

## 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

Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema.

## Aurkibide Orokorra

Laburpena.....	2
Aurkibide Orokorra.....	3
Irudien Aurkibidea .....	11
Taulen Aurkibidea.....	12
1 Sarrera .....	13
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 Gogorak.....	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 .....	28
4.1 Deskribapena .....	28
4.2 Abantailak eta Desabantailak .....	28
4.2.1 Abantailak.....	28
4.2.2 Desabantailak .....	28
4.3 Proposatutako Hobekuntzak .....	29
4.4 Prestakuntza .....	30
5 Arauak eta Erreferentziak.....	32
5.1 Aplikatutako Legedia eta Araudia .....	32
5.2 Bibliografia .....	32
5.3 Metodoak.....	33
5.3.1 OpenUP.....	33
5.3.2 ABRD.....	33
5.3.3 MDE .....	33
5.4 Tresnak.....	33
5.4.1 Git .....	34

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

1.4	CCII-N2016-02 .....	40
1.5	CMMI .....	40
1.6	CMS .....	40
1.7	DOT .....	40
1.8	Drupal .....	40
1.9	IBM.....	40
1.10	OpenUP .....	40
1.11	PlantUML .....	40
1.12	ProMeta .....	41
1.13	ProWF .....	41
1.14	RUP.....	41
1.15	SPICE .....	41
1.16	UML.....	41
1.17	UNE .....	41
1.18	URPS.....	41
1.19	Workflow .....	41
1.20	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	Artefaktuak .....	45
9	Hipotesiak eta Murritzapenak .....	47
9.1	Hipotesiak .....	47
9.2	Murritzapenak .....	47
10	Aukeren Azterketa eta Egingarritasuna .....	48

10.1	Arkitektura .....	48
10.1.1	Bizagi .....	48
10.1.2	ProWF .....	49
10.1.3	ProMeta .....	50
10.2	CMS aukerak .....	50
10.2.1	Wordpress .....	51
10.2.2	Drupal .....	52
10.2.3	Joomla .....	52
10.3	Dokumentazioa Hosting Aukerak .....	52
10.3.1	GitHub Pages .....	52
10.3.2	GitLab Pages .....	53
10.3.3	Netlify .....	53
10.4	Drupal Hosting Aukerak .....	53
10.4.1	000webhost .....	53
10.4.2	Heroku .....	53
10.4.3	Acquia .....	53
10.4.4	Pantheon .....	53
10.4.5	Platform.sh .....	53
10.5	Datu-base kudeaketa sistema .....	53
10.5.1	MySQL .....	53
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 .....	54
10.7.2	Ordenagailu pertsonala .....	54
10.8	Bertsio kontrola .....	54
10.8.1	GitHub .....	54
10.8.2	GitLab .....	55
10.9	Metaereduentzako tresnak .....	55
10.9.1	Eclipseren tresnak .....	55
10.10	Metaeredua .....	55
10.10.1	SPEM .....	55
10.10.2	UMA .....	55
10.11	Metodologia aukeratu .....	55

10.11.1	OpenUp .....	55
10.11.2	RUP .....	55
10.11.3	ABRD.....	55
10.12	Metodologia definitu .....	55
10.12.1	EPF Composer .....	55
10.12.2	Rational Method Composer .....	56
10.12.3	Editore grafikoa .....	56
10.12.4	Testu editorea .....	56
10.13	Prozesua bistaratu .....	56
10.13.1	DOT.....	56
10.13.2	XPDL .....	56
10.13.3	Webgunea .....	56
10.14	Proiektuaren kudeaketa .....	56
10.14.1	ClickUp.....	56
10.14.2	Quire.....	56
10.14.3	Trello .....	56
10.15	Denboraren kontrola .....	56
10.15.1	Toggl Track .....	56
10.15.2	Clockify .....	56
10.15.3	WakaTime .....	56
10.16	Gantt diagrama.....	57
10.16.1	Teamgantt .....	57
10.16.2	Elegantt .....	57
10.16.3	Ganttproject.....	57
10.17	Diagramak.....	57
10.17.1	PlantUML.....	57
10.17.2	Draw.io .....	57
11	Proposatutako Sistemaren Deskribapena .....	58
11.1	Azpisistemak .....	58
11.2	ModelEditor .....	58
11.2.1	Arkitektura .....	58
11.2.2	Analisia .....	59
11.2.3	Pakete Dependentsia Diagrama .....	59
11.2.4	Pakete Diagrama .....	59
11.2.5	Klase Diagrama .....	60
11.2.6	Diseinua .....	61
11.2.7	Garapena .....	61

11.2.8	Proba .....	61
11.3	IO-System .....	61
11.3.1	Arkitektura .....	61
11.3.2	Analisia .....	62
11.3.3	Diseinua .....	62
11.3.4	Garapena .....	62
11.3.5	Proba .....	62
11.4	Hedapena .....	62
11.5	Etorkizunerako Hobekuntzak .....	63
12	Arriskuen Analisia .....	63
12.1	Arriskuak .....	63
12.2	Arriskuen Analisia .....	65
12.3	Arazoak .....	66
13	Proiektuaren Antolamendua eta Kudeaketa .....	67
13.1	Proiektuaren Antolamendua .....	67
13.1.1	Taldeak eta rolak .....	67
13.1.2	Informazio-sistema .....	67
13.1.3	Komunikazio-kanalak .....	67
13.2	Proiektuaren Kudeaketa .....	67
13.2.1	Integrazioaren Kudeaketa .....	67
13.2.2	Irismenaren Kudeaketa .....	68
13.2.3	Epeen Kudeaketa .....	68
13.2.4	Kostuen Kudeaketa .....	68
13.2.5	Kalitate Kudeaketa .....	69
13.2.6	Giza Baliabideen Kudeaketa .....	69
13.2.7	Komunikazioen Kudeaketa .....	69
13.2.8	Arriskuen Kudeaketa .....	70
13.2.9	Erosketen Kudeaketa .....	70
13.2.10	Interesatuen Kudeaketa .....	70
14	Denbora Planifikazioa .....	71
14.1	Mugarriak .....	71
14.2	Lan-atazak .....	71
14.3	LDE diagrama .....	72
14.4	Gantt diagrama .....	73
14.5	Iterazioak .....	74
14.6	Neurtutako denborak .....	75
14.6.1	Denbora hilabeteka .....	75



14.6.2	Hilabete bateko denbora .....	76
14.6.3	Aste bateko denbora .....	76
14.6.4	Denbora atazaka .....	77
14.6.5	WakaTime .....	77
14.7	Desbiderapenak .....	78
15	Aurrekontuaren Laburpena .....	79
16	Dokumentuen Lehentasun Ordena .....	80
17	Memoriaren Eranskinak.....	81
17.1	Sarrerako Dokumentazioa .....	81
17.2	Analisia eta Diseinua .....	81
17.2.1	Arkitektura Kuadernoa .....	81
17.2.2	Analisiaren Eredua .....	81
17.2.3	Diseinuaren Eredua .....	81
17.3	Tamaina eta Esfortzu Estimazioa .....	81
17.4	Kudeaketa Plana .....	81
17.4.1	Integrazioaren Kudeaketa .....	81
17.4.2	Irismenaren Kudeaketa .....	81
17.4.3	Epeen Kudeaketa.....	81
17.4.4	Produktuaren Kostuen Kudeaketa .....	81
17.4.5	Kalitate Kudeaketa .....	81
17.4.6	Giza Baliabideen Kudeaketa .....	81
17.4.7	Komunikazioen Kudeaketa .....	81
17.4.8	Arriskuen Kudeaketa .....	81
17.4.9	Erosketen Kudeaketa .....	81
17.4.10	Interesatuen Kudeaketa .....	81
17.5	Segurtasun Plana .....	81
17.6	Beste Eranskinak .....	81
17.6.1	Hedapena .....	81
17.6.2	Garapena .....	81
17.6.3	Ingurunea .....	81
17.6.4	Proba .....	81
18	Sistemaren Espezifikazioa.....	81
18.1	Glosategia .....	81
18.2	Ikuspegia .....	81
18.3	Betebeharren Espezifikazioa.....	81
18.4	Erabilpen Kasuak.....	81
18.5	Erabilpen Kasuen Eredua .....	81

19	Aurrekontua.....	81
20	Ikerlanak .....	81

## Irudien Aurkibidea

1.1. Irudia. Proiektuaren webgunean memoria bistaratu. ....	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 webgunearen 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 tradizionalaren eskema. ....	23
3.8. Irudia. Erregeletan oinarritutako sistemen eskema: klaseak, objektuak, erregelak eta egitateak. ....	24
3.9. Irudia. MDERen 3 abstrakzio maila: eredua, metaeredua eta meta-metaeredua. ....	25
3.10. Irudia. ProWF sistemak behar dituen osagaiak eta haien arteko erlazioak. ....	26
5.1. Irudia. OpenUP bizi-zikloa EPF Composer tresnan. ....	35
5.2. Irudia. PlantUMLren lengoain idatzitako sekuentzia-diagrama. ....	36
5.3. Irudia. Draw.io diagrama adibidea. ....	36
5.4. Irudia. Proiektuan sortutako Drupal webgunea. ....	37
5.5. Irudia. XAMPP kontrol panela Apache eta MySQL hasieratzeko. ....	37
5.6. Irudia. ProMeta webguneko Pantheon kontrol panela. ....	38
8.1. Irudia. OpenUP metodologiako bizi-zikloaren faseak. ....	45
10.1. Irudia. Bizagi Modeler erabiliz sortutako prozesua. ....	49
10.2. Irudia. Bizagiren arkitektura. ....	50
10.3. Irudia. Workflowetan oinarritutako arkitektura. ....	50
12.1. Irudia. Arriskuen inpaktu eta probabilitate analisia. ....	65
14.1. Irudia. LDE diagrama. ....	73
14.2. Irudia. Gantt diagrama. ....	73
14.3. Irudia. Proiektuaren denbora hilabeteka. ....	76
14.4. Irudia. Hilabete bateko denbora. ....	76
14.5. Irudia. Aste bateko denbora. ....	77
14.6. Irudia. Proiektuaren denbora atazaka. ....	77

## 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 .....	46
10.1.Taula. CMS ezberdinen ezaugarrien konparaketa.....	51
12.1. Taula. Identifikatutako arriskuen zehaztasunak.....	65
12.2. Taula. Proiektuan zehar edukitako arazo nagusiak .....	66
13.1. Taula. Kalitate kudeaketako rolak eta ardurak.....	69
14.1. Taula. Proiektuko mugarri garrantzitsuak.....	71
14.2. Taula. Lan-atazen zehaztasunak.....	72
14.3. Taula. Proiektuko iterazioen fasea, helburuak eta datak.....	75
15.1. Taula. Aurrekontua giza baliabideen eta erreminten kostuekin.....	79

# 1 Sarrera

Dokumentu hau Julen Etxaniz Aragonesesen Informatika Ingeniaritzako Gradu Amaierako Lanaren memoria da. Lan hau Euskal Herriko Unibertsitateko (UPV-EHU) Donostiako Informatika Fakultatean landu da, Juan Manuel Pikatza izanik tutorea.

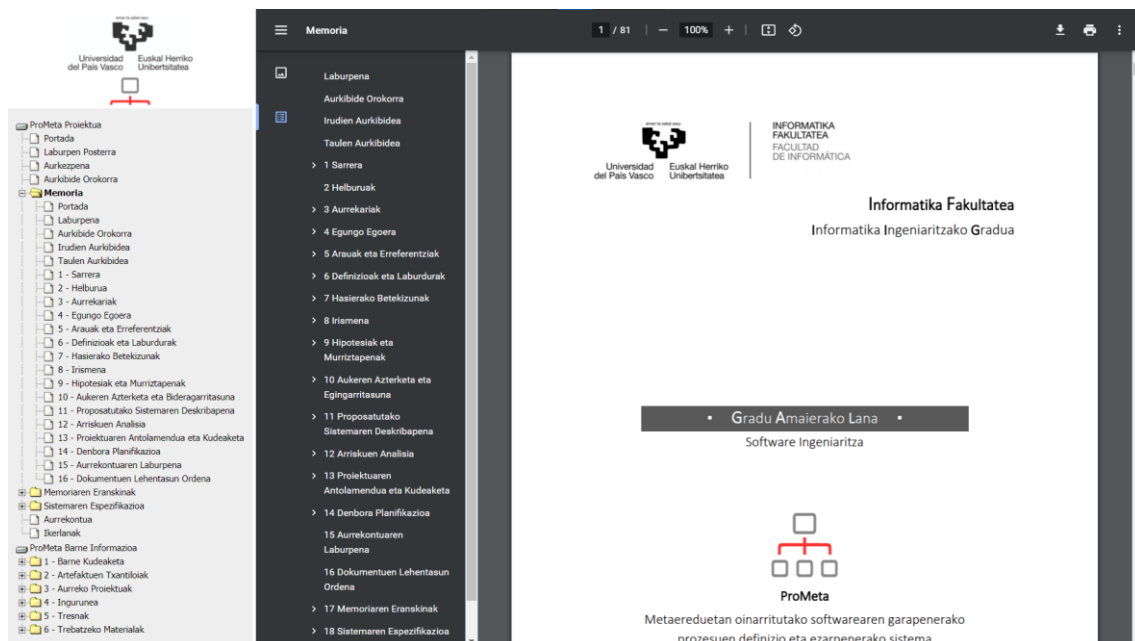
Dokumentu honetan ProMeta proiekturen inguruko aspektu guztiak azalduko dira. ProMeta izena prozesu edo profesional eta metaeredu hitzetatik dator. Portadan ikusten den proiektuaren logoak metaeredu bat irudikatzen du. Proiektuaren izenburua **Metaereduetan oinarritutako softwarearen garapenerako prozesuen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

Proiektua OpenUP metodologia erabilita gauzatu denez, bertako pausoak jarraitu dira. Gainera, dokumentazio formala 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 guztia 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 bistartzen da 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 bistartu.

## 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.

Bigarrengean, metodologiaren ereduaren 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, ezinbestekoa 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 Graduak 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.

<b>Arazoaren Deskribapena</b>	Proiektu informatikoen elaboraziorako ezagutza erabilgarria pilatzen da zenbait jardura modu automatikoan egiteko, adostutako gida bati 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.
<b>Interesatu Kaltetuak</b>	Proiektu informatikoak garatzen dituzten erakundeak.
<b>Arazoaren Eragina</b>	Estandarizazio maila baxua software proiektuen garapen prozesuan. Desadostasunak, atzerapenak eta akatsak eragiten ditu honek.
<b>Soluzioaren Abantailak</b>	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.

