pages/>

GitHub Pages-en antzekoa da, baina aukera gehiago eskaintzen ditu. Webgune estatiko sortzaile guztiak onartzen ditu, Gastby, Jekyll, Hugo, etab. Hori bai, webgune guztiak eraikitzeko Continuous Integration fitxategi bat behar da. GitHub-en ez da beharrezkoa, aukeretan aktibatzea nahikoa da. Gure webguneak ez duenez erabiltzen webgune estatiko sortzailerik, GitHub Pages erabiltzea hobe da simpleagoa delako.

10.3.3 Netlify

<https://www.netlify.com/>

GitHub-ekin integratu daiteke, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan. Aurrekoek bezala, webgune estatikoak bakarrik onartzen ditu baina aukera zabalagoak ditu. Kasu honetan webgune simple bat eraikitzea bakarrik interesatzen zaigunez, aurreko aukerekin nahikoa daukagu.

10.4 Drupal Hostinga

10.4.1 000webhost

<https://www.000webhost.com/>

Ez dauka GitHubekin integratziorik eta beraz kodea eskuz igo beharko litzateke aldi bakoitzean. PHP kodea onartzen du, eta beraz Drupalerako balio du. MySQL datu-basea eskaintzen du,xampp-ekin bateragarria. ProWF proiektuan aukera hau erabili da proiektuaren webgunerako. Drupal webgunea ez da eskuragarri jarri.

10.4.2 Heroku

<https://www.heroku.com/>

GitHubekin integratu daiteke, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan <https://devcenter.heroku.com/articles/github-integration>. Konfigurazioa GitHub Pages-ena baino zailagoa da. PHP kodea onartzen du eta beraz, Drupalerako balio du. Defektuz PostgreSQL datu-basea eskaintzen du. Xampp-ek, berriz, MySQL eskaintzen du. Drupal modu simplean instalatzeko ez dago prestatuta.

10.4.3 Acquia

Ez dauka doako planik <https://www.acquia.com/choosing-right-acquia-cloud-platform>. Drupal modu simplean instalatzeko prestatuta dago.

10.4.4 Pantheon

Doako plana dauka <https://pantheon.io/plans/pricing>. GitHub-ekin ondo integra daiteke, nahiz eta ez den horren simplea <https://pantheon.io/docs/guides/collaborative-development>. Drupal modu simplean instalatzeko prestatuta dago. Drupal-erako honek ematen du aukera onena.

10.4.5 Platform.sh

Ez dauka doako planik <https://platform.sh/pricing/>. Drupal modu simplean instalatzeko prestatuta dago.

10.5 Datu-base Kudeaketa Sistema

10.5.1 MySQL

Drupal-ekin bateragarria da, ematen dituen aukeretako bat da. Xampp-ek eskaintzen duen datu-basea da, eta beraz garapen lokalerako egokia. Xampp-eko phpMyAdmin bidez kontrola daiteke. Heroku erabiltzen badugu oso mugatuta dago, 5MB bakarrik eskaintzen dira doan <https://elements.heroku.com/addons/cleardb>.

10.5.2 PostgreSQL

Drupal-ekin bateragarria da, ematen dituen aukeretako bat da. Xampp-ek defektuz ez du eskaintzen, baina gehigarri moduan aktibatzeko aukera ematen du. Xampp-eko phpPgAdmin gehigarriaren bidez kontrola daiteke <https://beecreativos.com/postgresql-en-servidor-local-con-wamp-y-xampp/>. Heroku erabiltzen bada aukera egokia da, 1GB eskaintzen ditu <https://elements.heroku.com/addons/heroku-postgresql>. Webgunea Heroku-n eduki nahi badugu hau da aukera onena.

10.6 Datu-basearen Sorrera

10.6.1 Inferentzia Motorra

ProWF proiektuan inferentzia motorra erabiltzen da datu-basearen sorrerarako. Inferentzia motorra erabiltzen ez badugu bide honek ez dauka zentzurik.

10.6.2 Teneo

Teneo-k EMF eredu eta datu-base erlazionalen arteko mapaketa eskaintzen du <https://wiki.eclipse.org/Teneo>. Hibernate-rekin bateragarria da, objektu eta datu-base erlazional arteko mapaketarako tresna <https://hibernate.org/>. Aukera honek datu-basea automatikoki sortuko luke. Baina, datu-basea oso handia izango litzateke eta beharrezkoa baina elementu gehiago izango lituzke. Gainera, Teneo zaharkitua dago eta Eclipse-tena eta EMF-ren bertsio berriekin ez da bateragarria.

10.6.3 Xtext

Aukera onena Xtext eta Xtend erabiliz ereduko datuekin SQL INSERT-ak sortzea da. Honek lan dezente eskatuko luke. Baina, flexibilitate asko emango luke, datu-basea nahi dugun bezala diseina dezakegu. Horrela, behar dugun informazioa bakarrik izango dugu datu-basean, eta errazagoa izango da honekin lan egitea.

10.7 Lanerako Ingurunea

Lanerako ingurune egokia aukeratzea garrantzitsua da. Baita ere ingurunearekin arazoren bat badago alternatibak izatea.

10.7.1 Makina Birtuala

Juanmak lanerako makina birtuala eskaini dit. Ordenagailu horretan aurreko proiektuak daude eta erabilitako softwarea instalatuta. Honetara VPN bidez konektatu beharko nintzateke. Honen arazoa da lokalean lan egitean baino makalago joango dela.

10.7.2 Ordenagailu Pertsonala

Ordenagailu pertsonala nire gustura konfiguratuta daukat eta erabiltzen dudan softwarea instalatuta. Makina birtualean softwarea instala daiteke baina errazagoa da nire konputagailuan falta den softwarea instatzea. Izan ere, Drupal eta EHSIS bakarrik falta dira. Hori bai, bukaeran sistema zerbitzariko makina birtualera pasatuko da, besteek eskuragarri eduki dezaten.

10.8 Bertsio Kontrola

Bertsio kontrola oso garrantzitsua da, batik bat horrelako proiektyuetan.

10.8.1 GitHub

Bertsio kontrolerako Git eta GitHub erabiltzea da aukera onena. Informazio guztia GitHub-eko errepositorio batean egongo da eta webgunea publikatuta egongo da. Horrela, beharrezkoa denean Juanmari erakutsi ahal izango diot egiten ari naizena. Gainera, bertsio kontrolak segurtasuna eta trazabilitatea ematen du.

10.8.2 GitLab

GitLab erabiltzea ere aukera ona izan daiteke, antzeko aukerak eskaintzen ditu. GitHub-ekin praktika gehiago daukat eta beraz ez dauka zentzurik GitLab-era aldatzeak. GitLab-ek bakarrik eskaintzen duen zerbait beharko bagenu orduan bai.

10.9 Metaeredua

10.9.1 SPEM

Software Process Engineering metamodel. SPEM metaeredua cmof formatuan dago.

10.9.2 UMA

Unified Method Architecture. UMA metaeredua ecore formatuan dago.

10.10 Metodologia

10.10.1 OpenUp

OpenUP softwarea garatzeko metodo eta prozesu bat da, teknologien sektoreko enpresa multzo batek proposatutakoa, zeintzuk 2007an Eclipse Fundazioari dohaintzan eman zioten. Fundazioak lizentzia libre bezala arigitaratu du eta eredu gisa mantentzen du Eclipse Process Framework (EPF) proiektuaren barruan.

Metodologia honek garrantzi handia izan du proiektu osoan zehar. Batetik, proiektuaren helburuetako bat metodologia baten definizioa eta ezarpena izan da eta OpenUP izan da aukeratutako metodologia. Bestetik, proiektuaren elaborazio prozesurako OpenUP metodologia jarraitu da, dokumentazioa bilduz eta proiektuaren kontrola eramanez.

10.10.2 RUP

Kontuan hartuta proiektua eta proietkuko taldea txikiak direla, OpenUP metodologia erabiltzea nahikoa da. Gainera, metodologiako lehenengo bi edo hiru fase egitearekin nahikoa izango da. Talde handiagoa edo proiektu konplexuagoa izango balitz agian RUP bezalako metodologia konplexuagoa bat beharko genuke.

Metodologiaren ezarpenari dagokionez, simpleagoa da OpenUP metodologia ezartzea txikiagoa delako. Gainera, OpenUP metodologiarekin lan egiteko Eclipseren tresnak erabil daitezke. Tresna horiek libreak dira, eta egileak esperientzia du horiek erabiltzen.

10.10.3 ABRD

Agile Business Rules Development metodologiaren ereduak ere erabili da. Horrela, bi metodologia erabiliz ziurtatzentz da sistemaren egitura egokia dela.

10.11 Metodología Editorea

10.11.1 EPF Composer

Metodologia definitzeko aukera argiena da, horretarako egindako tresna baita. Jadanik hainbat metodologia definituta daude tresna honekin: OpenUp, ABRD, Scrum... Metodologia horietako bat erabili nahi badugu ez daukagu ezer diseinatu beharrik. Horiek oinarri bezala hartuta ere errazagoa da beste metodologia bat definitzea. Metodologiaren webgunea sortzeko aukera ematen du, dokumentazio moduan erabili ahal izateko. Nahiko zaharkitua dago, Java 8 32 bitemko bertsioa eskatzen du.

10.11.2 Rational Method Composer

EPF Composer-en antzeko tresna, baina IBMrena da eta ordainpekoa. RUP metodologia du oinarrian eta horrekin lana egin nahi badugu aukera interesgarria izan daiteke.

10.11.3 Editore Grafiko

EPF Composer baino simpleagoa den editorea edukitzea ondo etorriko litzateke. Ikono berdinak partekatuko lituzke, zuhaitz egitura simpleagoa edukiko luke.

10.11.4 Testu Editorea

Editore grafikoa baino simpleagoa den testu editorea bat edukitzea ere komenigarria da. Batzuetan, editore grafikoa baino erosoagoa da testu editorea erabiltzea. Onena ereduak bi editoreekin aldatzeko aukera izatea izango litzateke, eta edozein momentutan bien artean aldatzeko aukera izatea.

10.12 Prozesua Bistaratu

10.12.1 DOT

ProWF proiektuan prozesua definitzeko eta bistaratzen DOT lengoia erabili da. Beraz, aukera ona izan daiteke prozesua bistaratzen. Hori bai, honek lan dezente gehituko luke. Izan ere, eredu DOT lengoaiara pasatzea eskatuko luke metaereduak erabiliz.

10.12.2 XPDL

XPDL ere aukera egokia izan daiteke estandarra delako. Baino, oraingoz DOT aukera hobea izango litzateke, simpleagoa delako.

