

## Informatika Fakultatea

Informatika Ingeniaritzako Gradua

### ▪ Gradu Amaierako Lana ▪

Software Ingeniaritza



### **ProWF**

Software proiektuen elaboraziorako workflowetan  
oinarritutako sistemaren sorkuntza eta bizi-zikloa  
definitzeko metodologia baten ezarpena

---

Julen Rojo Raño

Tutorea: Juan Manuel Pikatza Atxa

2020ko iraila



## Esker Onak

---

Eskerrik asko Angelari, bere Gradu Amaierako Lana nire ondoan egin eta sufrimendua elkarrekin partekatzeagatik.

Juanmari, lan honen tutoreari, laguntzeko beti prest egoteagatik eta bere adiskidetasunarengatik. Ez ditut inoiz ahaztuko berak emandako aholku guztiak.

Nire aita-ama eta anaiari, konfinamenduko hilabete gogorretan nirekin egoteagatik.

Nire lagunei, proiektua dela eta etxean geratu naizen egun guztiak jasateagatik.

Nire neska-lagunari, bidaia hau atseginagoa egiteagatik eta lana ulertzeko jarri duen ahalegin guztiarengatik ;)

Eta amaitzeko, nire amona Martinari, lanaren hasieran bakarrik egon da, baina bere sostenguak proiektu guztian zehar jarraitu du. Goian bego.

# Laburpena

---

Sektore eta domeinu ezberdinetan jarduera jakin batzuk modu automatizatuan egiteko ezagutza erabilgarria metatzen da, kontsentsu handiko gidak jarraituz. Adibidez, proiektu informatiko baten elaborazio-prozesua egiteko. Horretarako, beharrezkoa da gida horiek definitzea eta kudeatzea, bere etorkizunerako mantenu eta hobekuntzak kontuan hartuz. Hori dela eta, gida horien edukia adierazteko lengoaia bat eta lengoaia horren bitartez sortutako ereduak egikaritzeko dituen sistema definitu behar da.

Lan honen kasuan gidak horiek *workflow*<sup>1</sup> izango dira eta sortuko den *workflow*-lengoiaren bitartez softwarearen bizi-zikloa<sup>2</sup> definitzen duen metodologia bat ezarriko da.

**Gako-hitzak:** workflow, softwarearen bizi-zikloa, metodologia, araua, sistematizazioa, automatizazioa, betekizunen ingeniarietza.

---

<sup>1</sup> *Workflow*, lan-fluxu: Aspektu operazionaleraino 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.

<sup>2</sup> Softwarearen bizi-zikloa: software-produktu baten garapenari aplikatutako egitura da. Softwarea garatzeko prozesu bat ezartzeko hainbat eredu daude, eta horietako bakoitzak ikuspegi ezberdin bat deskribatzen du prozesuan zehar egiten diren jarduera ezberdinetarako.

# Aurkibide Orokorra

---

Esker Onak .....	iii
Laburpena .....	iv
Aurkibide Orokorra.....	v
Irudien Aurkibidea .....	ix
Taulen Aurkibidea.....	xi
Laburdurak.....	xii
1. Sarrera .....	1
1.1 Testuingurua.....	1
1.2 <i>Software</i> proiektuen aurkezpena eta garapena.....	2
1.3 Proiektuaren webgunea.....	2
1.4 Arauak, erreferentziak eta definizioak .....	4
2. Proiektuaren Xedea .....	5
3. Aurrekariak.....	7
3.1 Bezeroen eskakizun gogorrak.....	7
3.2 Proiektuak aurkezteko arauak.....	8
3.3 Metodologiaren beharra .....	9
3.4 <i>BPM</i> : abantailak eta menpekotasunak .....	10
3.5 <i>Inferentzia motorrak</i> : CLIPSetik EHSISera .....	13
4. Egungo Egoera .....	15
4.1 Proiektuaren abiapuntua.....	15
4.2 Prestakuntza .....	15
5. Arauak eta Erreferentziak.....	17
5.1 Xedapen legalak eta aplikatutako araudia .....	17
5.2 Bibliografia.....	18
5.3 Metodoak, tresnak, ereduak, metrikak eta prototipoak.....	19
5.3.1 Metodoak.....	19
5.3.2 Tresnak.....	19
5.3.3 Ereduak .....	23
5.3.4 Metrikak .....	24
5.3.5 Prototipoak .....	24