<b>Bezeroa</b>	Software proiektuak garatzen dituen enpresa.
<b>Beharra</b>	Softwarea garatzeko prozesuaren bizi-zikloa definitu eta exekutatzeko duen sistemaren beharra, prozesua mantendu eta hobetu ahal izateko.
<b>Produktua</b>	ProMeta: Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenarako sistema.
<b>Abantaila</b>	Softwarea garatzeko prozesua definitu, kudeatu, mantendu eta hobetzeko erraztasuna.
<b>Alternatiba</b>	ProWF: Software proiektuen elaboraziorako workflowetan oinarritutako sistemaren sorkuntza eta bizi-zikloa definitzeko metodologia baten ezarpena. Ez ditu OpenUp bizi-zikloaren fase guztiak, hasiera eta elaborazio fasearen zati bat bakarrik.
<b>Hobekuntzak</b>	ProMeta sistemak metaereduak erabiltzen dituen erraztasuna ematen du aldaketak egiteko. Etorkizunean sistemaren atal bat aldatzea erabakitzen bada, metaeredua edo ereduak aldatzea nahikoa da. OpenUp metodologiaren bizi-zikloa osatzea du helburu. IO-System sistemaren editorea hobetzea ere aurreikusten da. Gainera, sistema zerbitzari batean jarriko da, edozein erabiltzailearentzat 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 aukeratzeko dituen.	Proiektua sortu Partaideak aukeratu
Administratzailea	Sistema kudeatzeaz arduratzen da.	Erabiltzaileen kontuak kudeatu Sisteman aldaketak egin
Proiektuko Kudeatzailea	Software proiektuaren liderra da, interesdunen arteko harremanak koordinatzen ditu, eta proiektuko taldea helburuetan enfokaturik egotea du ardura.	Proiektuaren ardura eta rola esleitu Prozesuaren gida automatikoak jarraitu
Proiektuko Analista	Software proiektuaren eskakizunak eta betebeharrak harrapatzen ditu.	Proiektuaren eskakizunak eta beharrak eskuratu

Izena	Deskribapena	Erantzukizunak
		Prozesuaren gida automatikoak jarraitu
Proiektuko Prozesu Ingeniaria	Software proiektuaren eskakizunak eta betebeharrak aztertuz, prozesuak sortzen ditu.	Prozesua sortu/editatu, metaeredu eta ereduetan aldaketak eginez.  Prozesuaren gida automatikoak jarraitu
Proiektuaren beste taldekideak	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 proiektuan 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 eraikitzekeko prozedurak dira. Prozedurek, bezeroek eskatutako kontsentsu maila altuko nazioarteko estandarretan oinarritutakoak izan behar dute eta, herrialde bakoitzean, normalizazio agentzia ofizialaren berariazko lan-taldeak egokitutakoak. Prozeduren kalitate-eredu ezberdinak daude, garrantzitsuenak *CMMI*<sup>1</sup> eta *SPICE*<sup>2</sup> dira, helburu berdina dutenak. Kalitate-ereduek, softwarea garatzeko praktika onenak definitzen dituzte, softwarea garatzen duten erakundeek prozesuak hobetzen laguntzeko. Ziurtagiri-emaeleek, praktika on horiek hartu eta beraien emaitza neurgarriak egiaztatzen dituzte garatzaileekin 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**, hornitzaileek kalitate bermeak erakusteko arau europarrei erreferentzia egin eta, dagokien 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. artikulua** 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

---

<sup>1</sup> *CMMI*: Sigla(ingelesez), Capability Maturity Model Integration. Software-sistemak garatzeko, mantentzeko eta erabiltzeko, prozesuak hobetzeko eta ebaluatzekeo eredu da, *CMMi* Institutuak administratutakoa.

<sup>2</sup> *SPICE*: Akronimoa(ingelesez), Software Process Improvement and Capability dEtermination. ISO/IEC 15504. Garapen-prozesuak hobetzeko, ebaluatzekeo, 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	Etap	Prozesuaren egoera
I	Hasierakoa	Kaotikoa da, kontrol zaila du, erreaktiboa da
II	Kudeatuta	Proiektuen ezaugarriak ditu, askotan erreaktiboa 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 ezinbestekoa 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 lortzeaz zegoela erreklamatzeko, tramitazioaren faltan zegoen. Hala ere, administrazioaren erantzuna errekurtsua baiestearen 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.

Ebidentzia 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 fokatzeari, 2018ko *CMMI 2.0* bertsioan eta estandar internazionaletan sendoki indartuta geratu da.

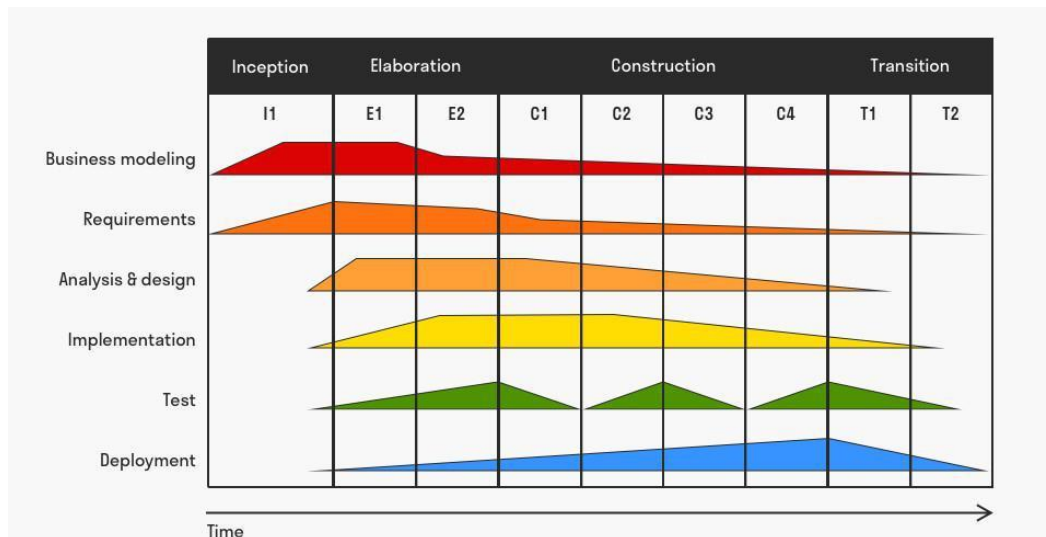
### 3.3 Metodologiak

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 ingeniariak edozein momentutan jakin behar du zer egin, noiz eta nola, bestela arazoak eta galerak agertzeko probabilitatea handituz joango da proiektuaren garapena luzatzen doan heinean.

Gainera, proiektuak zerotik 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, aldean arteko elkarriketa-fluxuari eutsi ahal izateko, garapen dinamikoagoa eta parte-hartzaileagoa 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 RUP<sup>3</sup> aurkitzen da. Softwarea garatzeko prozesu iteratibo bat da, Rational Software Corporation erakundeak sortua, IBM<sup>4</sup> 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.

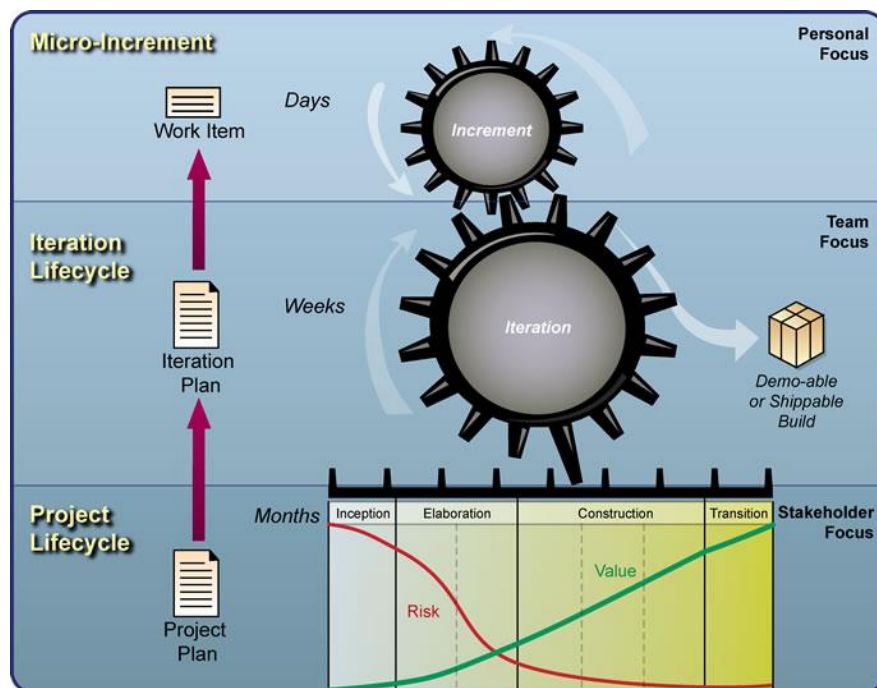


3.1. Irudia. RUPen prozesu iteratiboa, bizi-zikloaren faseak, iterazioak eta jarduerak.

RUPen oinarritutako metodologia sinple 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. Eraitza prozesu askoz sinpleagoa da, eta RUP printzipioekiko leiala izaten jarraitzen du. 3.2. Irudian ikus daiteke OpenUP prozesuaren laburpen bat.

<sup>3</sup> 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.

<sup>4</sup> IBM: Sigla(ingelesez), International Business Machines. Informatikarekin lotutako tresnak, programak eta zerbitzuak ekoiztu eta merkaturatzen dituen enpresa multinazionala da.



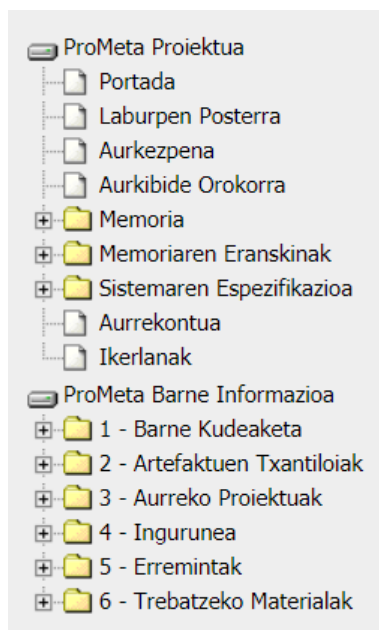
3.2. Irudia. OpenUP-en geruzak: mikro-gehikuntzak, iterazio bizi-zikloa eta proiektu bizi-zikloa.

### 3.4 Arauak

Proiektuen 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.

Espainian eta nazioartean araurik finkatuenak *UNE*<sup>5</sup> 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. Erreferentziatzko nazioarteko esparru eta estandarrak 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".

<sup>5</sup> UNE: Sigla(gazteleraz), 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, hornitzaile zein interesdunen arteko gatazkak sortuz.

## 3.5 BPM

*Business Process Management* negozio-prozesuak definitzera eta gauzatzera bideratutako metodologia edo ikuspegi 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 errepikakorrak 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 maneiatuz, egoeren jarraipena eginez, gertakariak mailakatuz, prozesuen sendotasuna eta trazabilitatea bermatuz, etab.
- Lan egiteko modua aldatzeko aukera eskaintzea, 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 aztertzeke metodoetan oinarritzen dira, merkatuaren joerak frogatzeko, hala nola zuzendaritza, heldutasuna eta parte-hartzaileak.

*Gartnerren* Koadrante Magikoa industria teknologikoaren azpisektore nagusiei buruz egiten duen merkatu-ikerketari 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 hornitzaileek 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 inplementatu daiteke, prozesu, erregela eta *workflow*entzako 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*-lengoaia propioa sortuz eta lengoaia horrekiko inferentzia motorra eraikiz.

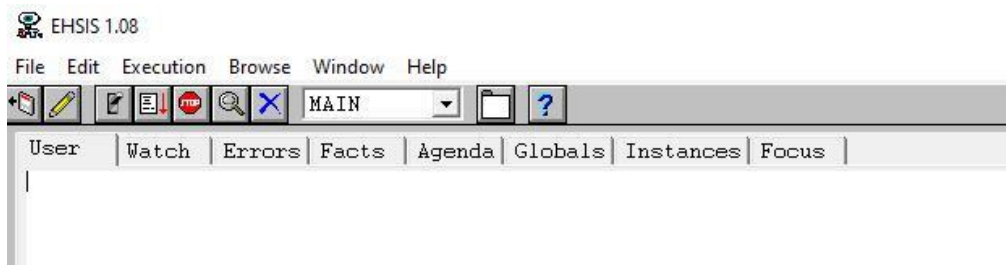
## 3.6 Inferentzia Motorrak

ProWF proiektuan teknologia propioa sortzea erabaki zen, *workflow*-lengoaia propioa sortuz eta lengoaia horrekiko inferentzia motorra eraikiz.

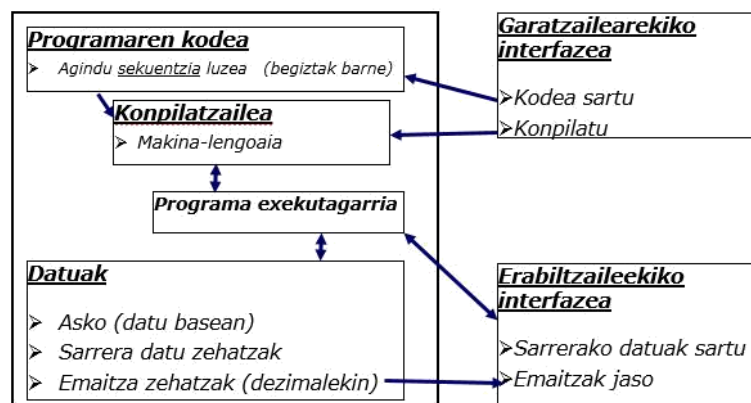
*CLIPS* (*C Language Integration Production System*) sistema adituak ekoizteko eta exekutatzeko garapen-ingurunea ematen duen tresna da, *NASA*k sortu eta jabari publikoan utzi zuena. Bere lengoaiak erregelak, objektuetara bideratutako programazioa eta programazio prozedurala erabiltzen du jakintzak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

- **Garraiaagarritasuna.** C lengoaiak ematen diona.
- **Integrazioa edo zabalkortasuna.** Programazio prozeduralari esker funtzio berri ahaltsuak sor daitezke. Service-oriented Architecture (SOA) estandarra erabiltuta BPM edo beste edozein sistemekin integragarria da.
- **Interakzio edo disziplinarreko garapena.** Formakuntza ezberdinetako pertsonen ideiak azkar inplementatzeko aukera lengoaia sinple eta ahaltsu bat erabiltuta, objektu eta erregeletan oinarrituta.
- **Egiaztapen edo balidazio errazagoa.** Horretarako, funtzio bereziak erabiliz.