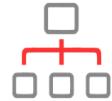
10.12.3 Webgunea

Prozesua bistaratzen EPF Composer-ekin publikatzen den webgunea erabiltzea da aukera onena. Izan ere, webguneak metodologiaren informazio guztia dauka, eta honen sorrera automatikoa da. Beraz, prozesua EPF Composer-ekin definitza komenigarria da.

10.13 Denboraren Kontrola

10.13.1 Toggl Track

Temporizadore baten bidez ataza bakoitzean pasatako denbora neurtu daiteke. Eskuz ere aldatu daiteke denbora ahaztu egin bazaigu. Abisatu egiten du inaktibo bagaude eta denbora neurten ari bagara. Aspaldian denborarik ez badugu neurtu ere abisatzen du, ez ahazteko. Denbora proiektutan, atazatzen eta kategoriatan sailkatu daiteke. Edozein momentuko denboraren estatistikak ikusi eta deskargatu daitezke. Gailu mota guztieta rako aplikazioak daude, mugikorrerako ordenagailurako, nabigatzailerako, etab.



10.13.2 Clockify

Toggl Track-en antzeko funtzionamendua dauka. Doako planean aukera batzuk gutxiago eskaintzen ditu.

10.13.3 WakaTime

Plugin moduan instalatu daiteke editore askotan. Gure kasuan VSCode eta Eclipse editoreetan instalatu da, horiek izan baitira erabili diren editoreak. Denbora automatikoki neurten du eta metrikak sortzen ditu. Adibidez, programazio-lengoa eta proiektu bakoitzean pasatako denbora erakusten du. Ez du balio editoreetatik kanpo pasatako denbora neurteko. Hala ere, Toggl Track-en osagarri ona da, estatistika zehatzagoak eskaintzen dituelako. Doako planean 2 asteko historia bakarrik erakusten du.

10.14 Diagramak

10.14.1 PlantUML

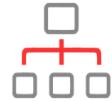
Hainbat motatako diagramak sortzeko aukera ematen duen softwarea da. Eskuz sortzen dira diagramak eta bi aukera daude aplikazioa erabiltzeko: online edo offline. Proiektuan zehar aplikazio hau erabili izan da dokumentazioan agertzen diren diagrama gehienak egiteko, bezeroak dokumentazio ulergarria jaso dezan.

10.14.2 Draw.io

Hainbat motatako diagramak sortzeko aukera ematen duen softwarea da. Eskuz sortzen dira diagramak eta bi aukera daude aplikazioa erabiltzeko: online edo offline. Proiektuan zehar aplikazio hau erabili izan da dokumentazioan agertzen diren diagrama batzuk egiteko.

10.14.3 UMLDoclet

Javadoc metadatuak erabiltzen ditu PlantUML diagramak automatikoki sortzeko eta HTML dokumentazioan txertatzeko. Diagramak irudi klikagarri gisa txertatzen dira eta pakete eta klaseko dokumentaziora estekatzen dira eskuragarri dagoenean. Pakete dependentzia, pakete eta klase diagramak sortzen ditu elementu guztiarako.



11 Proposatutako Sistemaren Deskribapena

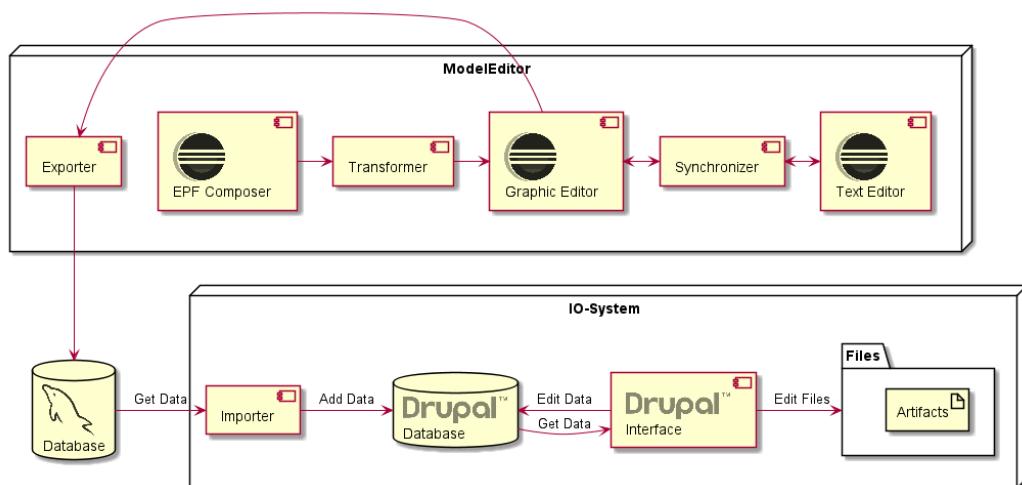
Kapitulu honetan planteatutako arazoa konpontzeko proposatzen den sistema, bere osagaiaak eta bere ezaugarriak deskribatzen dira.

11.1 Azpisistemak

ProWF sistema bi azpisistema ezberdinetan bananduta egongo da: **ModelEditor** eta **IO-System**.

- **ModelEditor**: sortutako editore grafikoa eta testu editorea erabiliz prozesuaren eredua sortzeko eta editatzeko aukera emango du. Sistema honen ardura prozesu ingeniarri rolak izango du.
- **IO-System**: CMS baten bitartez kudeatutako web-aplikazioa izango da. Helburua metodologia jarraitzen duten projektuen informazioa gordetzea da. Rol bakoitzak metodologian dituen ataza berdinak bete beharko ditu.

Idea orokor bat egiteko, 11.1. Irudian sistema osoaren arkitektura ikus daiteke, azpisistematan banatuta.



11.1. Irudia. ModelEditor eta IO-System azpisistemen arkitektura.

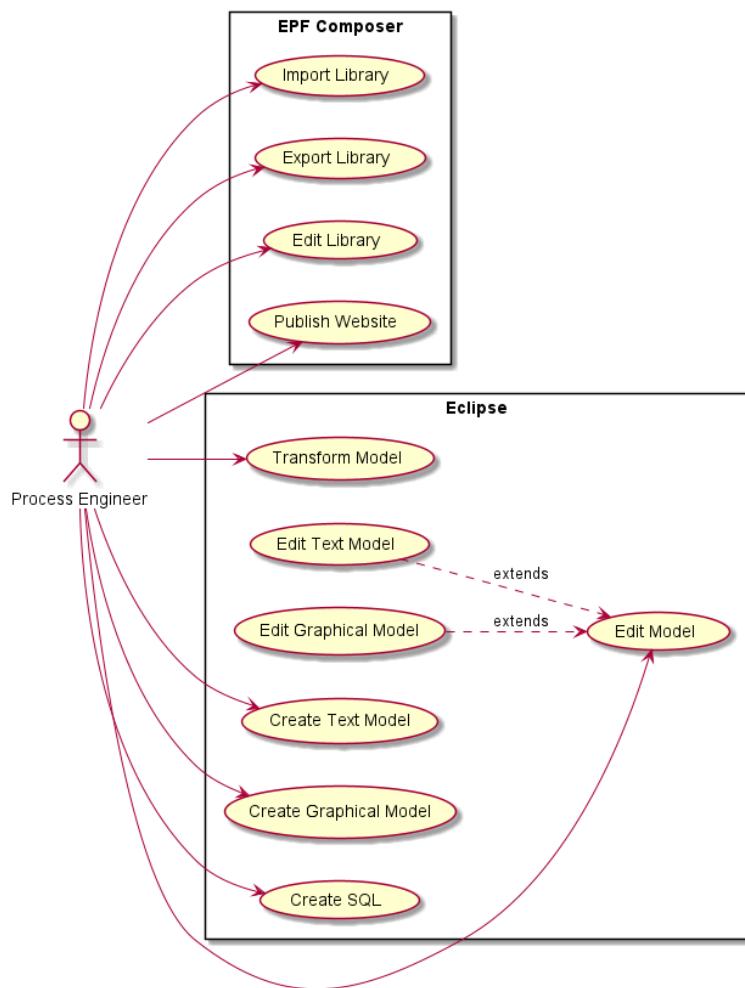
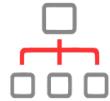
Ondoren, azpisistema bakoitzaren hainbat aspektua azalduko ditugu: arkitektura, analisia, diseinua, garapena, proba eta hedapena.

Bukatzeko, etorkizunerako hobekuntza posibleak aurkeztuko dira. Horrela, proiektu honi jarraipena ematea erraztuko da.

11.2 ModelEditor

11.2.1 Betekizunak

11.2. Irudian ModelEditor azpisistemako erabilpen kasuen ereduak ikus daitezke.



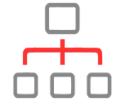
11.2. Irudia. ModelEditor azpisistemako erabilpen kasuen eredu.

11.2.2 Analisia

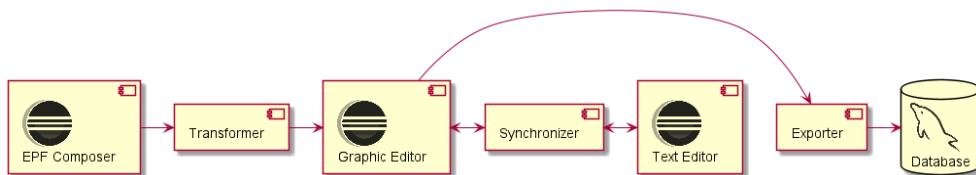
11.2.3 Arkitektura

ModelEditor azpisistemak honako osagaiak izango ditu. 11.3. Irudian ikus daitezke osagaia hauen arteko erlazioak.

- **EPF Composer:** Metodologiak modu grafikoan editatzeko aukera ematen du. Metodologiaren webgunea sortu daiteke bertatik.
- **Formatu Aldatzailea:** Metodologiaren formatua aldatzen du XMLtik UMara editore grafikoak erabili ahal izan dezan.
- **Editore Grafikoa:** Metodologiak modu grafikoan editatzeko aukera ematen du, metodologiaren jatorrizko ikonoak erabiliz.
- **Testu Editorea:** Metodologiak testu bidez editatzeko aukera ematen du, metaeredutik sortutako gramatika erabiliz.
- **Editore Sinkronizatzalea:** Metodologiaren informazioa editore batekin aldatutakoan bestea ere aldatzeaz arduratzen da.