5.4 Idazkerako kalitatearen kudeaketa plana .....	24
5.5 Beste erreferentzia batzuk .....	24
6. Definizioak eta Laburdurak.....	25
6.1 Termino eta kontzeptuen definizioak .....	25
7. Hasierako Betekizunak.....	27
7.1 Betekizun funtzionalak.....	27
7.2 Betekizun ez-funtzionalak.....	28
8. Irismena.....	29
8.1 Proiektuaren irismena.....	29
8.2 Proiektuak sortutako entregagarriak .....	30
9. Hipotesiak eta Murriztapenak.....	31
9.1 Hipotesiak .....	31
9.2 Murriztapenak .....	31
10. Aukeren Ikerketa eta Egingarritasuna .....	33
10.1 Arkitekturaren erabakia.....	33
10.2 Bizi-zikloa definitzen duen metodologiaren aukeraketa.....	35
10.3 <i>Workflow</i> -lengoaia bilatu .....	36
10.4 <i>CMSak</i> probatu.....	38
10.5 Inferentzia-motorraren erabakia .....	39
11. Proposatutako Sistemaren Deskribapena .....	41
11.1 Sistemaren testuingurua .....	41
11.2 <i>ProWF: Workflow Editor</i> .....	43
11.2.1 Arkitektura .....	43
11.2.2 Erabilitako teknologiak.....	43
11.2.3 Analisia .....	44
11.2.3.1 Workflowaren irudia sortu.....	44
11.2.3.1 Workflowaren informazioa datu-base erlazionalean gorde .....	45
11.2.4 Diseinua.....	45
11.2.5 Implementazioa .....	46
11.3 <i>ProWF: IO-System</i> .....	49
11.3.1 Arkitektura .....	49
11.3.2 Erabilitako teknologiak.....	50
11.3.3 Analisia .....	50



11.3.3.1 Proiektua sortu .....	51
11.3.3.2 Workflowa exekutatu.....	52
11.3.3.3 Workflowaren ekintzak egikaritu .....	52
11.3.3.4 Artefaktuak baloratu .....	53
11.3.3.5 Proiektuaren webgunea ikusi .....	53
11.3.4 Diseinua.....	54
11.3.5 Implementazioa .....	56
11.3.5.1 Drupal 8 eta eduki modularra.....	56
11.3.5.2 Workflowaren exekuzioa eta EHSIS_RT inferentzia motorra.....	56
11.4 Sistemaren abantailak eta desabantailak .....	56
11.4.1 Abantailak .....	57
11.4.2 Desabantailak .....	57
11.5 Etorkizunerako hobekuntzak.....	58
11.6 Ondorioak.....	59
12. Proposatutako Sistemaren Arrisku Analisia.....	61
12.1 Arriskuen identifikazioa .....	61
12.2 Arriskuen analisi kuantitatibo eta kualitatibo.....	61
12.3 Arriskuak arintzeko estrategiak.....	62
13. Proposatutako Sistemaren Antolaketa eta Kudeaketa .....	65
13.1 Proiektuaren antolaketa .....	65
13.1.1 Taldekideak .....	65
13.1.2 Ardurak eta betekizunak .....	65
13.1.3 Lan-ingurunea.....	66
13.1.4 Informazio-sistema.....	66
13.1.5 Komunikazio-kanalak.....	66
13.2 Proiektuaren kudeaketa .....	66
14. Proposatutako Sistemaren Denboraren Planifikazioa .....	67
14.1 Mugarri garrantzitsuak .....	67
14.2 Lan-ataza nagusien diagrama (LDE) .....	67
14.3 Lan-atazen denbora estimazioa .....	69
14.4 Iterazio-prozesua .....	70
15. Proposatutako Sistemaren Aurrekontua.....	73
15.1 Orokortasunak.....	73