EHSIS, ordea, Euskal Herriko Unibertsitateko (EHU) *ERABAKI* taldeak hedatutako ingurunea da, *CLIPS* 6.04, *FuzzyCLIPS*<sup>14</sup> 6.04, objektuetara eta gertaeretara bideratutako programazioa, interfazeen garapena, komunikazioa eta leihotetan oinarritutako ingurunea integratzen duena. *EHSIS* inguruneak (ikus 3.6 irudia), softwarea garatzeko baliabide tradizionaletan oinarritutako aplikazioak sortzeaz gain (ikus 3.7 irudia), *COOL*<sup>15</sup> lengoaia erabiltuta, ezagutzen oinarritutakoak ere sor ditzake sistemaren arkitektura egoki batekin baliatuz (ikus 3.8 irudia).



3.6. Irudia. EHSISen garapen ingurunea.



3.7. Irudia. Programazio tradizionalaren eskema.



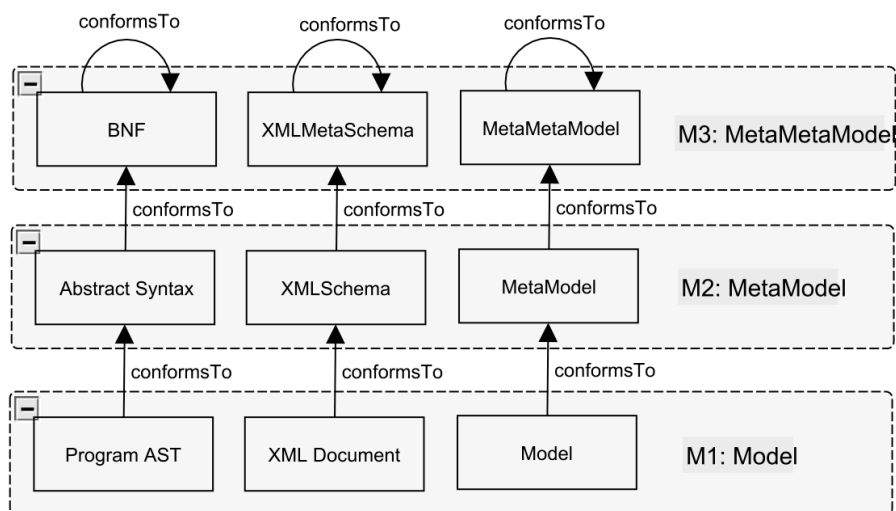


*EHSISen* garapen inguruneak badu bere produkzio bertsioa, *EHSIS\_RT* deitutakoa. Web-zerbitzuetarako bertsioa ere badu, *Mairi* deitutakoa.

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.** Ereduek 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 arrazoitze 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 arrazoitze beharrezkoak diren kontzeptuak dauzka. M3 mailako elementuek M2-ko elementuak kategorizatzen ditu. Meta-metaeredu esan ohi zaio.



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

## 3.8 CMS

ProWF eta ProMeta proiektuen 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 arabera 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

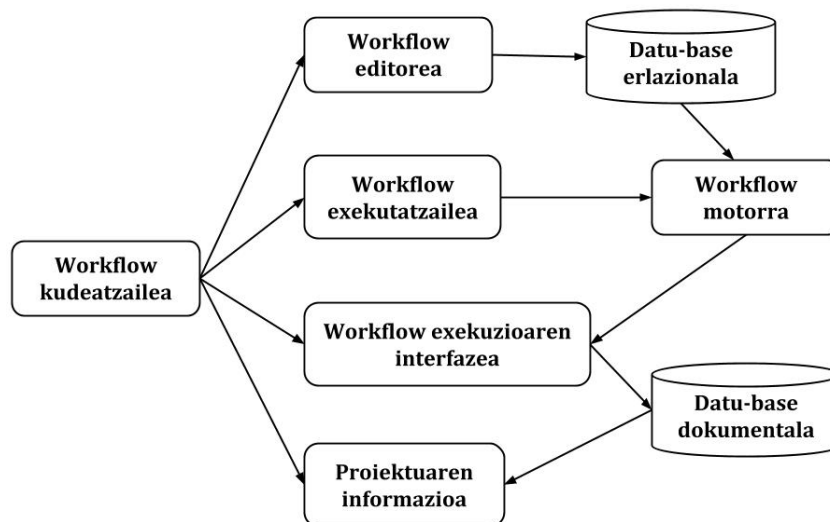
Esan bezala, ezinbestekoa da ProWF sistema ulertzea proiektu hau horren jarraipena delako. Horretarako, bertan zehazten diren rolak, sistemaren osagaiak eta azpisistemak deskribatuko dira.

*ProWF* sistemaren erabiltzaileek hurrengo rol hauekin lan egin behar dute:

- **Prozesu Sortzailea:** sortutako *workflow*-lengoaia baliatuz, softwareen garapenerako bizi-zikloa ezartzen duen metodologia definituko duena.
- **Analista:** sortutako *workflow*-ereduak adierazten duen bizi-zikloa exekutatzeaz eta aplikatzeaz arduratuko da.
- **Proiektuko Zuzendaria:** proiektuak sortu eta proiektuko partaideen rola esleituko ditu.
- **Kalitate Arduraduna:** *workflow*aren exekuzioan zehar sortuko diren artefaktuen kalitatea bermatzea du helburu, balorazioak emanez.

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*-lengoaia 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 *workflow*aren informazioa gordeko duen inferentzia motorra da.
- **Workflow exekutatzailea:** *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 biltegitratuko du.
- **Datu-base dokumentala:** *workflow*aren exekuzioan zehar bete behar diren artefaktuak biltegitratuko 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 osagaiak eta haien arteko erlazioak.

ProWF sistema bi azpisistema ezberdinetan bananduta egongo da:

- **Workflow Editor:** sortutako *workflow*-lengoaia 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* kudeatzailearen sarrera/irteerak maneiatzea du helburu. CMS baten bitartez kudeatutako web-aplikazioa izango da. Rolak: Proiektuko 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 izanburuak dioen moduan ProWF sistemak *software proiektuen 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 zehaztuko ditugu, eta gure ideiekin osatu. Izan ere, hobekuntza horietako batzuk aurrera eramango dira ProMeta proiektuan.

### 4.2 Abantailak eta Desabantailak

ProWF proiektuan 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-lengoiari lotutakoak eta workflowetan oinarritutako sistemari lotutakoak.

Workflow-lengoiari 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.
- Lengoiak softwarearen bizi-zikloaren ezaugarri esanguratsuenak harrapatzen ditu.

Workflowetan oinarritutako sistemari lotutakoak:

- Interfaze sinple 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 exekutatzeke 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 berdinetan sailka daitezke.

Workflow-lengoiari lotutakoak:

- Workflow-eredua nabigagarria denez, hainbat fitxategi eraldatu behar dira CLIPS lengoiako 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 osagaiak instalatuz.
- Drupalen bidez sortutako web-aplikazioak ez ditu erantzun azkarrak ematen. Gunearen orrialdez aldatzean kargatu behar diren modulu eta beste aspektuek errendimendua murrizten diote.
- Webguneko editorea testu soilean dagoenez, ez du aukerarik ematen formatua emateko. Adibidez, ezin da letra lodia erabili eta ez dago taulak betetzeko aukerarik.
- Webgunean artefaktuen informazioa betez ez dira lortzen artefaktuen txantiloien 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 lengoaiaren arteko eredu bat sortuz, urrats bakar baten bidez eraldaketa eginez eta kanpoko softwareak (Gephi, Protégé) erabiltzea ekidituz. Produktibitatea, azkartasuna eta mantenugarritasuna bilatuz.
- *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*<sup>6</sup> motako testu-editore bat inplementatzea. Softwarearen bizi-zikloa definitzen duten metodologia askotan taulak eta Excel orriak bete behar dira, prototipo honetan, ordea, ez dago taulak txertatzeko aukerarik.

---

<sup>6</sup> WYSIWYG: sigla (ingelesez), What You See Is What You Get. Testu-prozesadoreei eta beste testu-editore batzuei aplikatutako esaldi bat da, azkenengo emailta 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 ordeztu Drupalen datu-basean gordetzea. Horrek Drupalek eskaintzen dituen aukerak aprobetxatzen ditu, datuak bistaratzeko eta editatzeko aukerak, adibidez.
- Drupal webgunearen itxura hobetu, defektuzko itxura aldatuz. Itxura egoki bat aurkitu webgunerako, dropdown menuak onartzen dituen.

## 4.4 Prestakuntza

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

Softwarearen Kalitatea irakasgai 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 ingeniartzari buruzkoak.
- Proiektu bat aurkezteko webgunea sortu eta antolatu *CCII-2016N-02* araua jarraituz.

Softwarearen Garapen Industrialak irakasgaien beste gai hauek landu ziren:

- Model Driven Engineering (MDE) edo ereduak bideratutako ingeniartzaren oinarritzko kontzeptuak: metaeredua eta eredua.
- Domain Specific Language (DSL) edo domeinu zehatzeko lengoaien sorrera.
- ATLAS Transformation Language (ATL) erabilera ereduaren arteko eraldaketak egiteko.
- Eclipse Modelling Framework (EMF) tresnen erabilera MDE-rako.

Web Sistemak irakasgaien 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 Graduako edo Ingeniaritza Teknikoko titulazioak bete beharreko kompetentzia profesionalak eta Gradu Amaierako Lanen izaera profesionala ezartzen duen Errege Dekretua.

Administrazioa Publikoak ezarritako Sektore Publikoko Kontratuen legeak eta aurkeztutako kexak:

- [BOE-261-2007-18874](#). 30/2007 Legea, urriaren 30ekoa, Sektore Publikoko Kontratuei buruzkoa.
- [BOE-A-2011-17887](#). 3/2011 Legegintzako Errege Dekretua, azaroaren 14koa, Sektore Publikoko Kontratuei buruzko Legearen testu bategina onartzen duena.
- [BOE-A-2017-12902](#). 9/2017 Legea, azaroaren 8koa, 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 antolaketa arauak eta proiektuaren aurkezpenerako aplikatu den araua CCII-N2016 estandarra da.

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

### 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/Proiektua/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/Proiektua/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-gel-hy.github.io/EPF/ABRD/index.htm> helbidetik eskuratua

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

Gonzalez, J. L. (2019). *BETRADOK proiektua: Betekizunen trazabilitate inpaktu-analisi automatikoa eta dokumentazio formalaren sorkuntza 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 proiektua: Software proiektuen elaboraziorako workflowetan oinarritutako sistemaren sorkuntza 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 enpresa 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.

### 5.3.3 MDE

Model Driven Engineering (MDE) edo eredu bidezko ingeniarietza softwarea garatzeko metodologia erabili da.

## 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 guztien 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 guztiak eskaintzeaz gain bere ezaugarri propioak gehitzen ditu. Proiektuan Git errepositorio guztiak igotzeko 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. Tenporizadore 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

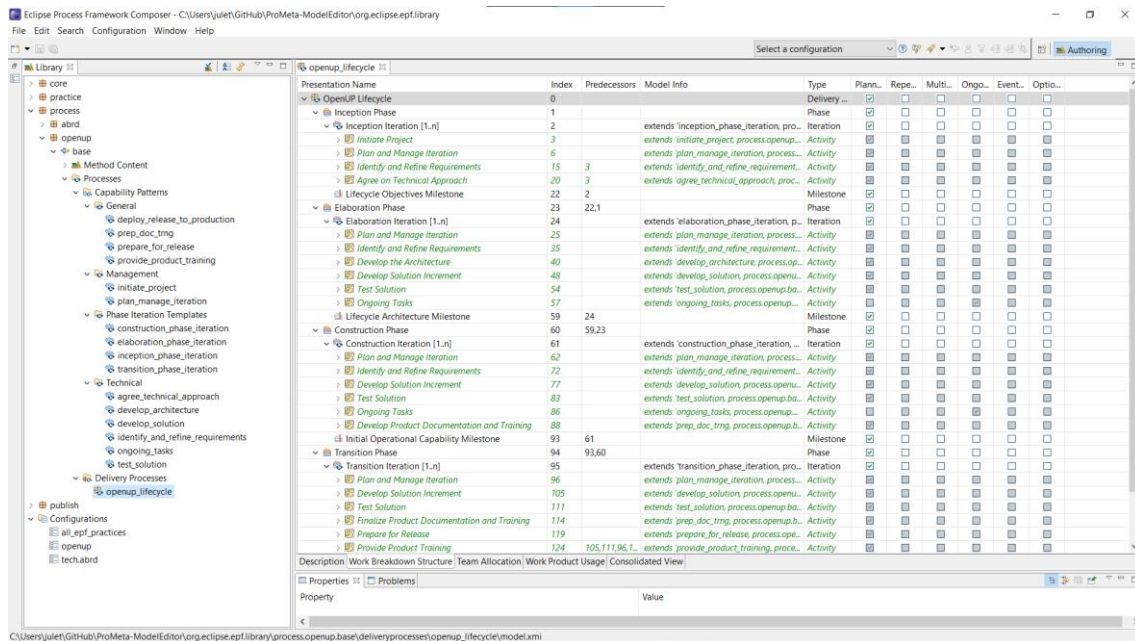
Sun Microsystems-ek garatutako programazio lengoaia eta plataforma informatikoa da. Plataforma makina birtual bat da eta Java programazio lengoaia eta garapen tresnak erabiliz garatutako aplikazioak exekutatzeko gai da. Proiektuan Java 8 bertsioa erabili da EPF Composer-erako eta Java 11 eta 15 Eclipseko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

#### 5.4.6 Eclipse IDE

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

#### 5.4.7 EPF Composer

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



5.1. 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 analizatzailer 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 lengoia da, XML fitxategiak transformatzeko balio duena. Proiektuan ModelEditor azpisteman XMI eredua UMara transformatzeko erabili da.

## 5.4.10 Visual Studio Code

Microsoftek garatutako kode editorea. Bertsio kontrola, sintaxi nabarmentzea 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.

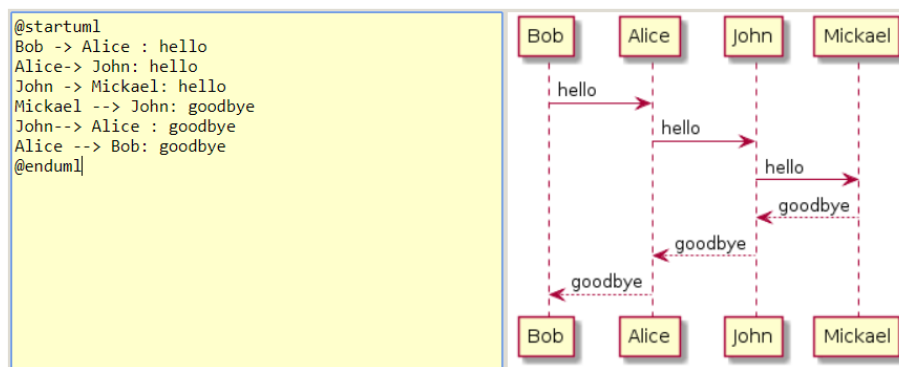
## 5.4.11 Microsoft Office

Microsoftek garatutako ofimatika 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 lengoia definitu bat erabiliz, UML diagramak sortzeko balio du. Adibidez, 5.2. Irudia lengoia deskriptiboa erakusten da eta horretatik sortutako diagrama simple baten eredua.