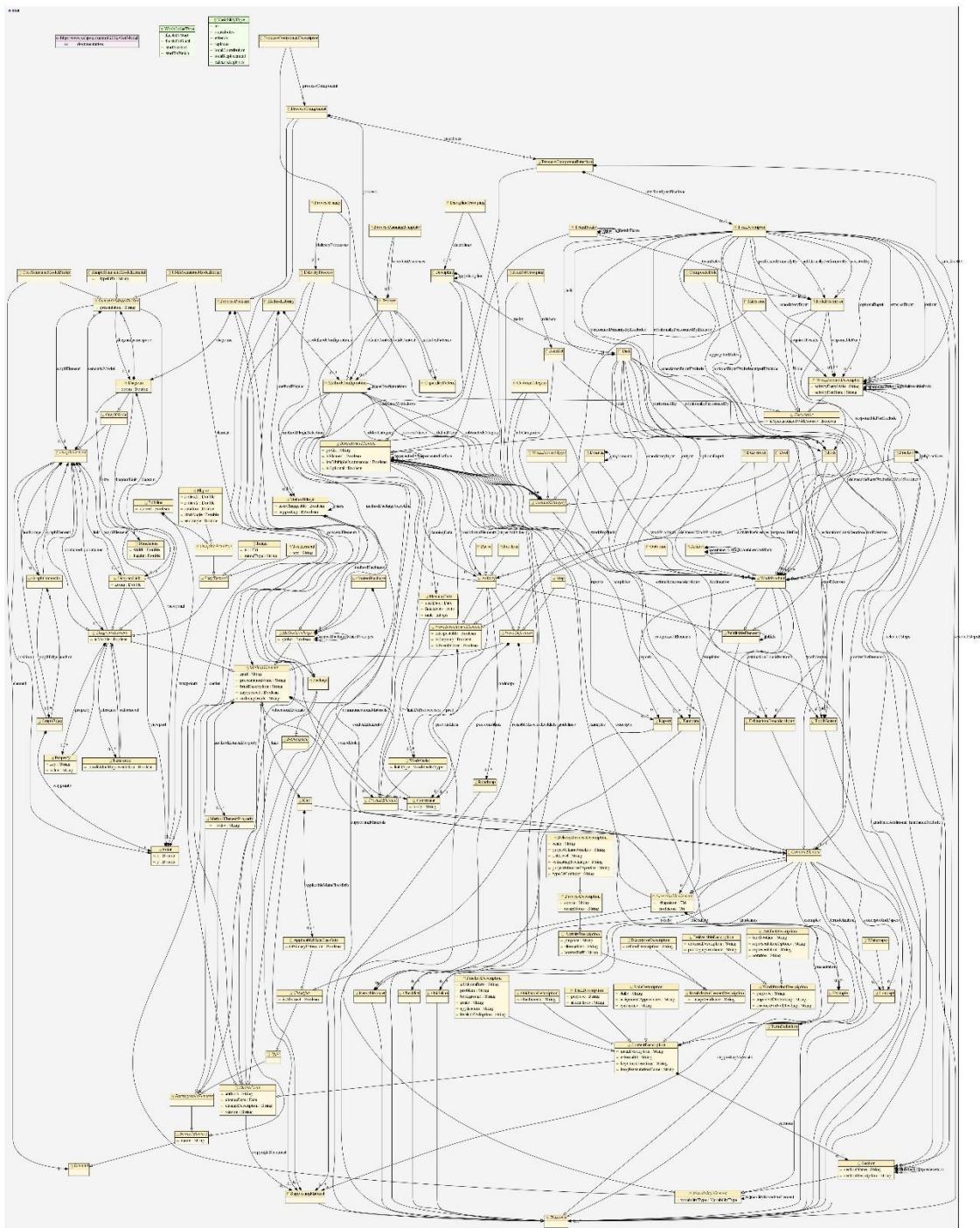
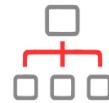
- **Kode Sortzailea:** Metodologien informazioa datu-basean gorde ahal izateko SQL kodea sortzen du.



11.3. Irudia. ModelEditor azpisistemaren arkitektura.

11.2.4 Diseinua

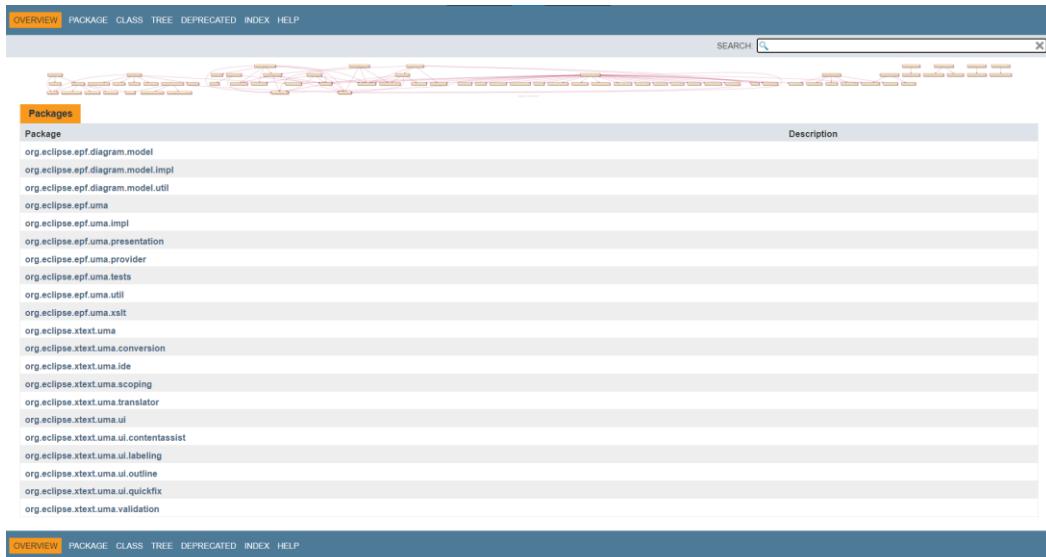
Diagrama proiektuaren webgunean ikus daiteke.



11.2.5 Garapena

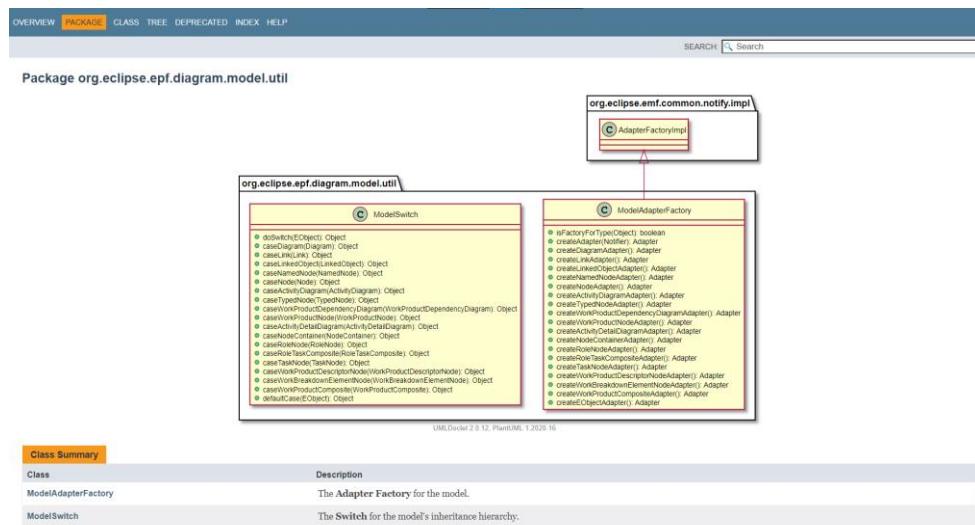
UMLDoclet tresnak Javadoc metadatuak erabiltzen ditu PlantUML diagramak automatikoki sortzeko eta HTML dokumentazioan txertatzeko. Diagramak irudi klikagarri gisa txertatzen dira eta pakete eta klaseko dokumentaziora estekatzen dira eskuragarri dagoenean. Pakete dependentzia, pakete eta klase diagramak sortzen ditu elementu guztiarako.

11.4. Irudian ModelEditor azpisistemako dokumentazio webgunearen hasierako orria agertzen da. Bertan paketeak zerrendatzen dira eta goiko aldean paketeen arteko dependentzia diagrama ikus daiteke.



Pakete bat aukeratzen badugu, diagrama eta klaseak bistaratuko dira. 11.5. Irudian adibide bat ikus daiteke.

11.4. Irudia. ModelEditor-en paketeak eta paketeen arteko dependentziak.



11.5. Irudia. Pakete diagrama adibidea.

Klase bat aukeratzen badugu, bere diagrama eta metodoak bistaratuko dira. 11.6. Irudian ikus daiteke klase diagrama baten adibidea.



OVERVIEW PACKAGE CLASS TREE DEPRECATED INDEX HELP

SUMMARY NESTED | FIELD | CONSTR | METHOD DETAIL FIELD | CONSTR | METHOD SEARCH:

Package org.eclipse.epf.diagram.model.util

Class ModelSwitch

```
java.lang.Object
org.eclipse.epf.diagram.model.util.ModelSwitch
public class ModelSwitch
extends java.lang.Object
```

The **Switch** for the model's inheritance hierarchy. It supports the call `doSwitch(object)` to invoke the `caseXXX` method for each class of the model, starting with the actual class of the object and proceeding up the inheritance hierarchy until a non-null result is returned, which is the result of the switch.

See Also:
ModelPackage

Constructor Summary

Constructors	Description
<code>ModelSwitch()</code>	Creates an instance of the switch.

Method Summary

All Methods	Instance Methods	Concrete Methods	Description
Modifier and Type	Method		
<code>java.lang.Object</code>	<code>caseActivityDetailDiagram(ActivityDetailDiagram object)</code>		Returns the result of interpreting the object as an instance of ' <code>Activity Detail Diagram</code> '.
<code>java.lang.Object</code>	<code>caseActivityDiagram(ActivityDiagram object)</code>		Returns the result of interpreting the object as an instance of ' <code>Activity Diagram</code> '.

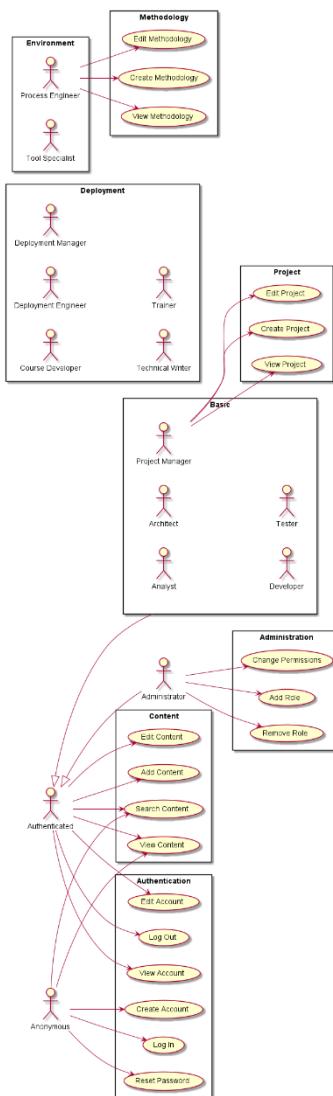
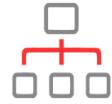
11.6. Irudia. Klase diagrama adibidea.