16. Oinarrizko Dokumentuen Ordena.....	75
ERANSKINAK .....	77
I. Memoriaren Eranskinak .....	77
A1: Sarrerako dokumentazioa .....	77
A2: Analisi eta Diseinua .....	77
A3: Tamaina eta esfortzu estimazioak.....	77
A4: Proiektuaren kudeaketa plana.....	77
A5: Segurtasun plana.....	78
A6: Gainerakoak.....	78
II. Sistemaren Espezifikazioa .....	78
III. Aurrekontua.....	78
IV. Proiektuaren Barne Kudeaketa .....	78
IV.1. Arriskuen Analisia .....	79
IV.1.1 Arriskuen identifikazioa .....	79
IV.1.2 Arriskuen analisi kuantitatibo eta kualitatiboa.....	80
IV.1.3 Arriskuak arintzeko estrategiak .....	81
IV.2. Proiektuaren Antolaketa eta Kudeaketa .....	83
IV.2.1 Proiektuaren antolaketa .....	83
IV.2.1.1 Taldekideak .....	83
IV.2.1.2 Ardurak eta betekizunak .....	83
IV.2.1.3 Lan-ingurunea.....	83
IV.2.1.4 Informazio-sistema.....	83
IV.2.1.5 Komunikazio-kanalak.....	83
IV.2.2 Proiektuaren kudeaketa.....	84
IV.3. Denboraren Planifikazioa .....	85
IV.3.1 Mugarri garrantzitsuak .....	85
IV.3.2 Lan-ataza nagusien diagrama (LDE) .....	85
IV.3.3 Lan-atazen denbora estimazioa .....	87
IV.3.4 Iterazio-prozesua.....	88
IV.3.5 Desbiderapenak .....	90
IV.4. Aurrekontuaren Laburpena.....	93
IV.4.1 Orokortasunak.....	93



# Irudien Aurkibidea

---

1.1 irudia: ProWF proiektuaren webgunea .....	2
1.2 irudia: Proiektuaren webgunearen ezkerreko menua .....	3
3.1 irudia: CCII-2016N-02 araua betetzen duen proiektuaren webgune baten antolamendua. ...	9
3.2 irudia: RUPen prozesu iteratiboa .....	10
3.3 irudia: OpenUP metodologiaren bizi-zikloa eta iterazioak .....	10
3.4 irudia: 2018ko BPM Suiten Koadrante Magikoa, Gartner .....	12
3.5 irudia: 2019ko The Forrester Wave txostena, Forrester .....	12
3.6 irudia: EHSISen garapen ingurunea .....	13
3.7 irudia: Programazio tradizionalaren eskema.....	14
5.1 irudia: Graphviz tresnaren bitartez sortutako workflow-eredua .....	20
5.2 irudia: Gephi softwarearen interfaze grafikoa .....	20
5.3 irudia: Protégé softwarearen interfaze grafikoa .....	21
5.4 irudia: EHSISen garapen ingurunea .....	21
5.5 irudia: XAMPP programaren interfaze grafikoa .....	22
5.6 irudia: PlantUML softwarearen bitartez sortutako diagrama .....	23
8.1 irudia: OpenUP metodologiako bizi-zikloaren faseak .....	29
10.1 irudia: Bizagi Modeler erabiliz sortutako prozesua .....	34
10.2 irudia: Bizagiren arkitektura .....	35
10.3 irudia: Workflowetan oinarritutako arkitektura .....	35
10.4 irudia: Workflow-lengoiaren bitartez sortutako irudia.....	37
10.5 irudia: Datu-basean gorde beharreko kode numerikoak workflowan agertzen dira.....	37
10.6 irudia: Negozio-logika ez da lan-fluxuan agertzen.....	38
10.7 irudia: Domeinuari (OpenUP) buruzko ezagutza objektu bezala jarrita. ....	40
11.1 irudia: ProWF sistemak behar dituen osagaiak eta haien arteko erlazioak.....	42
11.2 irudia: Workflow Editor azpisistemaren arkitektura.....	43
11.3 irudia: Workflow Editor azpisistemaren erabilpen kasuen diagrama. PlantUML. ....	44
11.4 irudia: Datu-base erlazionalaren diseinua (I) .....	46
11.5 irudia: Graphml formatuak definitutako klaseak .....	48
11.6 irudia: IO-System moduluaren arkitektura.....	49
11.7 irudia: IO-System azpisistemaren erabilpen-kasuen diagrama. PlantUML. ....	51

11.8 irudia: Datu-base erlazionalaren diseinua (II) .....	54
11.9 irudia: Datu-base dokumentalaren diseinua.....	55
12.1 irudia: arriskuen larritasuna adierazten duen grafikoa .....	62
14.1 irudia: Lan-ataza nagusien diagrama (WBS/LDE).....	68
14.2 irudia: hilabete bakoitzean egin beharreko lanaren estimazioa .....	69