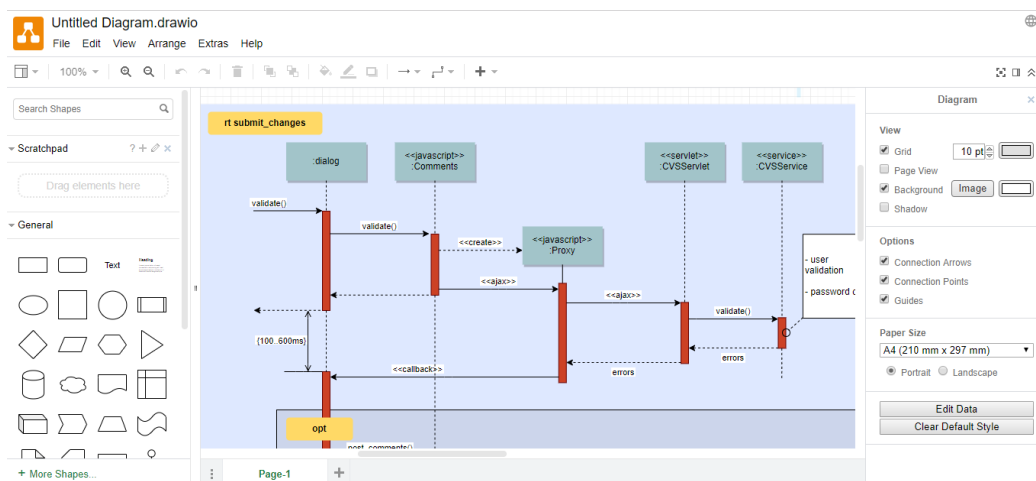
Proiektu honetan, OpenUP metodologiak eskatzen dituen erabilpen kasuen ereduak sortzeko erabili da.



5.2. 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. 5.3. Irudian dria diagrama adibide bat ikus daiteke.

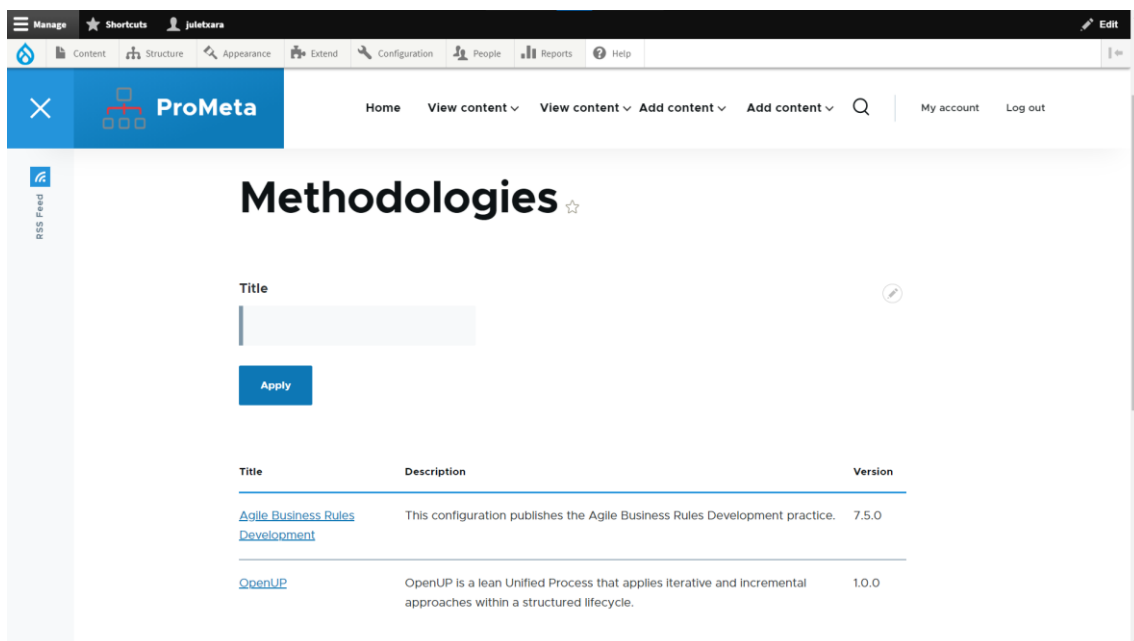


5.3. Irudia. Draw.io diagrama adibidea.

### 5.4.14 Drupal

Drupal edukiak kudeatzeko sistema edo CMS librea, modularra eta oso konfiguragarria da. Sistema dinamikoa da, hau da, zerbitzariaren edukiak modu finkoan biltegitatu beharrean, orrien testu-edukia eta beste konfigurazio batzuk datu-base batean biltegitatzen dira eta web-ingurune bat erabiliz editatzen dira. 5.4. Irudian proiektuan sortutako Drupal webgunea ikus daiteke.

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

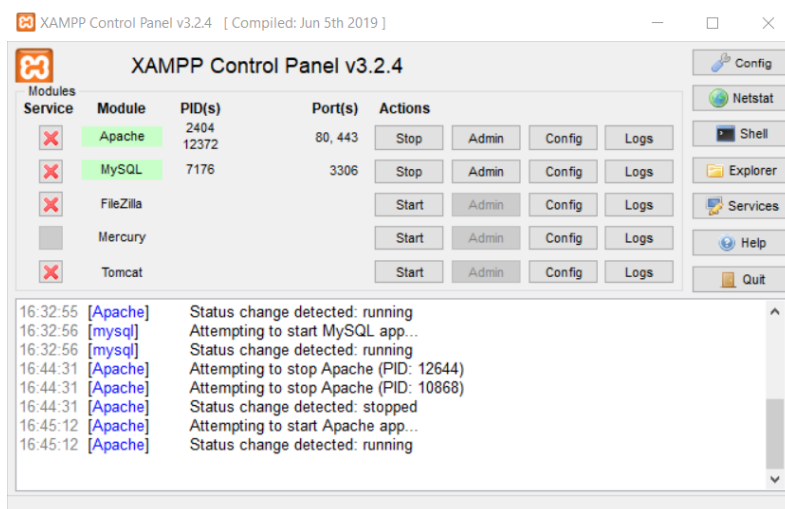


5.4. Irudia. Proiektuan sortutako Drupal webgunea.

#### 5.4.15 XAMPP

XAMPP software libreko pakete bat da, nagusiki *MySQL* datu-baseen kudeaketa sistema eta *Apache* web zerbitzaria integratzen duena. 5.5. Irudian ikus daiteke XAMPPen erabileraren adibide bat.

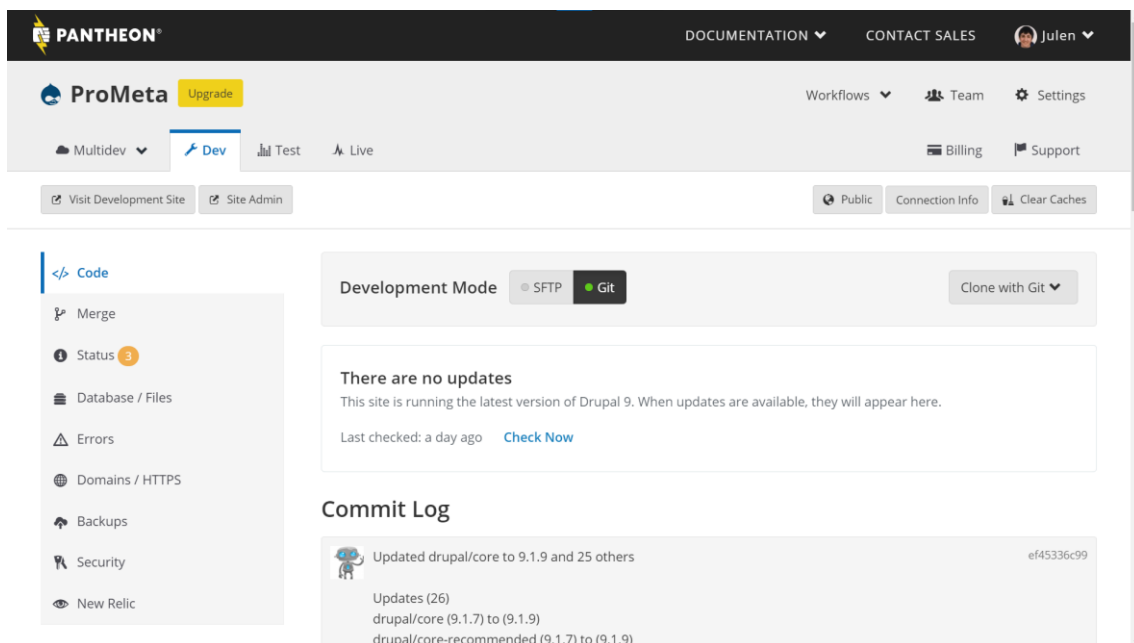
Proiektuan honetan Drupal webgunea lokalean garatzeko erabili da.



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

#### 5.4.16 Pantheon

Pantheon kode irekiko Drupal eta WordPress webguneetarako hosting plataforma da. 5.6. Irudian Pantheon kontrol panela ikus daiteke.



5.6. 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 lizentziapean libre mantentzeko asmoz sortutako fork bat da. Datu-base erlazional moduan erabili da proiektuan.

#### 5.4.18 phpMyAdmin

phpMyAdmin PHP-n idatzitako tresna da MySQL administrazioa web orrien bidez kudeatzeko. Proiektuan 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. Proiektuan Drupal-erako programazio lengoia moduan erabili da.

### 5.5 Ereduak

Proiektuan hainbat eredu erabili dira, batzuek dokumentazioarekin erlazionatuak eta beste batzuk inplementazioarekin.

#### 5.5.1 CCII-N2016-02

Arauen atalean aipatu den moduan, estandar honetan oinarrituta antolatu da memoria eta proiektuaren webgunea.

#### 5.5.2 OpenUP

OpenUP metodologiaren eredua erabili da. Eredu horrek metodologiaren informazio guztia du.

Metodologia hau jarraitzeko, bere webgunean artefaktu batzuen txantiloak daude eskuragarri. Txantilo horiek jarraituz OpenUP metodologiaren bitartez sortutako artefaktuak idatzi dira.

### 5.5.3 ABRD

Agile Business Rules Development metodologiaren eredua ere erabili da. Ereduek metodologiaren informazio guztia du.

Metodologia hau jarraitzeko, bere webgunean artefaktu batzuen txantiloak 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.

### 5.5.5 Ecore

EMF-ko Ecore meta-metaeredua erabili da UMA metaeredua definitzeko. Meta-metaeredu honen helburua edozein metaeredu modelatu ahal izatea da.

## 5.6 Metrikak

Proiektuaren helburuekin erlazionaturako 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 proiektuko 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.

### 5.6.4 Kalitatea

Proiektuaren kalitatea kudeatu da.

## 5.7 Prototipoak

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

### 5.7.1 ProMeta ModelEditor

Proiektuko metaeredu, eredu eta editoreak biltzen dituen prototipoa.

### 5.7.2 ProMeta IO-System

Proiektuko Drupal webgunea eta datu-basea biltzen dituen prototipoa.

## 5.8 Beste Erreferentziak

## 6 Definizioak eta Laburdurak

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

### 1.1 Bizagi

Bi produktu osagarri dituen softwarea da, prozesuen modelatzaile (Bizagi Modeler) bat eta BPMren suite ofimaticoa bat (Bizagi Studio).

### 1.2 BPM

Sigla(ingelesez), Business Process Management. Enpresei prozesuak automatikoki modelatzeko, inplementatzeko eta exekutatzeak aukera ematen dien software teknologia da.

### 1.3 CCII

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

### 1.4 CCII-N2016-02

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

### 1.5 CMMI

Capability Maturity Model Integration. Software-sistemak garatzeko, mantentzeko eta erabiltzeko, prozesuak hobetzeko eta ebaluatzeak ereduak da, CMMi Institutuak administratutakoa.

### 1.6 CMS

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

### 1.7 DOT

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

### 1.8 Drupal

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

### 1.9 IBM

Sigla(ingelesez), International Business Machines. Informatikarekin lotutako tresnak, programak eta zerbitzuak ekoiztu eta merkaturatzen dituen enpresa multinazionala da.

### 1.10 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 proiektuaren bizi-zikloaren lau fasetan banatzen du: Hasiera, Elaborazioa, Eraikuntza eta Trantsizioa. Informazio gehiago: [OpenUp 1.0](#) edo [OpenUp 1.5](#).

### 1.11 PlantUML

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



## 1.12 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.

## 1.13 ProWF

Software proiektuen 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.

## 1.14 RUP

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

## 1.15 SPICE

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

## 1.16 UML

Unified Modeling Language (Modelaketarako lengoiaia bateratua) sistemak zehaztu, diseinatu eta eraikitzeko lengoiaia da, printzipioz objektuei orientatutako programaziorako prestatuta dagoena. UML aplikazio baten garapen fase guztiak modelatzeko lengoiaia homogeneo bat definitzen saiatzen da, bezeroaren zehaztapenetatik hasita programatzailearen diseinu xehera arte.

## 1.17 UNE

Sigla(gaztelaraz), Una Norma Española. Comités Técnicos de Normalización (CTN) batzordeak sortutako arauen, arau esperimentalen eta txostenen (estandarrek) multzoak dira.

## 1.18 URPS

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

## 1.19 Workflow

Aspektu operazionalekin 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.

## 1.20 WYSIWYG

Sigla (ingelesez), What You See Is What You Get. Testu-prozesadorei 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-funtzionalak sistemaren funtzionalitateekin zuzenean erlazionatuta ez dauden betebeharrak 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 estandarren 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 URPS<sup>7</sup> 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 memorizatzeke 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. Sistemak 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 berreskuratu 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 proiektuetan aritzen direnentzat ulerterrazak eta ezagunak izan behar dira.