11.2.6 Proba

11.3 IO-System

11.3.1 Betezizunak

11.7. Irudian IO-System azpisistemako erabilpen kasuen ereduagertzen da.



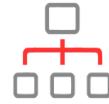
11.7. Irudia. IO-System azpisistemako erabilpen kasuen eredua.

11.3.2 Analisia

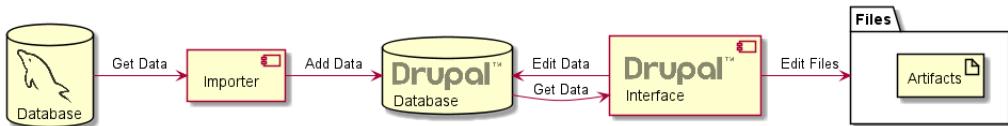
11.3.3 Arkitektura

IO-System azpisistemak honako osagaiak izango ditu. 11.8. Irudian osagaien arteko loturak ikus daitezke.

- **Datu-basea:** Datu-base erlazionalak ereduaren informazio garrantzitsuena gordeko du, Drupal webgunerako beharrezkoa izan daitekeena. Hau da, OpenUP eta ABRD metodologien faseak, iterazioak, jarduerak, atazak, artefaktuak, rolak, etab.
- **Drupal datu-basea:** Drupal sistemaren datu-basea eduki guztia gordetzeaz arduratzen da, fitxategiak izan ezik. Adibidez, erabiltzaileak, rolak, baimenak eta eduki motak gordetzean dira.
- **Drupal datu importatzalea:** Datu-importatzalea Drupal modulu multzo bat izango da. Hauen ardura aurretik aipatutako lehenengodatu-basetik Drupal datu-basera edukia importatzea da. Horretarako, nodoak sortu beharko dira, eta Drupal arduratuko da edukia gordetzeaz.
- **Drupal fitxategiak:** Fitxategiak datu-basetik kanpo gordeko dira. Gure kasuan fitxategi gehienak artefaktuei dagozkienak izango dira, DOC eta PDF dokumentuak.

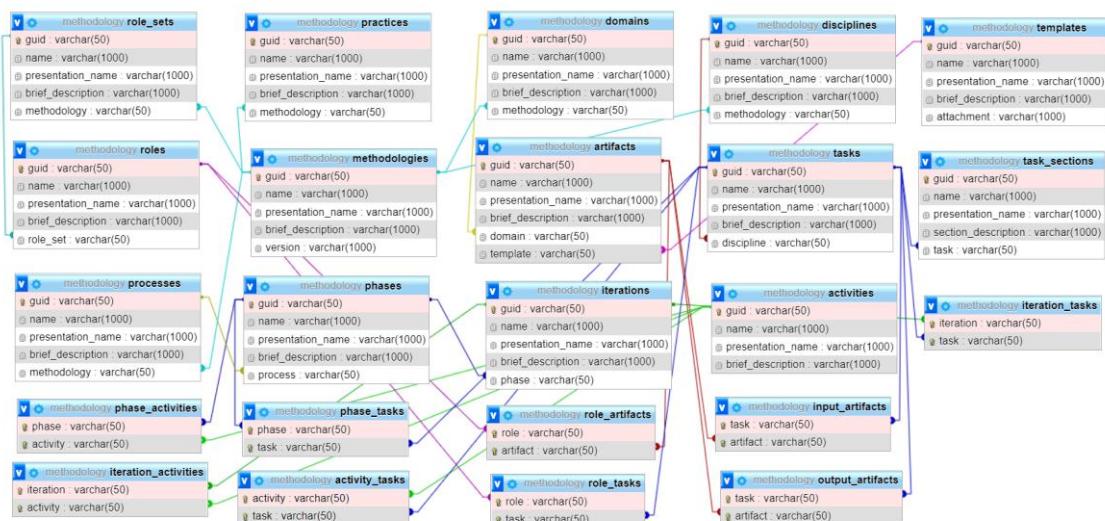


- Drupal interfazea:** Interfazean edukia bistaratu eta aldatzeko aukera guztiak egongo dira. Esan bezala, edukia Drupal datu-basean gordeko da. Gainera, erabiltzailearen kontuekin zerikusia duten aukerak ere egongo dira. Horrez gain, administratzaleak aukera gehiagarri asko izango ditu webgunea kudeatzeko interfaze bidez.



11.8. Irudia. IO-System azpisistemaren arkitektura.

11.3.4 Diseinua



11.3.5 Garapena

11.3.6 Proba

11.4 Hedapena

Proiectuaren dokumentazioa eta implemetazioa publikoki eskuragarri egongo dira GitHub bidez eta webguneetan. Printzipioz, lana bukatu ondoren ere eskuragarri jarraituko dute, edozeinek kontsultatu ahal izan ditzan.

Proiectuaren dokumentazioaren kidea GitHub-en egongo da eskuragarri: <https://github.com/juletx/ProMeta>. Webgune hori automatikoki eraikiko da aldaketa bakoitzarekin <https://juletx.github.io/ProMeta> GitHub Pages erabiliz. GitHub Pages aukera ona da kasu honetan webgunea estatikoa delako.

Aurreko bi proiecturen webguneekin ere berdina egin dut, ProWF eta BETRADOK. ProWF proiectu honen aurrekaria denez kontsultatzeko behar dut. Eta BETRADOK proiectua antzekoa denez ongi etorriko zait ideiak hartzeko. ProWF proiectuaren errepositorioa <https://github.com/juletx/BETRADOK> eta webgunea <https://juletx.github.io/ProWF/>. BETRADOK proiectuaren GitHub errepositorioa <https://github.com/juletx/BETRADOK> eta GitHub Pages webgunea <https://juletx.github.io/BETRADOK/>.

Proiectuaren metaereduen atalaren implementazioaren kidea ere GitHub-eko errepositorio batean dago: <https://github.com/juletx/ProMeta-ModelEditor>. Kodearen dokumentaziorako webgune bat erabiliko da, aurreko kasuetan bezala GitHub Pages erabiliz <https://juletx.github.io/ProMeta-ModelEditor>.

Prozesuaren webguneak ere aparteko GitHub errepositorioa edukiko du: <https://github.com/juletx/ProMeta-IO-System>. ProWF proiektuaren IO-System ere errepositorio batean jarriko da: <https://github.com/juletx/ProWF-IO-System>.

Dokumentazioarekin egiten den bezala, ondo egongo litzateke webgunea aldaketa bakoitzarekin automatikoki eraikitzea. Edo gutxienez Git-en bidez kontrolatu ahal izatea kode lokala eta zerbitzarikoa. Kasu honetan webgunea dinamikoa denez, beste hosting bat aurkitu beharko da, Drupal-erako balio duena.

Aukeren azterketa sakona egin eta gero, [Pantheon](#) erabiltzea erabaki dut. Honek 3 webgune sortzeko aukera ematen du garapena errazteko: Development <https://dev-prometa.pantheonsite.io/>, Test <https://test-prometa.pantheonsite.io/> eta Live <https://live-prometa.pantheonsite.io/>. Development webgunea garapenerako erabiltzen da. Live webgunea erabiltzaileek edukia gehitzeko da. Test webgunea Development-eko hobekuntzak probatzeko erabiltzen da, Live webguneko edukiarekin. Webguneak Test-en funtzionatzen badu, Live-n ere funtzionatuko du.

Estrategia honekin ziurtatzen da nik eta tutoreak uneoro atal bakoitzaren azkenengo bertsioa konsultatu dezakegula. Honek tutorearekin errebisioak egitea errazten du. Gainera, git bertsio kontrolari esker egindako aldaketa guztiak ikus daitezke. Horrez gain, webguneak automatikoki eraikitzeak lana errazten du, ez baitaukat zerbitzari batera igotzen ibili beharrik aldaketak dauden bakoitzean.

Proiektua amaitutakoan, lana [GAUREn](#) matrikulatu behar da eta zuzendariak oniritzia eman behar du. Ondoren, ikasleak lana [ADDI](#) plataforma igoko du. Horrez gain, ikasleak bere lanaren posterra bidali behar du dif.gral@ehu.eus helbidera. Gainera, zuzendariak emandako makinara igoko da lana, proiektu honi jarraipena emateko eskuragarri egon dadin.

Proiektu berriekin domeinu honetan sakondu eta emaitza hobeak lortu ahal izateko, orain arte bezala, proiektu honen emaitzen **jabetza intelektuala** partekatua izango da egile eta tutorearen artean.

11.5 Etorkizunerako Hobekuntzak

12 Arriskuen Analisia

Atal honetan proiektuan zehar identifikatutako arriskuen inguruko analisia egiten da.

12.1 Arriskuak

12.1. Taulan identifikatutako arriskuen zehaztasunak agertzen dira. Proiektuaren fase guztietai identifikatutako arriskuak zerrendatzen dira eta horien probabilitatea eta inpaktua zehazten da, arazoen magnitudea neurtzeko. Gainera, arriskuei aurre egiteko mitigazio estrategiak definitzen dira.

ID	Data	Izena	Deskribapena	Inpaktu	Probabilitatea	Magnitudea	Mitigazio Estrategia
1	25/02/2021	Lan ingurunearen prestakuntza	Lan ingurunearen instalazioan eta prestakuntzan agertu daitezken arazoak eta denbora galerak.	2	50%	1,0	Instalazioan zehar egindako dokumentu batean idatzi, instalatuko dudan softwarearen espezifikazioak ondo irakurri.
2	25/02/2021	Lan tresnekin arazoak	Lan tresnak erabiltzerakoan ager daitezkeen arazoak: bertsio	4	40%	1,6	Instalatzerakoan ongi ziurtatu bertsioak bateragarriak direla. Beharrezkoak diren