#### Eranskinak

IV.1.1 irudia: arriskuen larritasuna adierazten duen grafikoa .....	80
IV.3.1 irudia: Lan-ataza nagusien diagrama (WBS/LDE).....	85
IV.3.2 irudia: hilabete bakoitzean egin beharreko lanaren estimazioa.....	87

# Taulen Aurkibidea

---

3.1 taula: CMMIren heldutasun-mailak.....	8
8.1 taula: OpenUP metodologiako artefaktuak.....	30
10.1 taula: CMS ezberdinen konparazio-aula .....	39
12.1 taula: Arriskuen identifikazio-zerrenda .....	61
12.2 taula: Arriskuen identifikazio-zerrenda .....	62
12.3 taula: Arriskuak arintzeko estrategiak .....	63
14.1 taula: Proiektuko mugarri garrantzitsuak.....	67
14.2 taula: lan-ataza bakoitzari estimatutako ordu kopurua .....	70
14.3 taula: Proiektuko iterazioen banaketa eta bakoitzaren helburuak.....	71
15.1 taula: proiektuaren aurrekontua .....	74

## Eranskinak

IV.1.1 taula: Arriskuen identifikazio-zerrenda .....	79
IV.1.2 taula: Arriskuen identifikazio-zerrenda .....	80
IV.1.3 taula: Arriskuak arintzeko estrategiak.....	81
IV.3.1 taula: Proiektuko mugarri garrantzitsuak.....	85
IV.3.2 taula: lan-ataza bakoitzari estimatutako ordu kopurua .....	88
IV.3.3 taula: Proiektuko iterazioen banaketa eta bakoitzaren helburuak .....	89
IV.3.4 taula: ataza bakoitzari emandako ordu kopuru totala eta estimatutakoaren diferentzia .	90
IV.4.1 taula: proiektuaren aurrekontua .....	94

# Laburdurak

---

## Laburdura

## Esanahia

<b>BPM</b>	Business Process Management
<b>BPMN</b>	Business Process Model and Notation
<b>CCII</b>	Consejo de Colegios de Ingeniería Informática
<b>CLIPS</b>	C Language Integration Production System
<b>CMMi</b>	Capability Maturity Model Integration
<b>CMS</b>	Content Management System
<b>CSS</b>	Cascading Style Sheets
<b>GrAL</b>	Gradu Amaierako Lana
<b>HTML</b>	HyperText Markup Language
<b>IBM</b>	International Business Machines
<b>JMP</b>	Juan Manuel Pikatza
<b>JRR</b>	Julen Rojo Raño
<b>OpenUP</b>	Open Unified Process
<b>PHP</b>	Hypertext Pre-Processor
<b>RUP</b>	Rational Unified Process
<b>SPICE</b>	Software Process Improvement and Capability Determination
<b>SWEBOK</b>	Software Engineering Body of Knowledge
<b>UML</b>	Unified Modeling Language
<b>UNE</b>	Una Norma Española
<b>WF</b>	Workflow

# 1. Sarrera

---

Dokumentu hau **Julen Rojo Raño**, Informatika Ingeniaritzako Graduoko ikaslearen **Gradu Amaierako Lanaren** memoria da, bere tutorea **Juan Manuel Pikatza** izanik eta Euskal Herriko Unibertsitateko (UPV-EHU) **Donostiako Informatika Fakultatean** landutakoa.

Dokumentu honetan aurkitzen da **ProWF**<sup>3</sup> proiektua gauzatzeko beharrezkoak izan diren pausu eta aspektu guztiak, Informatika Fakultateak emandako GrAL eredua eta bere arauak jarraituz.

Gradu Amaierako Lan honen titulu ofiziala hurrengoa hau da: **“Software proiektuen elaboraziorako workflowetan oinarritutako sistemaren sorkuntza eta bizi-zikloa definitzeko metodologia baten ezarpena”**.