### 7.4.2 Diseinu eta Nabigazio Betekizunak

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

---

<sup>7</sup> 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 kontsistentzia mantendu beharko du, itxuraz berdina 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

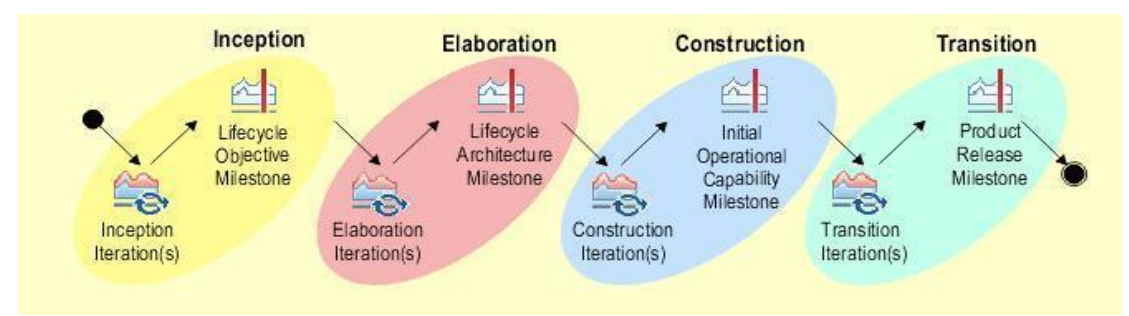
Ez da aurreikusten erabiltzailean pertsonalizazio aukerarik izatea. Baliteke, interfazearen itxura edo hizkuntza aldatzeko aukeraren bat gehitzea.

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 fasea kanpoan geratu da. Hurrengo zerrendan deskribatzen dira irismena definitzen duten ezaugarriak:

- OpenUP metodologiak eskatutako dokumentuak betetzea. Horretarako OpenUP metodologiak bere webgunean eskaintzen dituen txantiloak 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.
- ProMeta ModelEditor azpisistemaren garapena.
- ProMeta IO-System azpisistemaren garapena.
- Proiektuaren memoria idaztea.
- Proiektuaren defentsa prestatzea. Horretarako, memorian idatzitako aspektu guztiak laburbiltzen dituen aurkezpen bat prestatuz.
- Proiektuaren posterra egitea.

8.2 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 murritzapenak deskribatuko dira.

### 9.1 Hipotesiak

Proiektuarekin lanean hasteko hipotesi batzuk atera 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 aurreratueta 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 koherentzia eta datu-basearen produktibitatea handitzea lortuz.
- Sistema iteratiboki hobetzen joango denez, estandarretan oinarritutako garapenak bere mantenua eta hedapena errazten ditu.

### 9.2 Murritzapenak

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

- Kostu ekonomikoa ez da batere aldatu hasierako planteamendutik. Izan ere, proiektuan zehar ez da kostu gehigarriarik sortu, erabilitako teknologia guztiak doakoak izan direlako.
- Denborari dagokionez, ekainaren 20rako proiektua bukatzeko murritzapena bete behar da.
- Kalitateari dagokionez, proiektuak denbora eta kontu murritzapenen barruan kalitate onargarria izan behar du.
- Betekizunen ingeniari-tza eta bizi-zikloa definitzen duen metodologia bat jarraitzea proiektuaren elaborazio eta garapenerako: OpenUP. Bezeroak egiten duen exigentzia da metodologia bat erabiltzea.
- Proiektuaren dokumentuen antolaketarako CCII-2016N-02 estandarra erabiltzea.
- Sortuko den software proiektuen elaboraziorako sistema web bidez atzigarria izan behar du.

## 10 Aukeren Azterketa eta Egingarritasuna

Kapitulu honetan proiektuko 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 bitartez 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 konfiguratzuz, 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.

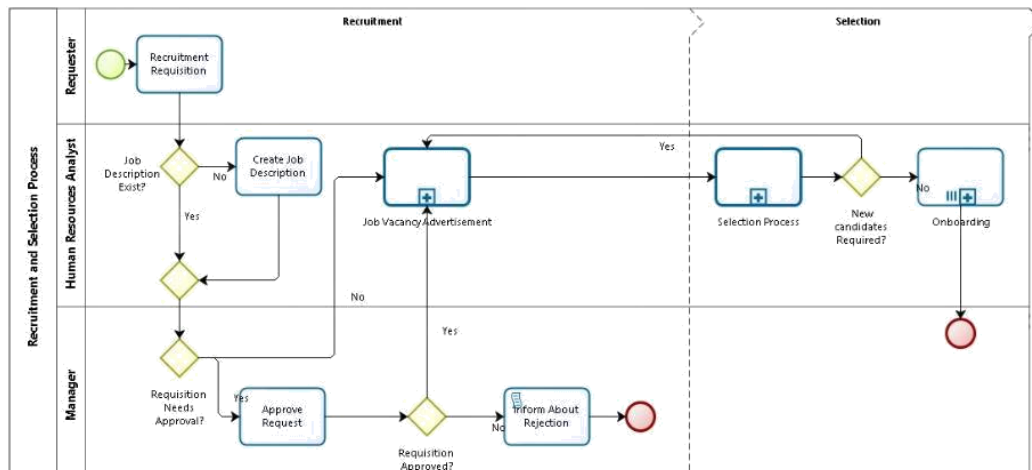
Proiektu honetarako bideragarria izango litzateke software hau erabiltzea, hurrengo bi arrazoiengatik:

- 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 lengoaiaren (*BPMN*) arauak errespetatuz, *workflow* azpi-lengoaia bat sortu izango litzateke.
- Soluzioaren sarrera/irteerak kudeatzeko *Bizagi Studio*ren 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* ofimatikoa 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.

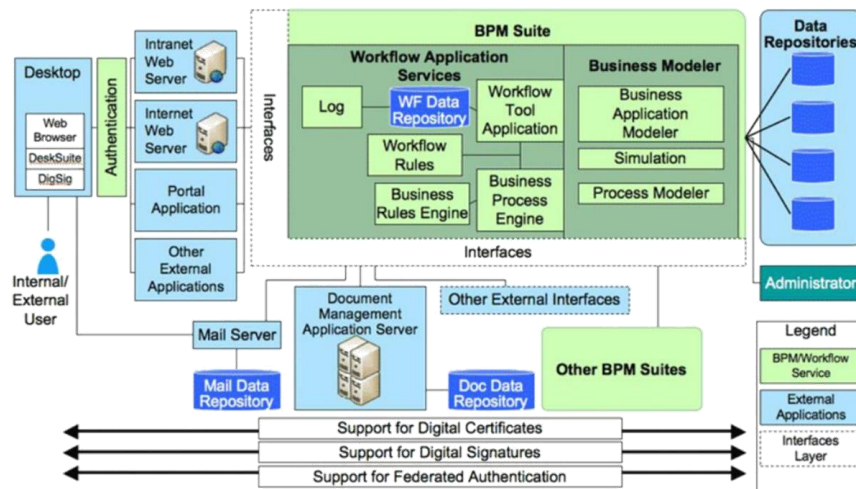
### 10.1.2 ProWF

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

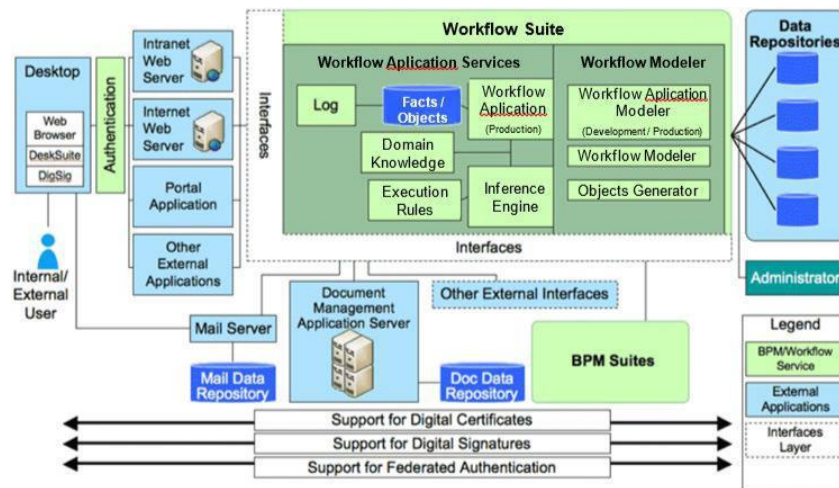
*Workflow*etan oinarritutako web-aplikazio bat zerotik eraikitzea apustu handia zen, denbora gehiago eman beharko litzateke arkitekturaren garapenean, baina, ordea, pisuzko arazoak zituen:

- *Bizagiren* desabantailak oso kaltegarriak ziran. Honela, menpekotasun teknologikoa saihestu eta kostu ekonomikoa asko gutxitzen da.
- *Workflow*-lengoaia 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, *workflow*etan oinarritutako arkitektura hau zerotik eraikitzea ez da lan erraza eta sistema konplexua bat sortu behar da, gainera, *Bizagik* eta *BPMN*<sup>18</sup> estandarrak eskaintzen dituzten funtzionalitate asko galduko ditugu. 10.2. Irudian *Bizagiren* arkitektura ikus daiteke eta 10.3. Irudian *workflow*etan oinarritutako arkitektura propioaren. Esan beharra dago agertzen diren funtzionalitate guztiak ez direla inplementatuko, *BPM* eta *workflow* sistema kudeatzailea alderatzea bakarrik du helburu.



10.2. Irudia. Bizagiren arkitektura.



10.3. Irudia. Workflowetan oinarritutako arkitektura.

### 10.1.3 ProMeta

ProMeta proiektuan sistemaren abstrakzio-maila igotzea erabaki da. Metodologiaren informazioa workflow lengoian edo BPMNn eskuz definitu ordez, metaeredu eta ereduen bidez gordetzen da. Horrela, ereduen informazioa berrerabil daiteke eta beste ereduetara eraldaketak egin. Ereduen 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. Izan ere, BPMNren eta workflow-lengoaiaren metaereduak izanik, eraldaketa egitea ez litzateke horrenbeste lan izango. Gainera, eredutik datu-baserako SQL kodea edo webgunerako HTML kodea sortzeko aukera ere egongo litzateke.

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

## 10.2 CMS aukerak

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 nabarmentzen diren puntuetan.

Ezaugarria	Wordpress	Joomla	Drupal	Oharrak
<b>Kode irekia</b>	✓	✓	✓	-
<b>Dokumentazio sinple eta ondo egituratuta</b>	✓	✓	✓	-
<b>Komunitate aktiboa eta foroak</b>	✓	✗	✓	Hemen Wordpress da nagusia.
<b>Estentsio gehigarri eta moduluen hedapena</b>	✓	✗	✓	<i>Joomla</i> estentsio gehigarriak ditu ere, baina ez askorik.
<b>Beginner-friendly (erabiltzaile berrientzako erabilerraza)</b>	✓	✗	✗	<i>Joomla</i> eta <i>Drupalekin</i> zaila izan daiteke hasieran bere konfigurazioa edo gunearen itxura aldatzen jakitea edo
<b>Erabiltzaileen kudeaketa erraza</b>	✗	✗	✓	<i>Drupalen</i> bitartez rolak sortu/esleitu eta baimen espezifikoak eman daitezke
<b>Programazio-lengoaia</b>	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:

- *Drupalen* 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 erakundeek txostenetan, CMS atalean, liderra den *Acquia* enpresak *Drupal* erabiltzen du oinarri bezala.

ProMeta proiektuak CMSari dagokionez antzeko helburuak dituenaz, aurreko arrazoi guztiak mantentzen dira. Gainera, *Drupal* erabiltzeak orain beste abantaila bat du, aurreko proiektuaren zati batzuk berrerabiltzeko 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 exekutatzeko inguruneetarako, GPL lizentziarekin eta software libre da.

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

Arrakastaren arrazoietakoa bat garatzaile eta diseinatzaileen komunitate izugarria da, bere muinean programatzeaz edo komunitatearentzako pluginak eta txantiloak 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 lizentziarekin, 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 koherentzia.

Drupal-en diseinua bereziki egokia da Interneteko komunitateak eraikitzeke 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 dinamikoa 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) lizentziapean argitaratua.

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

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

## 10.3 Dokumentazioa Hosting Aukerak

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 eraikitzeke 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 sinpleagoa 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 sinple bat eraikitzea bakarrik interesatzen zaigunez, aurreko aukerekin nahikoa daukagu.

## 10.4 Drupal Hosting Aukerak

### 10.4.1 000webhost

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

Ez dauka GitHubekin integraziorik 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 sinplean instalatzeko ez dago prestatuta.

### 10.4.3 Acquia

Ez dauka doako planik <https://www.acquia.com/choosing-right-acquia-cloud-platform>. Drupal modu sinplean 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 sinplea <https://pantheon.io/docs/guides/collaborative-development>. Drupal modu sinplean 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 sinplean 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](https://elements.heroku.com/addons/cleardb).

## 10.5.2 PostgreSQL

Drupal-ekin bateragarria da, ematen dituen aukeretakoa bat da. Xampp-ek defektuz ez du eskaintzen, baina gehigarri moduan aktibatzeak 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-aren eta EMF-ren bertsio berrieke ez da bateragarria.

### 10.6.3 Xtext

Aukera onena Xtext eta Xtend erabiliz ereduak 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 dudana softwarea instalatuta. Makina birtualean softwarea instala daiteke baina errazagoa da nire konputagailuan falta den softwarea instalatzea. 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 proiektuetan.

### 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 Metaereduentzako tresnak

### 10.9.1 Eclipseren tresnak

Eclipse Modeling Framework (EMF), ATL Transformation Language (ATL) eta XTEXT izan daitezke erabili beharreko tresnak. Ez dakigu guztiak erabiliko ditugun, baina horiekin nahikoa izango dela uste dugu. Tresna horiek ezagutzen ditut aurreko urteko SGI irakasgaitik.

## 10.10 Metaeredua

### 10.10.1 SPEM

Software Process Engineering metamodel. SPEM metaeredua cmof formatuan dago.

### 10.10.2 UMA

Unified Method Architecture. UMA metaeredua ecore formatuan dago.

## 10.11 Metodologia aukeratu

### 10.11.1 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 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.

### 10.11.2 RUP

Kontuan hartuta proiektua eta proiektuko 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, sinpleagoa 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.

### 10.11.3 ABRD

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

## 10.12 Metodologia definitu

### 10.12.1 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 biteko bertsioa eskatzen du.



### 10.12.2 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.

### 10.12.3 Editore grafikoa

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

### 10.12.4 Testu editorea

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

## 10.13 Prozesua bistaratu

### 10.13.1 DOT

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

### 10.13.2 XPDL

XPDL ere aukera egokia izan daiteke estandarra delako. Baina, oraingoz DOT aukera hobe izango litzateke, sinpleagoa delako.

### 10.13.3 Webgunea

Prozesua bistaratzeko 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 definitzea komenigarria da.