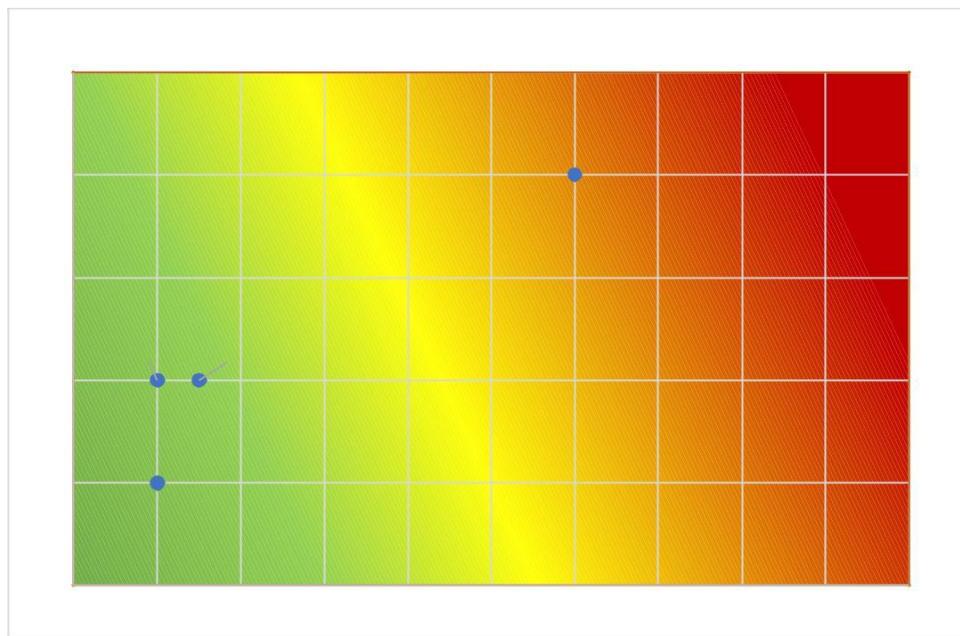
			bateraezintasunak, pluginak, erroreak...				pluginak bakarrik instalatu.
3	25/02/2021	OpenUp ereduarekin arazoak	OpenUp eredua zaharra denez, baliteke guztiz osatua ez egotea eta bateraezintasunak egotea UMA metaereduarekin.	3	60%	1,8	Ahalik eta bateragarrienak diren OpenUp eta UMA bertsiaoak aukeratu, ahal bada berrienak.
4	25/02/2021	Xtext-ekin arazoak	Xtext tresnarekin arazoak. Xtext tresnak askotan erroreak sortzen ditu. Normalean erraz konpontzen dira, baina gure eredua oso handia denez, zailagoa izan daiteke.	4	80%	3,2	Xtext-ekin lan egitean aldaketak murriztu. Errorerik ez dagoenean lana aurreratzeko aprobetxatu.
5	25/02/2021	Tresnen aukeraketa okerra	Tresna okerra aukeratzeak lana asko atzeratu dezake. Izan ere, honek tresna aldatu beharra edo lana okerrago egitea eragin dezake.	3	50%	1,5	Aukeren azterketa sakona egin tresnentzako. Beharrezko bada hainbat tresna probatu, egokiena zein den jakiteko.
6	25/02/2021	Memoriaren eta eranskinen arteko koherentzia falta	CCII-2016N-02 eta OpenUP metodologia jarraitu denez, eranskinetan dokumentu asko daude eta koherentzi faltak agertu daitezke dokumentu guztien artean.	1	50%	0,5	Eranskin guztiak berrikusi, batez ere, proiektuaren hasieran idatzitakoak. Memoriak eranskinekiko lehentasuna duela zehaztu.
7	25/05/2021	Memoria idazteko denbora falta	Gerta daiteke memoria idazteko denbora nahikoa ez izatea, bukaerarako uzten bada. Oso garrantzitsua da momoria ondo idaztea.	5	50%	2,5	Memoria osoa ez utzi bukerarako, pixkanaka kapitulu batzuk idazten joan nahiz eta proiektua bukatu gabe egon.
8	25/05/2021	Proiektua amaitzeko denbora falta	Baliteke proiektua amaitzeako denboraz juxtu ibiltzea. Horrek kalitatea jaistea eragin dezake, lana presaka egiteagatik.	4	75%	3,0	Lana modu egokian antolatu eta konstantea izaten saiatu. Denbora aprobetxatu eta gauza garrantzisuenetan zentratu. Ez dago dena bukatu beharrik, etorkizuneko lan bezala utz daiteke.
9	25/05/2021	Egindako lanaren galera	Egindako lana galtzeak lana berregin behar izatea eragin dezake. Honek denbora galtzeaz gain frustrazioa eragiten du.	3	50%	1,5	Bertsio kontrola erabili proiektuaren informazio guzgia gordetzeko. Datu guztien segurtasun kopiak egin egunero.
10	25/05/2021	Ordenagailuar ekin arazoak	Nire ordenagailuarekin arazoak izateak denbora galtzeak eragin ditzake. Ordenagailua konpondu bitartean lana egin ahal ez izatea eragin dezake.	3	25%	0,8	Alternatiba moduan lanerako tutoreak emandako makina birtuala edukitzea.

11	25/05/2021	Makina birtualarekin arazoak	Makina birtualekin arazoak izateak lanerako ingurune hori eskuragarri ez izatea eragin dezake. Nire kasuan ordenagailu pertsonala erabili dudanez ez dauka eragin handirik.	2	50%	1,0	Ordenagailu pertsonala erabili garapenerako ingurune nagusi moduan. Makina birtuala alternatiba moduan eduki arazoren bat badago erabiltzeko.
12	25/05/2021	Proiektuaren planteamendu aldaketa	Proiektuan ziurgabetasun handia dagoenez, hurbilpen aldaketak egon daitezke. Honek atzerapenak eragin ditzake.	3	75%	2,3	Proiektuaren osagaiak ahalik eta flexibleenak egiten saiatu. Horrela, hurbilpena aldatu arren egindako lana aprobetxatu daiteke.

12.1. Taula. Identifikatutako arriskuen zehaztasunak.

12.2 Arriskuen Analisia

Arriskuen analisia egiteko arriskuen inpaktu eta probabilitatea bidertzen dira, eta magnitudea kalkulatzen da. Horrela, arrisku garrantzitsuenak zein diren argiago izango dugu uneoro. 12.1. Irudian arriskuen inpaktu eta probabilitate analisia ikus daiteke. Arrisku garrantzitsuenak eskuinean eta goian agertzen dira.



12.1. Irudia. Arriskuen inpaktu eta probabilitate analisia.

13 Proiektuaren Antolamendua eta Kudeaketa

Kapitulu honek proposatutako sistemaren proiektuaren antolaketa eta kudeaketa egiteko jarraitu beharreko plana deskribatzea du helburu. Honetarako, *Project Management Institutek* gomendatutako PMBOK gida jarraitu da.

13.1 Proiektuaren Antolamendua

Proiektuaren antolaketari dagokionez, garrantzitsua da hasieratik hainbat aspektu argi edukitzea: taldekieen rolak, informazio-sistema eta komunikazio-kanalak.

13.1.1 Taldekideak eta rolak

Proiektu hau Gradu Amaierako Lana denez, ni naiz proiektuaren kide bakarra, Julen Etxaniz. Beraz, nik hartu beharko ditut OpenUp metodologian beharrezkoak diren rol guztiak: Proiektu Kudeatzailea, Analista, Arkitektoa, Probatzailea, Garatzailea, etab. Proiektuaren tutorea Juan Manuel Pikatza izango da.

13.1.2 Informazio-sistema

Sistemaren hedapenean komentatu den moduan, proiektuaren dokumentazioa eta implementazioa publikoki eskuragarri egongo dira GitHub bidez eta webgune bidez. Printzipioz, lana bukatu ondoren ere eskuragarri jarraituko dute, edozeinek konsultatu ahal izan ditzan. Gainera, amaitutakoan zuzendariak emandako makinara igoko da lana, proiektu honi jarraipena emateko eskuragarri egon dadin.

Estrategia honekin ziurtatzen da nik eta tutoreak uneoro atal bakoitzaren azkenengo bertsioa konsultatu dezakegula. Honek tutorearekin errebisioak egitea errazten du. Gainera, git bertsio kontrolari esker egindako aldaketa guztiak ikus daitezke, eta egindako lana galtzea saihesten da. Horrez gain, webguneak automatikoki eraikitzeak lana errazten du, ez baitaukat zerbitzari batera igotzen ibili beharrik aldaketak dauden bakoitzean.

13.1.3 Komunikazio-kanalak

Gehienetan etxetik egingo dut lana. Batzuetan EHUKO liburutegian ere arituko naiz lanean. Bilera gehienak BBC bidez online egingo ditugu eta beste batzuk presentzialki Informatika Fakultatean. Printzipioz, bilerak astero egingo ditugu aurretik adostutako ordu batean. Beharrezko bada eguna edo ordua aldatu dezakegu. Bilerez gain posta elektronikoa erabiliko dugu elkarrekin komunikatzeko.

13.2 Proiektuaren Kudeaketa

Aurrelik aipatu den bezala, proiektuaren kudeaketa PMBOK gida-liburuaren arabera egin da. Hori dela eta, liburu horrek aipatzen dituen kudeaketa-motak deskribatzen dira atal honetan.

13.2.1 Integrazioaren Kudeaketa

Atal honen helburua proiektuan zehar ematen diren prozesu desberdinak identifikatu, definitu, konbinatu, bateratu eta koordinatzeko eman beharreko pausuak kudeatzea da.

Proiektuaren hasiera tutorearekin egin beharreko lana adostutakoan izan zen. Lan eskaintza publikoan aukeratutakoa izan zen. Enpresarik ez dagoenez, ez dago bestelako akordiorik sinatu beharrik.

Proiektuaren plangintza eramateko OpenUP metodologiaren Proiektu Plana erabili da, helburuak eta irismena definitzeko. Gainera, CCII-N2016-02 estandarraren arabera, proiektuko kudeaketako beste esparru batzuetako kudeaketa planak ere egin dira, hurrengo ataletan daudenak.

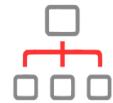
Proiekta zuzentzeko eta kudeatzeko ardura bi aktore garrantzitsuenena izango da: Julen Etxaniz egilea eta Juan Manuel Pikatza tutorea. Proiekta epe motzetan zuzentzeko eta kudeatzeko OpenUP metodologia erabili da, hilabete bateko iterazioekin. Iterazioak definitzeko Iterazio Planak erabili dira.

13.2.2 Irismenaren Kudeaketa

Atal honen zereginak da proiektuaren irismena kudeatzea, hots, proiektuaren espezifikazioak eta denbora zeintzuk diren kontuan hartuta, proiekta zenbateraino garatu daitekeen estimatzea eta kudeatzea.

Proiektuaren Ikuspegia bete den lehenengo dokumentua izan da. Izen ere, dokumentu horretan konpondu nahi dugun arazoa eta proposatutako produktua deskribatzen dira. Gainera, proiektuko interesatuen erantzukizunak ere zehazten dira.

Proiektuaren betekizunak definitzeko OpenUP metodologiako Ikuspegia eta Betebeharren Espezifikazioa dokumentua erabili dira. Ikuspegiak betekizun funtzional eta ez-funtzionalak zehaztu dira. Betebeharren



Especifikazioan beste betekizunak definitu dira: kalitatea, interfazeak, konplimentua eta. Erabilpen Kasuen dokumentuek eta ereduek ere betekizunak ulertzen laguntzen dute.