## 10.14 Proiektuaren kudeaketa

### 10.14.1 ClickUp

### 10.14.2 Quire

### 10.14.3 Trello

## 10.15 Denboraren kontrola

### 10.15.1 Toggl Track

Tenporizadore 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. Gailu mota guztietarako aplikazioak daude, mugikorrerako ordenagailurako, nabigatzailerako, etab.

### 10.15.2 Clockify

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

### 10.15.3 WakaTime

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



## 10.16 Gantt diagrama

### 10.16.1 Teamgantt

### 10.16.2 Elegantt

### 10.16.3 Ganttproject

## 10.17 Diagramak

### 10.17.1 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.

### 10.17.2 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.

## 11 Proposatutako Sistemaren Deskribapena

Kapitulu honetan planteatutako arazoa konpontzeko proposatzen den sistema, bere osagaiak 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 ingeneriari rola izango du.
- **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.

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.

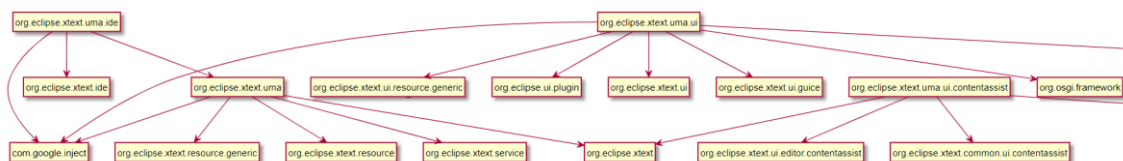
### 11.2 ModelEditor

#### 11.2.1 Arkitektura

ModelEditor azpisistemak honako osagaiak izango ditu:

- **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 Sinkronizatzailea:** Metodologiaren informazioa editore batekin aldatutakoan bestea ere aldatzeaz arduratzen da.
- **Kode Sortzailea:** Metodologiaren informazioa datu-basean gorde ahal izateko SQL kodea sortzen du.

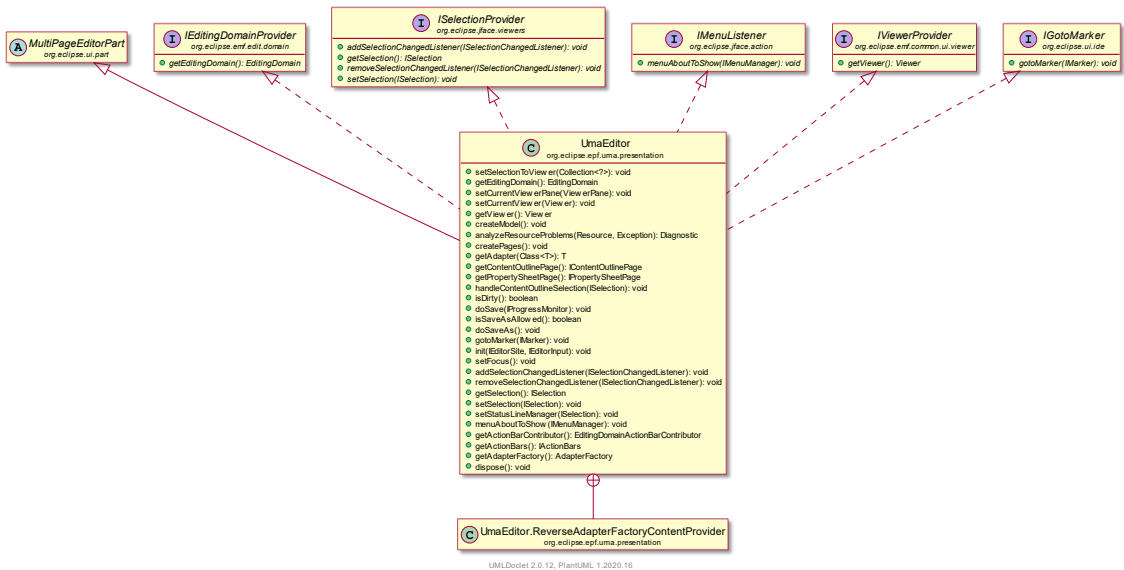
### 11.2.3 Pakete Dependenzia Diagrama



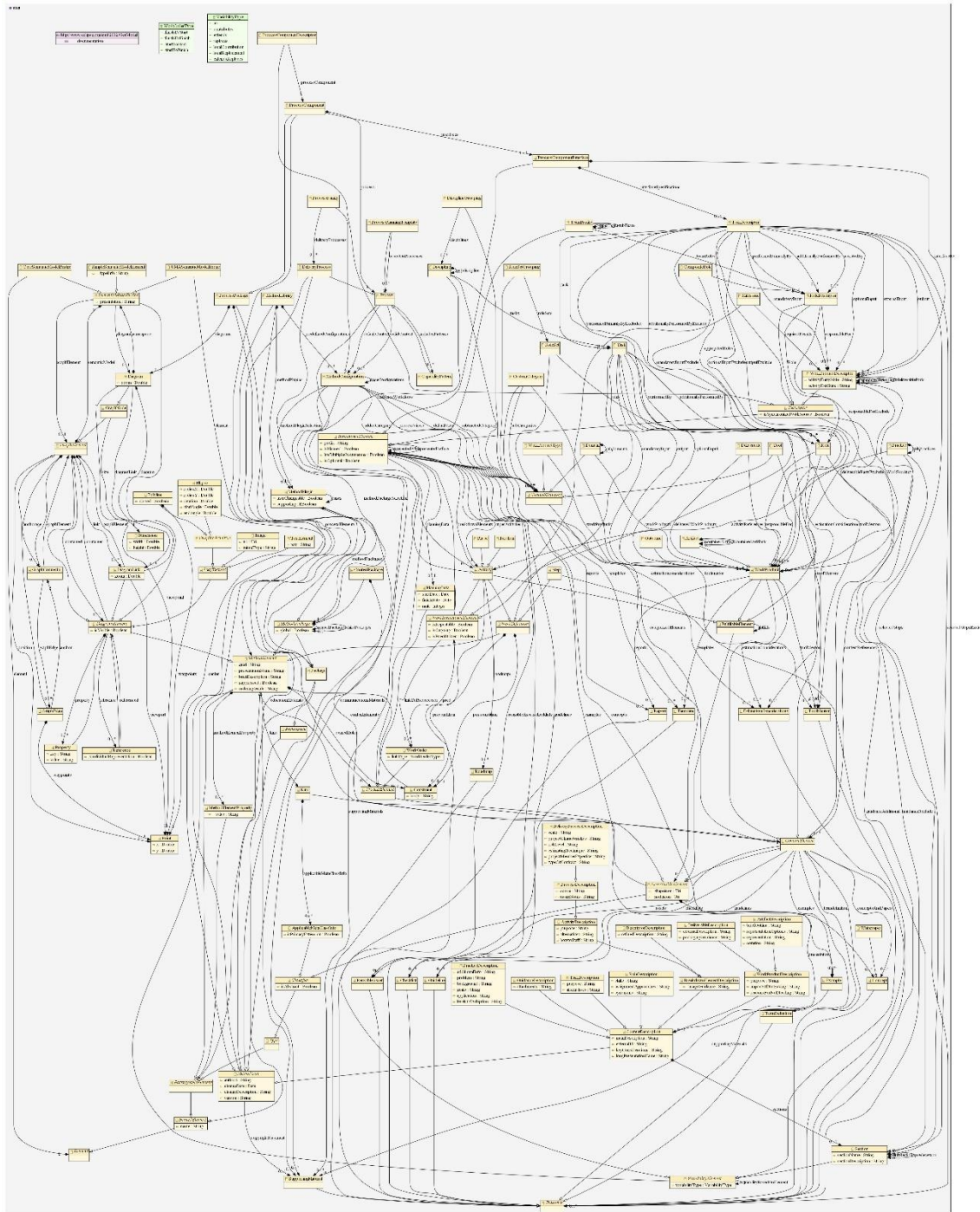
#### 11.2.4 Pakete Diagrama

UMLDoclet 2.0.12, PlantUML 1.2020.16

11.2.5 Klase Diagrama



## 11.2.6 Diseinua



## 11.2.7 Garapena

## 11.2.8 Proba

## 11.3 IO-System

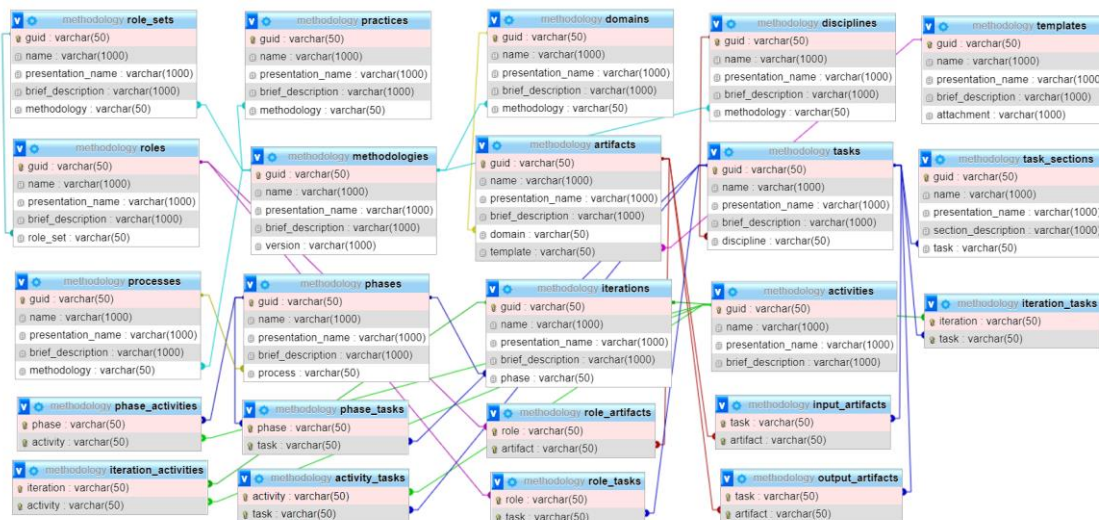
### 11.3.1 Arkitektura

IO-System azpisistemak honako osagaiak izango ditu:

- **Datu-basea:** Datu-base erlazionalak ereduaren informazio garrantzitsua gordeko du, Drupal webgunerako beharrezkoa izan daitekeena. Hau da, OpenUP eta ABRD metodologiaren 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 inportatzailea:** Datu-inportatzailea Drupal modulu multzo bat izango da. Hauen ardura aurretik aipatutako lehenengodatu-basetik Drupal datu-basera edukia inportatzea 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, administratzaileak aukera gehiagarri asko izango ditu webgunea kudeatzeko interfaze bidez.

### 11.3.2 Analisia

### 11.3.3 Diseinua



### 11.3.4 Garapena

### 11.3.5 Proba

## 11.4 Hedapena

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

Proiektuaren dokumentazioaren kodea 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 proiekturen webguneekin ere berdina egin dut, ProWF eta BETRADOK. ProWF proiektu honen aurrekaria denez kontsultatzeko behar dut. Eta BETRADOK proiektua antzekoa denez ongi etorriko zait ideiak hartzeko. ProWF proiektuaren errepositorioa <https://github.com/juletx/BETRADOK> eta webgunea <https://juletx.github.io/ProWF/>. BETRADOK proiektuaren GitHub errepositorioa <https://github.com/juletx/BETRADOK> eta GitHub Pages webgunea <https://juletx.github.io/BETRADOK/>.

Proiektuaren metaereduen atalaren inplementazioaren kodea 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](https://dev-prometa.pantheonsite.io/) 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 kontsultatu 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 [GAUR](#)en matrikulatu behar da eta zuzendariak oniritzia eman behar du. Ondoren, ikasleak lana [ADDI](#) plataformara igoko du. Horrez gain, ikasleak bere lanaren posterra bidali behar du [dif.gral@ehu.eus](mailto: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 emailtzen **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 guztietan identifikatutako arriskuak zerrendatzen dira eta horien probabilitatea eta inpaktua zehazten da, arazoaren magnitudea neurtzeko. Gainera, arriskuei aurre egiteko mitigazio estrategiak definitzen dira.

ID	Data	Izena	Deskribapena	Inpaktua	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 egindakoa dokumentu batean idatzi, instalatuko dudako softwarearen espezifikazioak ondo irakurri.
2	25/02/2021	Lan tresnekin arazoak	Lan tresnak erabiltzerakoan ager daitezkeen arazoak: bertsio bateraezintasunak, pluginak, erroreak...	4	40%	1,6	Instalatzeko ongi ziurtatu bertsioak bateragarriak direla. Beharrezkoak diren pluginak bakarrik instalatu.
3	25/02/2021	OpenUp ereduarekin arazoak	OpenUp eredu zaharra denez, baliteke guztiz osatua ez egotea eta bateraezintasunak egotea UMA metaereduarekin.	3	60%	1,8	Ahalik eta bateragarriak diren OpenUp eta UMA bertsioak 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 eredu 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 tresnarentzako. Beharrezkoa 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 memoria 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 amaitzeko 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 garrantzitsuenetan zentratu. Ez dago dena bukatu beharrik, etorkizuneko lan bezala utz daiteke.

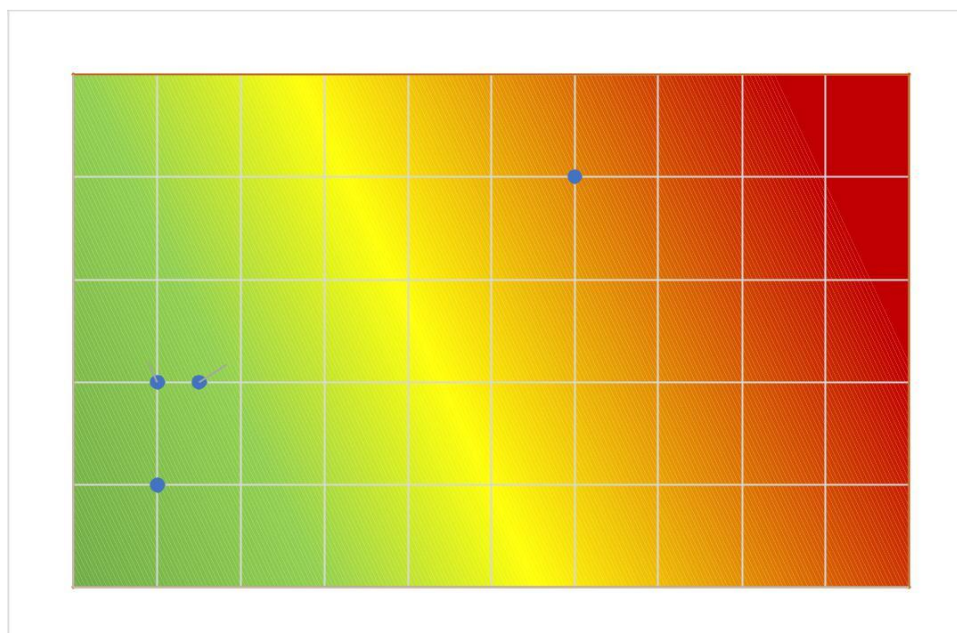


9	25/05/2021	Egindako lanaren galera	Egindako lana galtzeak lana berregin behar izatea eragin dezake. Honek denbora galtzeaz gain frustrazioa eragiten du.	3	50%	1,5	Bertsio kontrola erabili proiektuaren informazio guztia gordetzeko. Datu guztien segurtasun kopiak egin egunero.
10	25/05/2021	Ordenagailuarekin 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 inpaktua eta probabilitatea bidertzen dira, eta magnitudea kalkulatzeko. 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.

## 12.3 Arazoak

Aurreko arriskuak kontuan hartuz, interesgarria da jakitea proiektuan egon diren arazoak. Horrela, jakin dezakegu identifikatutako arriskuetatik zein gertatu diren, eta zein arrisku ez ditugun identifikatu. 12.2. Taulan ikus daitezke proiektuko arazo nagusiak.

Arazoa	Deskribapena
Ordenagailuarekin arazoak	Urtarrilaren bukaera aldera ordenagailuarekin arazoak izan nituen. Pantaila urdinean geratu zitzaidan. Konpontzen saiatu nintzen baina azkenean Windows berrinstalatu behar izan nuen. Informaziorik ez nuen galdu GitHub-en nuelako guztia eta segurtasun kopiak nituelako. Hala ere, sistema eta aplikazioak berrinstalatzen denbora dezente pasatu nuen. Horren guztiaren ondorioz astebete inguru galdu nuen.
Xtext-ekin arazoak	<p>Arazo pila bat ari naiz edukitzen xtext-kin eta horrek denbora asko galtzea eragin du. Bat-batean erroreak agertzen dira eta oso zaila da konpontzea. Batzuetan kodea berriz sortzea nahikoa da. Bestetan sortutako kodea ezabatu eta berriz sortu behar da. Horrek funtzionatzen ez badu, metadata ezabatu eta proiektuak berriz inportatu. Kodea sortzen denbora asko behar duenez, prozesu hau asko luza daiteke.</p> <p>Erroreak hobeto kontrolatzen ikasi dut eta gertatutakoan azkarrago konpontzen. Oraindik denbora kentzen dit, baina nahiko ondo konpondu naiz.</p>
OpenUp ereduarekin arazoak	<p>EPF Composer-etik exportatutako ereduak ez du zuzenean balio editore grafikoarentzako. Hainbat transformazio egin behar dira erroreak konpontzeko.</p> <p>Transformazio guztiak bukatuta daude, EPF Practices liburtegi osoa daukat editore grafikoan.</p>
Editoreen arteko sinkronizazio arazoak	Editore grafikoan erreferentziak ondo dauden arren, testu editorera pasatzean errorea ematen dute. Errorea ematen jarraitzen du, oraingoz alde batera utzi dut beste atazekin jarraitzeko.

12.2. Taula. Proiektuan zehar edukitako arazo nagusiak

## 13 Proiektuaren Antolamendua eta Kudeaketa

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

### 13.1 Proiektuaren Antolamendua

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

#### 13.1.1 Taldekideak eta rola

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 inplementazioa publikoki eskuragarri egongo dira GitHub bidez eta webgune bidez. Printzipioz, lana bukatu ondoren ere eskuragarri jarraituko dute, edozeinek kontsultatu 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 kontsultatu 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. Bilerak gehienak BBC bidez online egingo ditugu eta beste batzuk presentzialki Informatika Fakultatean. Printzipioz, bilerak astero egingo ditugu aurretik adostutako ordu batean. Beharrezkoa bada eguna edo ordua aldatu dezakegu. Bilerekin gain posta elektronikoa erabiliko dugu elkarrekin komunikatzeko.

### 13.2 Proiektuaren Kudeaketa

Aurretik 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.

Proiektua zuzentzeko eta kudeatzeko ardura bi aktore garrantzitsuenena izango da: Julen Etxaniz egilea eta Juan Manuel Pikatza tutorea. Proiektua 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 zeregina da proiektuaren irismena kudeatzea, hots, proiektuaren espezifikazioak eta denbora zeintzuk diren kontuan hartuta, proiektua zenbateraino garatu daitekeen estimatzea eta kudeatzea.

Proiektuaren Ikuspegia bete den lehenengo dokumentua izan da. Izan 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 dokumentuak erabili dira. Ikuspegian betekizun funtzional eta ez-funtzionalak zehaztu dira. Betebeharren Espezifikazioan beste betekizunak definitu dira: kalitatea, interfazeak, konplimendua etab. Erabilpen Kasuen dokumentuek eta ereduak 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 xedea 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 bakoitzean 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 bukatzeko denborarik ez izatea eta epea atzeratu behar izatea. Horrek proiektua uztailan 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 behar. 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 guztiak Aurrekontuan definitzen dira, doakoak diren tresnak ere zerrendatuz.

### 13.2.5 Kalitate Kudeaketa

Atal honetan proiektuan zehar helburu diren kalitate betekizunak ahalik eta hoberen 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 arruntren 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 sekzio guztiekin beteta daudela. Gainera, webgunearen kalitate kontrola egin behar da.

Proiektuko rolen eta kalitate kudeaketako arduren banaketa 13.1. Taulan ikus daiteke. Proiektuaren egileak etengabe hiru aspektuak kontrolatu beharko ditu proiektuaren bukaerara arte. Proiektuaren 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 rola eta ardurak.

### 13.2.6 Giza Baliabideen Kudeaketa

Atal honen helburua da proiektuan parte hartuko duten aktore guztien kudeaketa deskribatzea da.

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.

Beraz, giza baliabideen kudeaketa nahiko sinplea 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 hoberena edukitzeko lehenengo interesatuak identifikatu dira. Ondoren, komunikazioak planifikatu dira. Komunikaziorako 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 egoeratik eta ordutegia aukeratzea errazagoa delako. Gainera, pantaila partekatzeko aukera izateak bileretan asko lagundu digu.

### 13.2.8 Arriskuen Kudeaketa

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

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

### 13.2.9 Erosketen Kudeaketa

Atal honen helburua proiektuko 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 erosi beharko balitz egileak kudeatuko luke tutorearen laguntzarekin.

### 13.2.10 Interesatuen Kudeaketa

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

Interesatuak eta horien erantzukizunak proiektuaren hasieran finkatuko dira Ikuspegia dokumentuan. Interesatu garrantzitsuenak kontuan hartzen direla ziurtatu beharko da proiektuaren 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 atazen denborak, iterazioak eta desbiderapenak zehaztea da. Denboraren jarraipena egiteko Toggle Track tresna erabili da. Tresna honekin proiektuko denbora guztiak kontrolatu dira, tenporizadore bat erabiliz.

Azpiatazak iterazioka 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 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 hasiera	2021/01/14
2. Iterazioaren hasiera	2021/02/15
3. Iterazioaren hasiera	2021/03/16
4. Iterazioaren hasiera	2021/04/17
5. Iterazioaren hasiera	2021/05/18
Lana GAURen matrikulatu	2021/06/11
Lanaren posterra bidali <a href="mailto:dif.gral@ehu.eus">dif.gral@ehu.eus</a> helbidera	2021/06/11
Lana ADDI plataforma digitalera igo	2021/06/20
Proiektuaren amaiera	2021/06/20

14.1. Taula. Proiektuko mugarri garrantzitsuak.

### 14.2 Lan-atazak

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

Hurrengo taulak adierazten du proiektuan 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

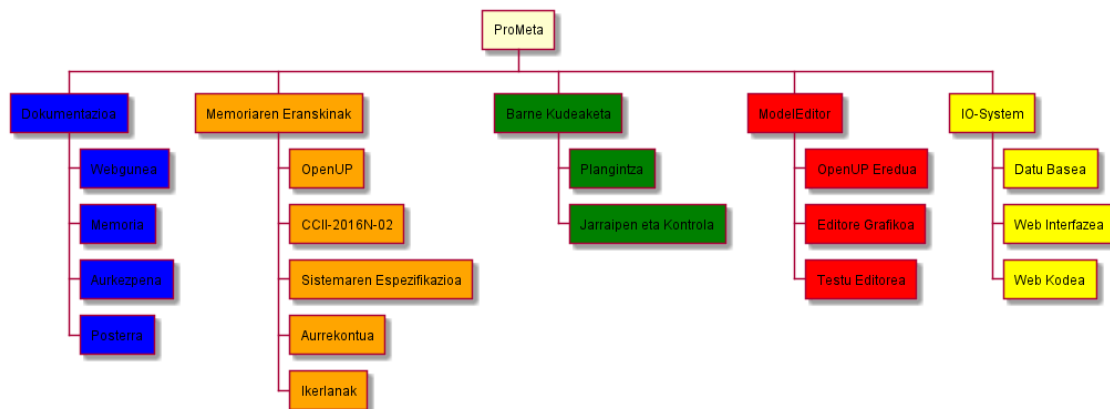
Memoria	3	5	I1,I2,I3,I4,I5			UPV/EHU, CCII-2016N-02
Memoriaren Eranskinak - OpenUp	1	3	I1,I2,I3,I4,I5			OpenUP
Memoriaren Eranskinak - CCII-2016N-02	1	3	I1,I2,I3,I4,I5			CCII-2016N-02
Sistemaren Espezifikazioa	1	2	I1,I2			OpenUP
Aurrekontua	2	1	I1,I2			CCII-2016N-02
Plangintza	3	2	I1			OpenUP
Aurreko 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	I3,I4			Drupal

14.2. Taula. Lan-atazen zehaztasunak

### 14.3 LDE diagrama

Aurreko taulak atazen zehaztasun guztiak biltzen ditu. Ideia orokor bat edukitzeko informazio gehiegi izan daiteke, diagramen bidez argiago ikusten da ataza banaketa. LDE diagramaren bitartez ataza eta azpiataza nagusiak modu argian adieraz daitezke. 14.1. Irudian ikus daiteke LDE diagrama.



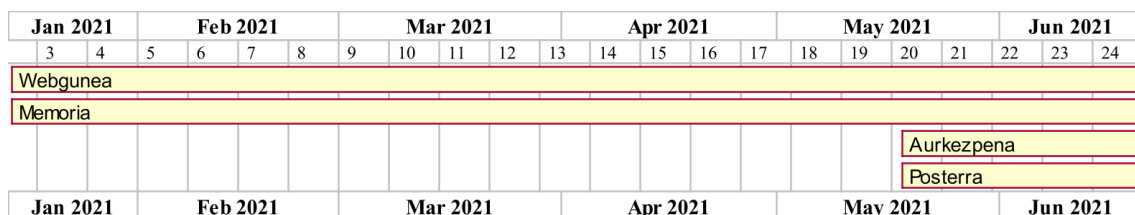


14.1. Irudia. LDE diagrama.

- Dokumentazioa
  - Webgunea
  - Posterra
  - Aurkezpena
  - Memoria
- Memoriaren Eranskinak
  - OpenUP
  - CCII-2016N-02
  - Sistemaren Espezifikazioa
  - Aurrekontua
  - Ikerlanak
- Barne Kudeaketa
  - Plangintza
  - Jarraipen eta Kontrola
- ModelEditor
  - OpenUP Eredua
  - Editore Grafikoa
  - Testu Editorea
- IO-System
  - Datu Basea
  - Web Interfazea
  - Web Kodea

## 14.4 Gantt diagrama

Gantt diagramaren bitartez atazen garapen denborak zehaztu daitezke. Aurreko atalean definitutako paketeak eta atazak kontuan hartuz, hurrengo irudian agertzen den iterazio bakoitzeko lanaren estimazioa egin da. 14.2. Irudian ikus daiteke Gantt diagrama.



14.2. Irudia. Gantt diagrama.

## 14.5 Iterazioak

Atal honetan, OpenUP metodologiaren bitartez jaso diren atazak iterazioetan zehar nolako banaketa izango duten deskribatuko da. Lan guztia 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 fasekoa. Helburuak lan-atazen antzekoak dira, baina informazio gehiagorekin.

Fasea	Iterazioa	Helburuak	Hasiera	Amaiera
Hasiera	I1	<p>Iterazio Plana.</p> <p>OpenUp Hasiera faseko artefaktuak.</p> <p>Aurreko proiektuen informazioa ulertu.</p> <p>OpenUp txantiloia eskuratu.</p> <p>Memoriaren txantiloia sortu (Fakultateko Arautegiak hemen erabiliko diren metodo eta arauak gomendatzen dutenekin ez dator beti bat. Konponbide batzuk eratuko dira.</p> <p>CCII arauaren informazioa jaso.</p> <p>CCII araua betetzen duen webgunea eraiki.</p> <p>OpenUp metodologiaren informazioa bildu (metaeredua, eredua...).</p> <p>Lan egiteko tresnak eskuratu (Eclipse, EPF Composer, EHSIS, Drupal, etab.).</p> <p>ProMeta proiektuak sortuko duen produktuak jarraituko duen garapen prozesuaren OpenUp eredua definitu.</p> <p>Metaereduan oinarritutako eredu editore grafiko eta testu editorearen prototipoa.</p>	2021/01/14	2021/02/14
Hasiera	I2	<p>Iterazio Plana.</p> <p>OpenUp Hasiera faseko artefaktuak.</p> <p>ProMeta proiektuak sortuko duen produktuak jarraituko duen garapen prozesuaren OpenUp eredua definitu.</p> <p>Metaereduan oinarritutako eredu editore grafiko eta testu editorearen prototipoa.</p> <p>Editore grafiko eta testu editorearen arteko sinkronizazioa. Eredua bera bi modutan editatzeko aukera.</p>	2021/02/15	2021/03/15

		Drupal webguneak erabiliko duen datu-basea sortu eredutik.		
Elaborazioa	13	Iterazio Plana.  Drupal webguneak erabiliko duen datu-basea sortu eredutik.  OpenUp Elaborazio faseko artefaktuak.  Web interfazea eraiki (Drupal).  Web kodea garatu (Drupal).  Webgunea.  Memoria.	2021/03/16	2021/04/16
Elaborazioa	14	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.