Proiektuaren irismena zehazteko, LDE diagrama definitu da lan-pakete eta ataza nagusiekin. Proiektua lan-paketetan zatitzeko ondorengo atazak egin dira:

1. Emangarriak eta erlazionatutako lana identifikatu eta aztertu.
2. LDE diagrama egituratu eta antolatu.
3. LDE diagramako goi mailako elementuak, zehatzagoak diren maila txikiagoko osagarrietan zatitu.
4. Identifikazio kodea garatu eta LDE diagramako elementuei kode hori esleitu.
5. Emangarrien deskonposaketa maila egokia dela ziurtatu.

13.2.3 Epeen Kudeaketa

Dokumentu honen xeedea da garrantzitsuak diren mugarriak identifikatzea, eta proiektuan zehar egon daitezkeen arazoak direla eta epeak aldatu behar badira, gertakizun hori kudeatzeko plana zehaztea.

Atazak zehaztu eta sekuentziatzeko OpenUPeko Lan Atazen Zerrenda erabili da, atazen kudeaketa egiteko aproposak diren datuak zehazten dituen taula. Adibidez, ataza bakoitzaren lehentasuna, tamaina, iterazioak eta denbora estimazioa zehazten da.

Atazen plangintza egiteko, Gantt diagrama erabili da. Atazen plangintza zehatza egitea oso zaila denez, diagrama honek iterazioko zehaztasuna izango du, hau da, hilabete bakotzean egingo diren atazak zehaztuko ditu.

Proiektuaren mugarriei dagokionez, iterazio hasiera eta amaierekin eta proiektuaren entregarekin zerikusi duten mugarriak zehaztu dira. Gerta daiteke proiektua epe horietan bukatzen denborarik ez izatea eta epea atzeratu behar izatea. Horrek proiektua uztalean aurkeztu ordez irailean aurkeztu beharra eragingo luke. Kontuan hartu behar da behin proiektuaren defentsa eskatzen denean derrigorrez aurkeztu behar dela, eta beraz ekainaren 11rako argi eduki behar dela noiz aurkeztu behar den.

13.2.4 Kostuen Kudeaketa

Atal honetan definitzen dira proiektua aurrera eramateko behar diren kostu ekonomikoak eta horiek nola kontrolatu.

Proiektuan egon daitezkeen kostu ekonomikoak aurreikusiak daudela onartzen da. Hori dela eta, pentsatzen da kostuak ez dutela plangintza eta kontrol berezirik beharko. Ahal den heinean, erabili beharreko softwarea librea izan behar da.

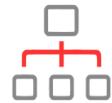
Beraz, estimatzen da tresna aldetik proiektuak ez duela kosturik izango. Hori bai, proiektuko kideen lan orduak kontuan hartuz, proiektuaren kostua zenbatekoa izango litzatekeen estimatzen da. Kostu hauek guztiek Aurrekontuan definitzen dira, doakoak diren tresnak ere zerrendatzu.

13.2.5 Kalitate Kudeaketa

Atal honetan proiektuan zehar helburu diren kalitate betekizunak ahalik eta hoheren gauzatzeko hurrengo jarraitu beharreko prozesua deskribatzen da.

Proiektuaren tamaina eta bereziki, egile bakar batek egindako proiektua dela kontuan hartuta, kalitatea zaintzeko hiru aspektu kontrolatu behar dira:

- **Implementazioaren kalitatea:** kode-mailako implementazioa egokia dela eta software arrunten patroiak jarraitzen direla bermatzeko.
- **Funtzionalitate eta ezaugarrien kalitatea:** produktuak bete behar dituen ezaugarri eta espezifikazioen kontrola.



- **Memoria, dokumentazioa eta aurkezpenaren kalitatea:** proiektuak atxikituta izan behar duen dokumentazio eta memoriaren kalitatea. Adostutako estandarrak betetzen diren (CCII eta OpenUP) ziurtatu behar da, bai eta artefaktu guztiak atal edo sekcio guztiekin beteta daudela. Gainera, webgunearren kalitate kontrola egin behar da.

Proietkuko rolen eta kalitate kudeaketako arduren banaketa 13.1. Taulan ikus daiteke. Proietkuaren egileak etengabeen hiru aspektuak kontrolatu beharko ditu proietkuaren bukaerara arte. Proietkuaren tutoreari dagokionez, memoria, dokumentazioa eta aurkezpenaren kalitate-kontrola egiten lagunduko du. Horretarako, astean behin egiten diren bileretan jasotako feedbacka erabiliko da.

Rolak	Ardurak
Egilea: Julen Etxaniz	Implementazioaren kalitatea Funtzionalitate eta ezaugarrien kalitatea Memoria, dokumentazioa eta aurkezpenaren kalitatea
Tutorea: Juan Manuel Pikatza	Memoria, dokumentazioa eta aurkezpenaren kalitatea

13.1. Taula. Kalitate kudeaketako rolak eta ardurak.

13.2.6 Giza Baliabideen Kudeaketa

Atal honen helburua da proietkuan parte hartuko duten aktore guztiak kudeaketa deskribatzea da.

Proietku hau Gradu Amaierako Lana denez, ni naiz proietkuaren kide bakarra, Julen Etxaniz. Beraz, nik harti beharko ditut OpenUp metodologian beharrezkoak diren rol guztiak: Proietku Kudeatzalea, Analista, Arkitektoa, Probatzailea, Garatzalea, etab. Proietkuaren tutorea Juan Manuel Pikatza izango da.

Beraz, giza baliabideen kudeaketa nahiko simplea izango da. Egileak rol bakoitzarekin pasatako denbora kudeatu beharko da. Tutorearekin komunikazioa kudeatuko da, postaz eta bileretan.

13.2.7 Komunikazioen Kudeaketa

Komunikazioen kudeaketari begira, atal honetan definitzen dira zein pausu jarraitzen diren kideen artean komunikatzeko.

Ahalik eta komunikazio hoherena edukitzeko lehenengo interesatuak identifikatu dira. Ondoren, komunikazioak planifikatu dira. Komunikazorako bi kanal nagusi erabiliko dira, korreoa eta bilerak.

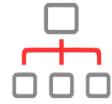
Korreo bidezko komunikazioa bilera baino lehenago komentatu beharreko aspektuetarako erabiliko da. Komunikazioa EHuko posta edo Gmail erabiliz gauzatuko da. Beraz, nahiko irregularra izango da, baliteke aste batzuetan hainbat aldiz idaztea eta beste batzuetan ezer ez.

Bilerei dagokionez, astean behin egingo dira adostutako orduan BBC bidez. Hala ere, tutorearekin adostuta eguna eta ordua alda daitezke beharrezkoa bada. Bilerak online egitea erabaki dugu Covid19-ren egoeragatik eta ordutegia aukeratzea errazagoa delako. Gainera, pantaila partekatzeko aukera izateak bileretan asko lagundi digu.

13.2.8 Arriskuen Kudeaketa

Atal honen zeregina proietkuan zehar gerta daitezkeen arriskuen identifikatzea, arrisku horiek ebazteko prozesua definitzea eta kontrolatzea da.

Lehenik eta behin, proietkuak planteatutakoan arrisku-zerrenda bat egin da OpenUP metodologia erabiliz. Arrisku bakoitzarentzat analisi kualitatibo eta kuantitatibo egin da, inpaktua eta probabilitatea estimatuz. Horrez gain, mitigazio estrategiak definituko dira arriskuei aurre egiteko. Gainera, gerta daiteke proietkuak garatu bitartean arrisku gehiago agertzea. Kasu horretan, arriskua zerrendatu eta analisia egingo da.



13.2.9 Erosketen Kudeaketa

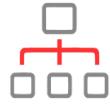
Atal honen helburua proietkuko erosketak kudeatzeko plana zehaztea da.

Kostuen kudeaketan azaldu den moduan, ez da aurreikusten erosketarik egin beharra. Izan ere, erabiliko diren tresna guztiak libreak izatea da helburua. Gainera, ordainpekoak diren software batzuk eskuragarri daude tutoreak eskainitako makina birtualean beharko balira. Hala ere, zerbait erozi beharko balitz egileak kudeatuko luke tutorearen laguntzarekin.

13.2.10 Interesatuen Kudeaketa

Atal honen helburua proietkuko interesatuak identifikatzeko, kudeatzeko eta kontrolatzeko pausuak zehaztea da.

Interesatuak eta horien erantzukizunak proietkuaren hasieran finkatuko dira Ikuspegia dokumentuan. Interesatu garrantzitsuenak kontuan hartzen direla ziurtatu beharko da proietkuaren garapenean. Esan bezala, interesatuen arteko komunikaziorako korreoa eta bilerak erabiliko dira.



14 Denbora Planifikazioa

Atal honen helburua da proiektuaren denbora aurre-planifikazioa azaltzea eta mugarri garrantzitsuak atazentzen denborak, iterazioak eta desbiderapenak zehaztea da. Denboraren jarraipena egiteko Toggle Track tresna erabili da. Tresna honekin proiektuko denbora guztiak kontrolatu dira, temporizadore bat erabiliz.

Azpiatzazak iterazioko edo beharrezko momentuan sortu dira. Izan ere, oso zaila da aurreikustea ezjakintasun handia duen proiektu batean zein atazetan egingo den lan hemendik hilabete batzuetara. Ordu kantitatea kontuan hartuko da, baina oso zaila da ataza bakoitzaren denbora eta epeak estimatzea. Horregatik, iterazioen planak eguneratzen joatea garrantzitsua da.

14.1 Mugarriak

Proiektua planifikatzeko garrantzitsua da mugarri garrantzitsuak biltzea eta momentu guztietan argi edukitzea. 14.1. Taulan eta 14.1. Irudian proiektuaren mugarri garrantzitsuenak eta datak azaltzen dira. Gehienek iterazio hasiera eta amaierekin edo entregekin dute zerikusia.

Mugarria	Data
Proiektuaren hasiera	2021/01/14
1. Iterazioaren amaiera	2021/02/14
2. Iterazioaren amaiera	2021/03/15
3. Iterazioaren amaiera	2021/04/16
4. Iterazioaren amaiera	2021/05/17
5. Iterazioaren amaiera	2021/06/20
6. Iterazioaren amaiera	2021/07/23
Lana GAURen matrikulatu	2021/07/23
Lanaren posterra bidali	2021/09/05
Lana ADDIra igo	2021/09/05
Proiektuaren amaiera	2021/09/05
Lanaren defentsa	2021/09/13

14.1. Taula. Proiektuko mugarri garrantzitsuak.



14.1. Irudia. Mugarri garrantzitsuen diagrama.

14.2 Lan-atazak

Lana modu egokian antolatu eta kontrolatzeko, ezinbestekoa da lana ataza eta azpiatazeten banatzea. Hala ere, ez da gomendagarria azpiataza gehiegi definitzea, kontrolatzeko eta ulertzeko zailagoa baita. Ataza hauek ez dira estatikoak, proiekta aurrera joan ahala berriak ager daitezke.