## 14.6 Neurtutako denborak

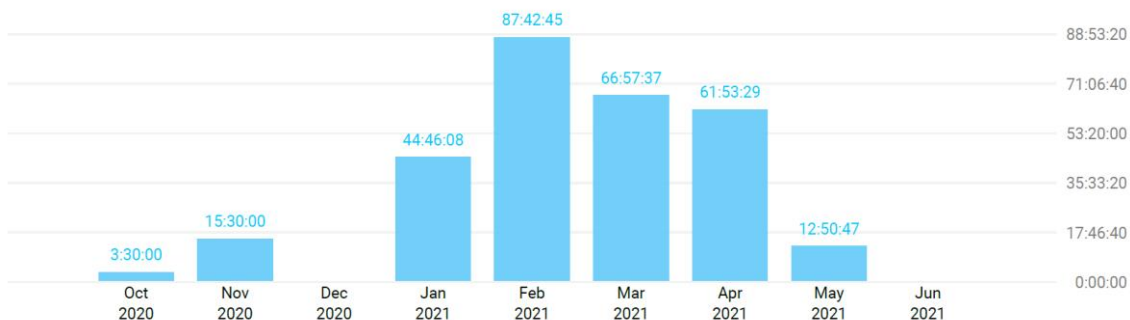
Toggle Track aplikazioa erabiliz neurtu dira proiektuko denborak. Ataza bakoitzaren denbora neurtu denez, ataza bakoitzeko denbora jakiteko ez daukagu kalkulurik egin beharrik, aplikazioak zuzenean esaten digu. Gainera, nahi dugun denborak bistaratu ditzakegu, adibidez hilabete bakoitzeko denbora edo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

### 14.6.1 Denbora hilabeteka

Guztira hilabete bakoitzean pasatako denbora ikus dezakegun hurrengo irudian.

01-10-2020 – 30-06-2021

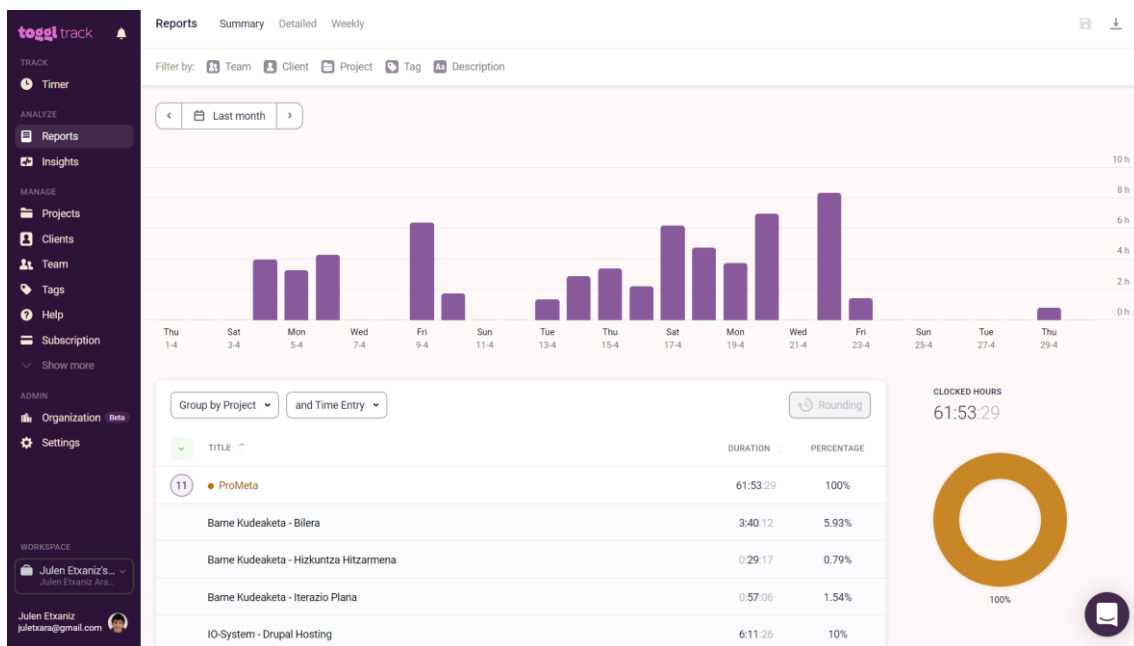
TOTAL HOURS: 293:10:46



14.3. Irudia. Proiektuaren denbora hilabeteka.

## 14.6.2 Hilabete bateko denbora

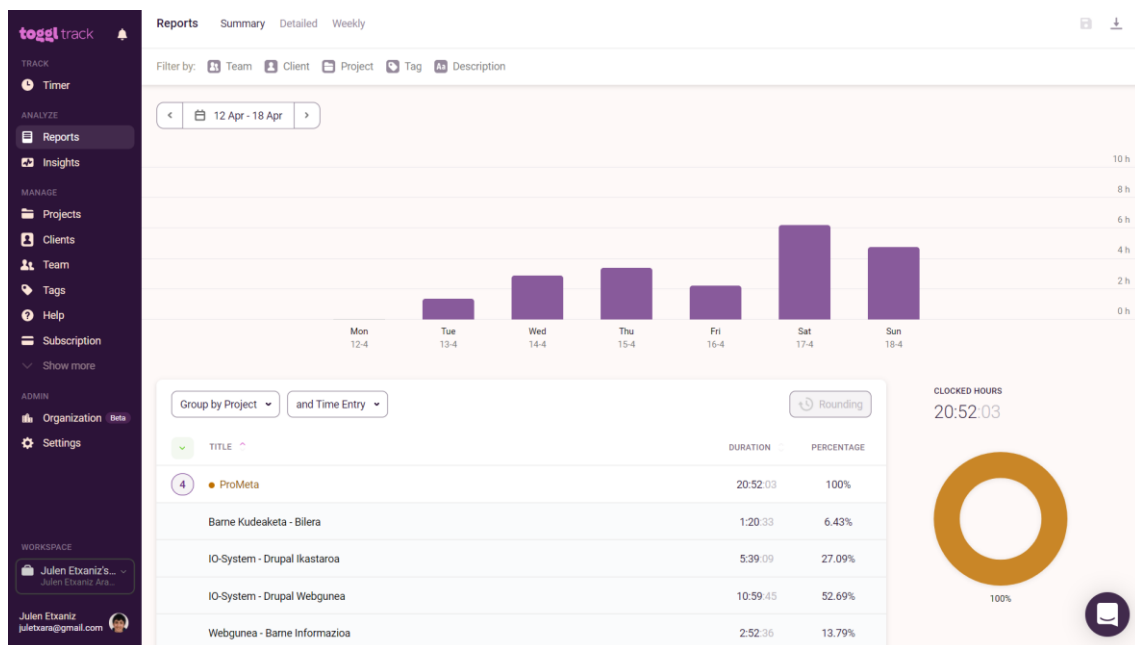
Hilabete bateko egun bakoitzeko denbora ikus dezakegu grafika batean. Gainera, azpian ataza bakoitzari eskainitako denbora eta denbora totala ikus ditzakegu.



14.4. Irudia. Hilabete bateko denbora.

## 14.6.3 Aste bateko denbora

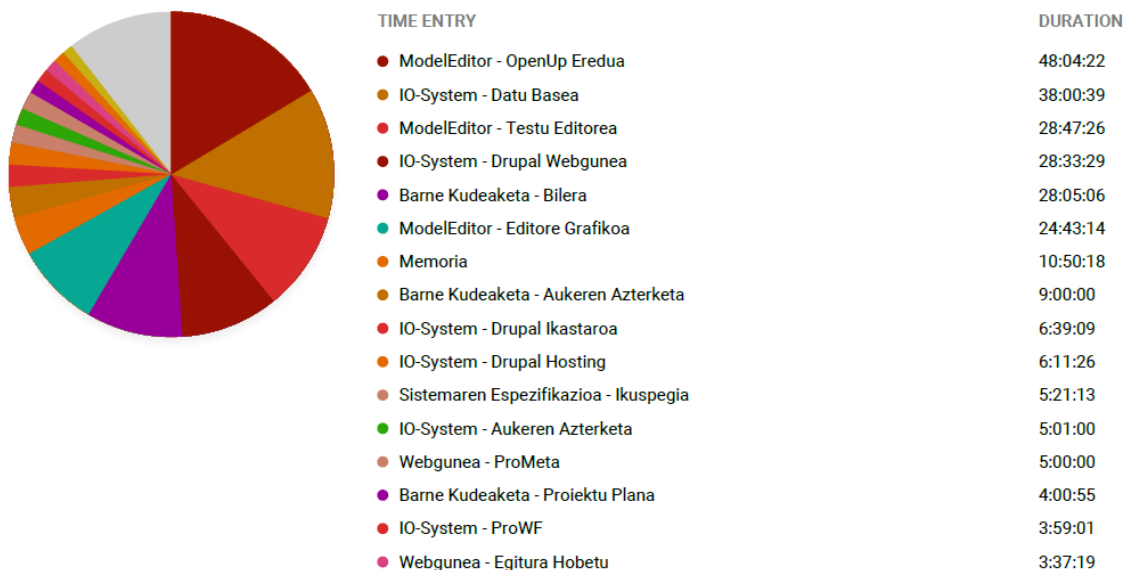
Aste bateko egun bakoitzeko denbora ikus dezakegu grafika batean. Gainera, azpian ataza bakoitzari eskainitako denbora eta denbora totala ikus ditzakegu.



14.5. Irudia. Aste bateko denbora.

## 14.6.4 Denbora atazaka

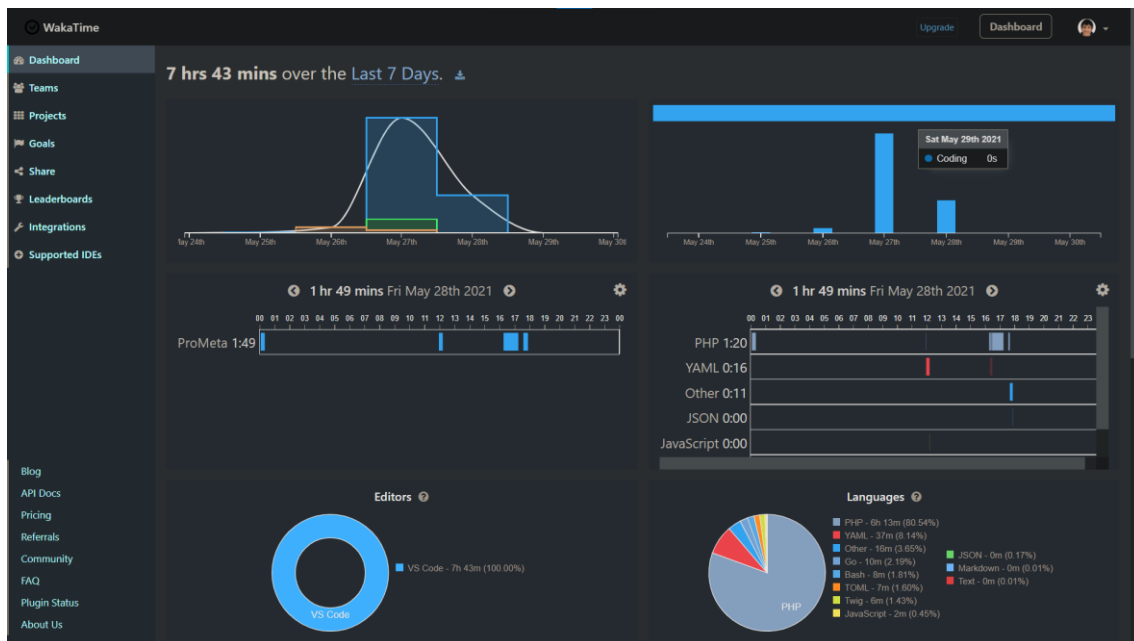
Ataza garrantzitsuenei guztira eskainitako denbora ikus dezakegu hurrengo irudian. Denbora gutxi eskaini zaien atazak grisez multzokatuta agertzen dira. Ataza horien denborak kontsulta daitezke arazorik gabe.



14.6. Irudia. Proiektuaren denbora atazaka.

## 14.6.5 WakaTime

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



## 14.7 Desbiderapenak

## 15 Aurrekontuaren Laburpena

Aurrekontua sortzeko *ALI (Asociación de Titulados Universitarios Oficiales en Informática)* elkarteak 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, proiektua garatzeko behar izan diren erreminten kostua kalkulatzea. Testing teknikoen eta auditoretza baten ziurtagiriaren kostua alde batera utzi da.

Giza baliabideen kostua Ekonomia eta Ogasun Ministerioaren 26/2010 Esparru Akordioan oinarrituta dago. Bezeroari begira software proiektu baten kide bakoitzarentzako erabiltzen dugun baremoa honakoa izango da:

- Proiektuko zuzendaria: 100€/ordua
- Arkitektoa: 70€/ordua
- Analista: 70€/ordua
- Garatzaileak: 50€/ordua
- Testerrak: 50€/ordua

Erabilitako erreminta guztiak doakoak izan dira, beraz, arkitektura propioa eraikitzearen erabakia egokia izan da. Bizagiren arkitektura erabiliz bere lizentziak eta urteroko mantenuak proiektuaren kostua handituko lukete. Beste alde batetik, ez dira aurkitu erabilitako erreminten premium lizentziarik, baina egotekotan aurrekontuan sartu daitezke, sistemaren kalitatea handitzeko asmoz. 15.1. 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	Testera	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

15.1. 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 kontsultatu 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](mailto:juletxara@gmail.com). Nahiago bada, nire webgunean kontakturako aukera gehiago daude: <https://julenetxaniz.eus/#contact>.



## 17 Memoriaren Eranskinak

### 17.1 Sarrerako Dokumentazioa

### 17.2 Analisia eta Diseinua

#### 17.2.1 Arkitektura Kuadernoa

#### 17.2.2 Analisiaren Eredua

#### 17.2.3 Diseinuaren Eredua

### 17.3 Tamaina eta Esfortzu Estimazioa

### 17.4 Kudeaketa Plana

#### 17.4.1 Integrazioaren Kudeaketa

#### 17.4.2 Irismenaren Kudeaketa

#### 17.4.3 Epeen Kudeaketa

#### 17.4.4 Produktuaren Kostuen Kudeaketa

#### 17.4.5 Kalitate Kudeaketa

#### 17.4.6 Giza Baliabideen Kudeaketa

#### 17.4.7 Komunikazioen Kudeaketa

#### 17.4.8 Arriskuen Kudeaketa

#### 17.4.9 Erosketen Kudeaketa

#### 17.4.10 Interesatuen Kudeaketa

### 17.5 Segurtasun Plana

### 17.6 Beste Eranskinak

#### 17.6.1 Hedapena

#### 17.6.2 Garapena

#### 17.6.3 Ingurunea

#### 17.6.4 Proba

## 18 Sistemaren Espezifikazioa

### 18.1 Glosategia

### 18.2 Ikuspegia

### 18.3 Betebeharren Espezifikazioa

### 18.4 Erabilpen Kasuak

### 18.5 Erabilpen Kasuen Eredua

## 19 Aurrekontua

## 20 Ikerlanak