Hurrengo taulak adierazten du proiektauan zehar sortutako lan-ataza guztien zehaztasunak adierazten ditu: lehentasuna, tamaina, iterazioak, esfortzu estimazioa, lan orduak eta erreferentzia materiala.

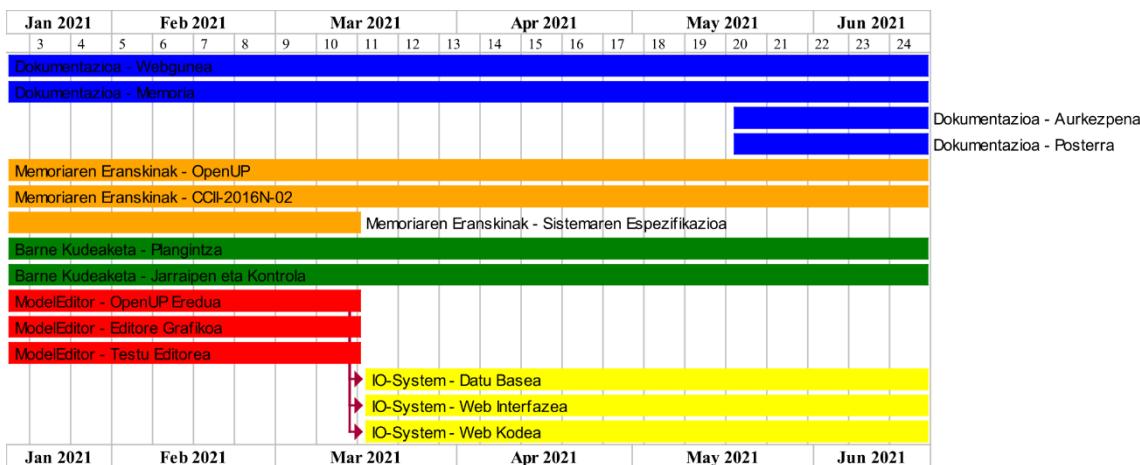
Izena	Lehentasuna	Tamaina	Iterazioak	Esfortzu estimazioa	Lan orduak	Erreferentzia materiala
Webgunea	1	3	I1,I2,I3,I4,I5			CCII-2016N-02
Posterra	2	2	I5			UPV/EHU
Aurkezpena	3	3	I5			UPV/EHU
Memoria	3	5	I1,I2,I3,I4,I5			UPV/EHU, CCII-2016N-02
Memoriaren Eranskinak - OpenUp	1	3	I1,I2,I3,I4,I5			OpenUP
Memoriaren Eranskinak - CCII-2016N-02	1	3	I1,I2,I3,I4,I5			CCII-2016N-02
Sistemaren Espezifikazioa	1	2	I1,I2			OpenUP
Aurrekontua	2	1	I1,I2			CCII-2016N-02
Plangintza	3	2	I1			OpenUP
Aurreko Proiektuak Aztertu	2	1	I1			ProWF, BETRADOK
Ingurunea Prestatu	2	1	I1,I2,I3,I4			Eclipse, Drupal
Barne Kudeaketa	1	3	I1,I2,I3,I4,I5			OpenUp
ModelEditor - OpenUp Eredua	1	5	I1, I2			OpenUP, EMF
ModelEditor - Editore Grafikoa	1	5	I1, I2			EMF
ModelEditor - Testu Editorea	1	5	I1, I2			Xtext
IOSystem - Datu Basea	1	3	I3			Drupal
IOSystem - Web Interfazea	1	5	I3,I4			Drupal

IOSystem - Web Kodea	1	5	13,14		Drupal
----------------------	---	---	-------	--	--------

14.2. Taula. Lan-atazen zehaztasunak

14.3 Gantt Diagrama

Gantt diagramaren bitartez atazen garapen denborak zehaztu daitezke. Aurreko ataletan definitutako paketeak eta atazak kontuan hartuz, 14.2. Irudian agertzen den Gantt diagraman iterazio bakoitzeko lanaren estimazioa egin da. Atazak LDE diagraman zehazten direnak dira, eta kolore berdinak adierazi dira.

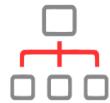


14.2. Irudia. Gantt diagrama.

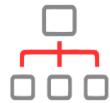
14.4 Iterazioak

Atal honetan, OpenUP metodologiaren bitartez jaso diren atazak iterazioetan zehar nolako banaketa izango duten deskribatuko da. Lan guzta bost iteraziotan banatu da, bakoitzak hilabete ingruuko iraupena izanik. 14.3. Taulan iterazio bakoitzaren fasea, helburuak eta datak agertzen dira. Lehenengo bi iterazioak Hasiera fasekoak dira, hurrengo biak Elaborazio fasekoak eta azkena Eraikuntza fasekoak. Helburuak lan-atazen antzekoak dira, baina informazio gehiagorekin.

Fasea	Iterazioa	Helburuak	Hasiera	Amaiera
Hasiera	I1	Iterazio Plana. OpenUp Hasiera faseko artefaktuak. Aurreko projektuen informazioa ulertu. OpenUp txantiloia eskuratu. Memoriaren txantiloia sortu (Fakultateko Arautegiak hemen erabiliko diren metodo eta arauak gomendatzenten dutenekin ez dator beti bat. Konponbide batzuk eratuko dira. CCII arauaren informazioa jaso. CCII araua betetzen duen webgunea eraiki.	2021/01/14	2021/02/14



		OpenUp metodologiaren informazioa bildu (metaeredua, eredu...). Lan egiteko tresnak eskuratu (Eclipse, EPF Composer, EHSIS, Drupal, etab.). ProMeta proiektuak sortuko duen produktuak jarraituko duen garapen prozesuaren OpenUp eredu definitu. Metaereduan oinarritutako eredu editore grafiko eta testu editorearen prototipoa.		
Hasiera	I2	Iterazio Plana. OpenUp Hasiera faseko artefaktuak. ProMeta proiektuak sortuko duen produktuak jarraituko duen garapen prozesuaren OpenUp eredu definitu. Metaereduan oinarritutako eredu editore grafiko eta testu editorearen prototipoa. Editore grafiko eta testu editorearen arteko sinkronizazioa. Eedu bera bi modutan editatzeko aukera. Drupal webguneak erabiliko duen databasea sortu eredutik.	2021/02/15	2021/03/15
Elaborazioa	I3	Iterazio Plana. Drupal webguneak erabiliko duen databasea sortu eredutik. OpenUp Elaborazio faseko artefaktuak. Web interfazea eraiki (Drupal). Web kodea garatu (Drupal). Webgunea. Memoria.	2021/03/16	2021/04/16
Elaborazioa	I4	Iterazio Plana. OpenUp Elaborazio faseko artefaktuak. Web interfazea eraiki (Drupal). Web kodea garatu (Drupal). Webgunea. Memoria.	2021/04/17	2021/05/17



Eraikuntza	15	Iterazio Plana. OpenUp Eraikuntza faseko artefaktuak. Web interfazea eraiki (Drupal). Web kodea garatu (Drupal). Webgunea zerbitzari batean jarri. Webgunea. Memoria. Posterra. Aurkezpena.	2021/05/18	2021/06/20
------------	----	---	------------	------------

14.3. Taula. Proiektuko iterazioen fasea, helburuak eta datak.

14.5 Neurtutako denborak

Toggle Track aplikazioa erabiliz neurtu dira proiektuko denborak. Ataza bakoitzaren denbora neurtu denez, ataza bakoitzeko denbora jakiteko ez daukagu kalkulurik egin beharrik, aplikazioak zuzenean esaten digu. Gainera, nahi dugun denborak bistaratu ditzakegu, adibidez hilabete bakoitzeko denbora edo aste bateko egun bakoitzeko.

14.5.1 Denbora hilabeteka

Guztira hilabete bakoitzean pasatako denbora ikus dezakegu hurrengo irudian.

01-10-2020 – 30-06-2021

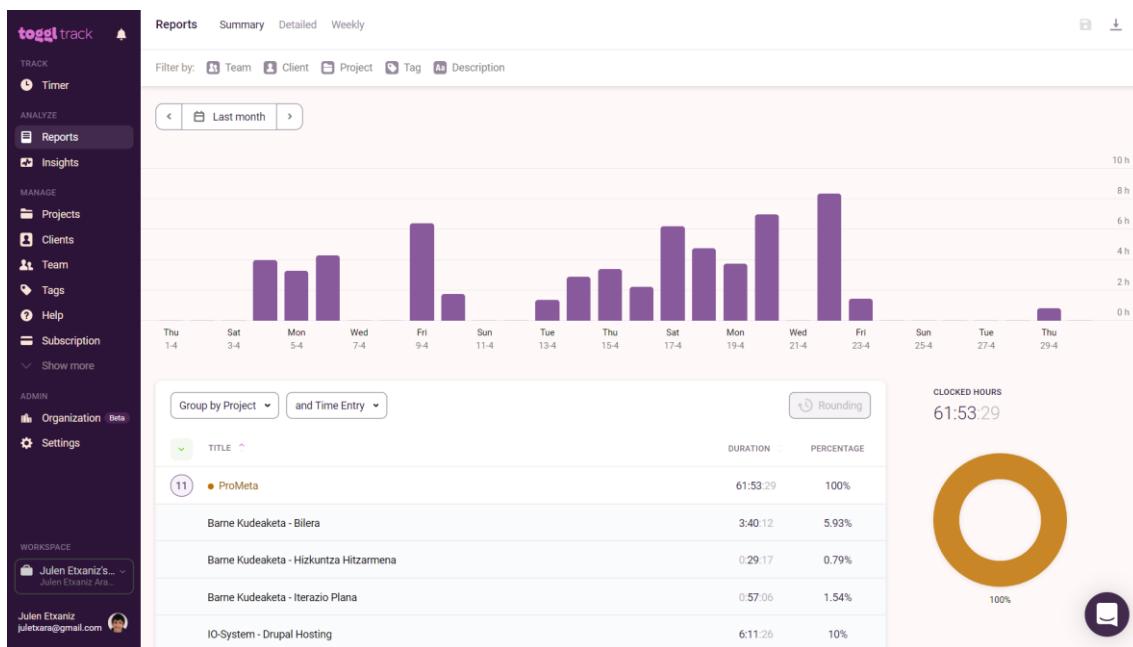
TOTAL HOURS: 293:10:46



14.3. Irudia. Proiektuaren denbora hilabeteka.

14.5.2 Hilabete bateko denbora

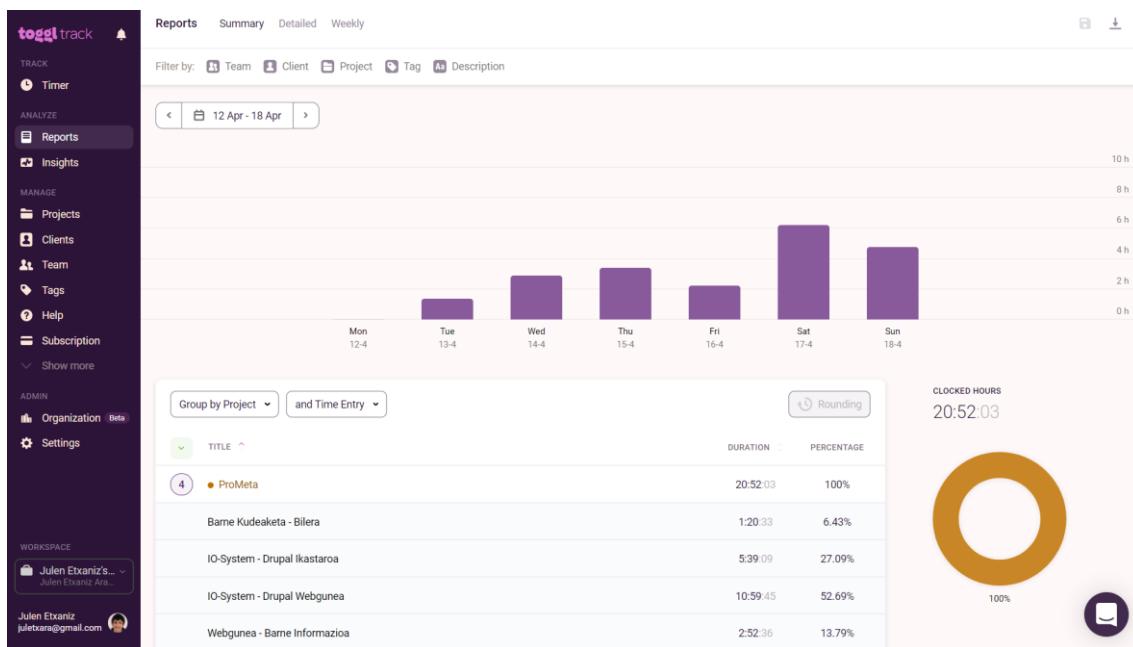
Hilabete bateko egun bakoitzeko denbora ikus dezakegu grafika batean. Gainera, azpian ataza bakoitzari eskainitako denbora eta denbora totala ikus ditzakegu.



14.4. Irudia. Hilabete bateko denbora.

14.5.3 Aste bateko denbora

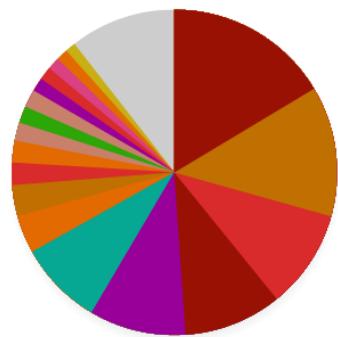
Aste bateko egun bakoitzeko denbora ikus dezakegu grafika batean. Gainera, azpian ataza bakoitzari eskainitako denbora eta denbora totala ikus ditzakegu.



14.5. Irudia. Aste bateko denbora.

14.5.4 Denbora atazaka

Ataza garrantzitsuenei guztira eskainitako denbora ikus dezakegu hurrengo irudian. Denbora gutxi eskaini zaien atazak grisez multzokatuta agertzen dira. Ataza horien denborak kontsulta daitezke arazorik gabe.



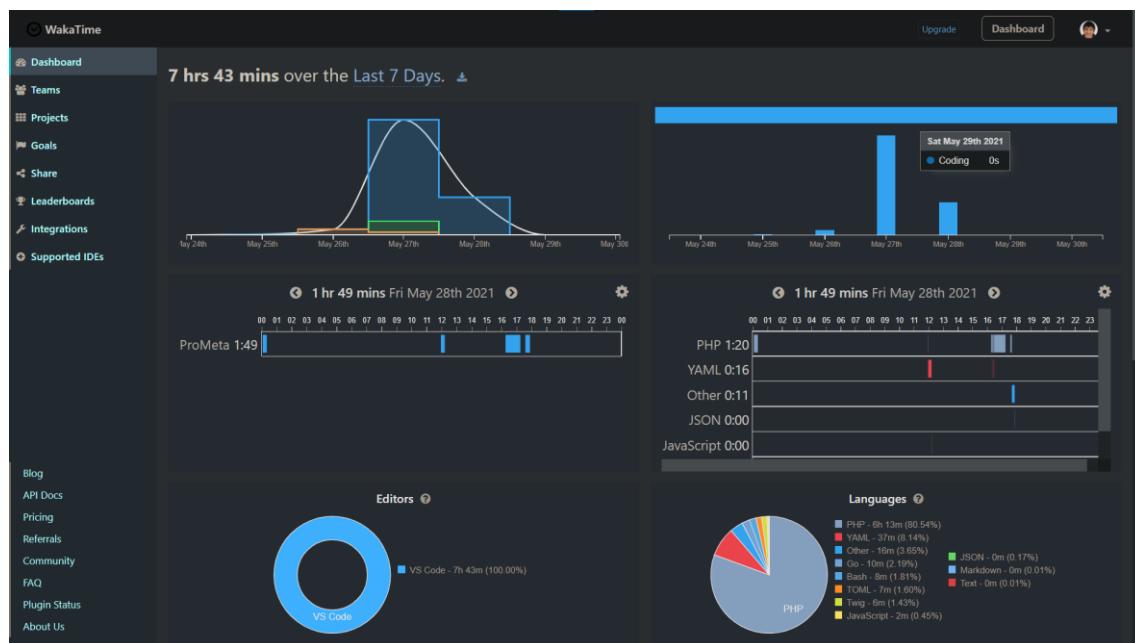
TIME ENTRY

● ModelEditor - OpenUp Eredua	48:04:22
● IO-System - Datu Basea	38:00:39
● ModelEditor - Testu Editorea	28:47:26
● IO-System - Drupal Webgunea	28:33:29
● Barne Kudeaketa - Bilera	28:05:06
● ModelEditor - Editore Grafikoa	24:43:14
● Memoria	10:50:18
● Barne Kudeaketa - Aukeren Azterketa	9:00:00
● IO-System - Drupal Ikastaroa	6:39:09
● IO-System - Drupal Hosting	6:11:26
● Sistemaren Espezifikazioa - Ikuspegia	5:21:13
● IO-System - Aukeren Azterketa	5:01:00
● Webgunea - ProMeta	5:00:00
● Barne Kudeaketa - Proiektu Plana	4:00:55
● IO-System - ProWF	3:59:01
● Webgunea - Egitura Hobetu	3:37:19

14.6. Irudia. Proiektuaren denbora atazaka.

14.5.5 WakaTime

WakaTime tresnak kodetzen denborari buruzko estatistika zehatzagoak eskaintzen ditu. Adibidez, programazio-lengoai, editore eta proiektu bakoitzaren denbora ikus daiteke.



14.6 Desbiderapenak

15 Aurrekontuaren Laburpena

Aurrekontua sortzeko ALI (*Asociación de Titulados Universitarios Oficiales en Informática*) elkarreka banatutako irizpide batzuk jarraitu dira. Batetik, giza baliabideen barne-kostuak eta kanpo-kostuak ateratzea bere ordu kopuruekin batera. Gure kasuen ez dago kanpo-kosturik. Bestetik, proiectua garatzeko behar izan diren erreminten kostua kalkulatzea.

Giza baliabideen kostua Ekonomia eta Ogasun Ministerioaren 26/2010 Esparru Akordioan oinarritura dago. Bezeroari begira software proiectu baten kide bakoitzarentzako erabiltzen dugun baremoa 15.1. Taulan ikus daiteke.

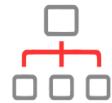
Kidea	Kostua (€/ordu)
Proiectuko zuzendaria	100
Arkitektoa	70
Analista	70
Garatzailea	50
Probatzailea	50

15.1. Taula. Software taldeko kideen kostu baremoa.

Erabilitako erreminta guztiak doakoak izan dira, beraz, arkitektura propioa eraikitzearen erabakia egokia izan da. Bizagiren arkitektura erabiliz bere lizenziak eta urteroko mantenuak proiectuaren kostua handituko luke. Beste alde batetik, ez dira aurkitu erabilitako erreminten premium lizenziarik, baina egotekotan aurrekontuan sartu daitezke, sistemaren kalitatea handitzeko asmoz. 15.2. Taulan proposatutako sistema bideragarria izateko aurrekontua ikus daiteke:

PARTIDA		PARAMETROAK		TOTALA BEZ GABE	TOTALA BEZ BARNE
	Giza Baliabideak	Orduak	Kostua		
1	Zuzendaria	0	100	0	0
2	Arkitektoa	0	70	0	0
3	Analista	0	70	0	0
4	Garatzailea	0	50	0	0
5	Probatzailea	0	50	0	0
Totala				0	0
	Erremintak	Lizentzia	Mantenua		
1	XAMPP	0	0	0	0
2	Drupal	0	0	0	0
3	VSCode	0	0	0	0
4	PlantUML	0	0	0	0
5	Eclipse	0	0	0	0
6	Git	0	0	0	0
7	GitHub	0	0	0	0
8	Java	0	0	0	0
9	Microsoft Office	0	0	0	0
10	Pantheon	0	0	0	0
Totala				0	0
TOTALA				0	0

15.2. Taula. Aurrekontua giza baliabideen eta erreminten kostuekin.



16 Dokumentuen Lehentasun Ordena

Dokumentazio luze honetan inkoherentziak egotea posible izan daiteke. Proiektuaren garapena luzea izan da, dokumentu asko idatzi dira eta gerta daiteke dokumenturen batean agertzen den baieztapen bat kontrajartzea beste dokumentu batean agertzen den baieztapen batekin edo dokumentu batean agertutako datu bat beste batean ezberdina izatea.

Hori dela eta, memoria izango da kontuan hartu beharreko informazioa inkoherentzien kasuan. Memoria dokumentu askoren bilketa da azken finean, baita proiektuaren azkenekoz idatzitako dokumentua. Horregatik, irakurleak memoria konsultatu beharko du zalantzarik izanez gero.

Hala ere, beti prest egongo naiz edozein zalantza edo arazo argitzeko. Nirekin kontaktuan jartzeko posta elektronikoa erabil daiteke: juletxara@gmail.com. Nahiago bada, nire webgunean kontakturako aukera gehiago daude: <https://julenetxaniz.eus/#contact>.