## 1.1 Testuingurua

*Softwarearen* kalitatea *softwarearen* ezaugarriak kontrolatzean eta ziurtatzean datza, bezero eta erabiltzaileen onurarako. *Softwarea* produktu immateriala da, ez da fabrikatzen, 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 daitezke baina, *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* eraikitzeke 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>4</sup> eta *SPICE*<sup>5</sup> dira, helburu berdina dutenak. Kalitate-ereduek, *softwarea* garatzeko praktika onenak definitzen dituzte, *softwarea* garatzen duten erakundeek prozesuak hobetzen laguntzeko. Ziurtagiri-emaileek, praktika on horiek hartu eta beraiei emaitza neurgarriak egiaztatzen dituzte garatzaileekin lankidetzan. Ebaluazio-prozesu batean parte hartu ahal izateko inplikatu guztiak ziurtagiri egokiaren jabe izan behar dira.

---

<sup>3</sup> ProWF: 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.

<sup>4</sup> *CMMI*: Sigla(ingelesez), Capability Maturity Model Integration. Software-sistemak garatzeko, mantentzeko eta erabiltzeko, prozesuak hobetzeko eta ebaluatzeko eredua da, *CMMI* Institutuak administratutakoa.

<sup>5</sup> *SPICE*: Akronimoa(ingelesez), *Software Process Improvement and Capability dEtermination*. ISO/IEC 15504. Garapen-prozesuak hobetzeko, ebaluatzeko, informazio-sistemak eta software-produktuak mantentzeko eredua da.

## 1.2 Software proiektuen aurkezpena eta garapena

Gratu Amaierako Lan honek software proiektu profesional baten itxura izateko eta dokumentazioa formala sortzeko CCII-2016N-02<sup>6</sup> araua erabili da. Gainera, proiektua osatzeko estandar hori betetzen duen webgune bat sortu da, arauak esaten duen dokumentuen antolamendua jarraituz.

Beste alde batetik, softwarearen bizi-zikloa ezartzen duen *OpenUP*<sup>7</sup> metodologia jarraitu da proiektuaren elaborazioan zehar eta metodologia horren bitartez sortutako artefaktu guztiak proiektuaren webgunean jarri eta lehen aipatutako CCII-2016N-02 araua errespetatuz antolatu dira.

Beraz, esan daiteke proiektu honek ongi betetzen dituela software ingeniariatzako proiektu batek beharrezkoak dituen betekizunak.

## 1.3 Proiektuaren webgunea


Aurreko atalean komentatutako webgunea honakoa da:

<https://prowfgral.000webhostapp.com/index.htm>

Webgune honen helburua ProWF proiektuaren dokumentazio guztia biltzea eta proiektuko bezero zein interesdunek eskura izatea da.

Ezkerreko menua erabiliz, proiektuko edozein dokumentu ikustea lortu daiteke: memoria, eranskinak, posterra, barne kudeaketarako dokumentuak etab.

Hasierako orrian ikusten den moduan (1.1 irudia), webgunea bi segmentutan dago banatuta. Ezkerrean nabigazio menua agertzen da, 1.2 irudian guztiz desplegatuta ikus daiteke. Eskumaldean, ordea, nabigazio menuan aukeratutakoa agertuko da.




**ProWF**

- ProWF Proiektua
  - Portada
  - Laburpen posterra
  - Aurkezpena
  - ProWF sistema (.RAR)
  - Aurkibide orokorra
  - MEMORIA
  - ERANSKINAK
  - ProWF: Barne Kudeaketa
  - Proiektuaren Barne Kudeaketa

**ProWF**

<b>Bezeroa:</b> Izen soziala: Universidad del País Vasco / Euskal Herriko Unibertsitatea I.F.K.: 48940 Helbidea: Barrio Sarriena s/n, Leioa, Bizkaia Telefonoa eta Faxa: 94 601 20 00	<b>Suministrador:</b> Izen soziala: ProWF I.F.Z.: K79003710 Helbidea: Donostia, Aizkorni Kalea 12º Telefonoa eta faxa: 652 75 10 73
<b>Datak eta sinadurak:</b> 04/02/2020 - 09/04/2020 <b>Laburpena:</b> Workflowetan oinarritutako sistemak software proiektuen elaborazioan. <b>Aplikazioa:</b> Bizi-zikloa definitzeko metodologia baten ezarpena. <b>Estimaturako iraupena:</b> 5 hilabete <b>Kostua:</b> 28.024	<b>Egilea:</b> Izena eta abizenak: Julen Rojo Raño Titulazioa: Software Ingeniaritza I.F.Z.: 79003710K Helbide profesionala: Bilbao, Garate Kalea 21º Telefonoa: 652 75 10 73 e-posta: julenrojo12@gmail.com

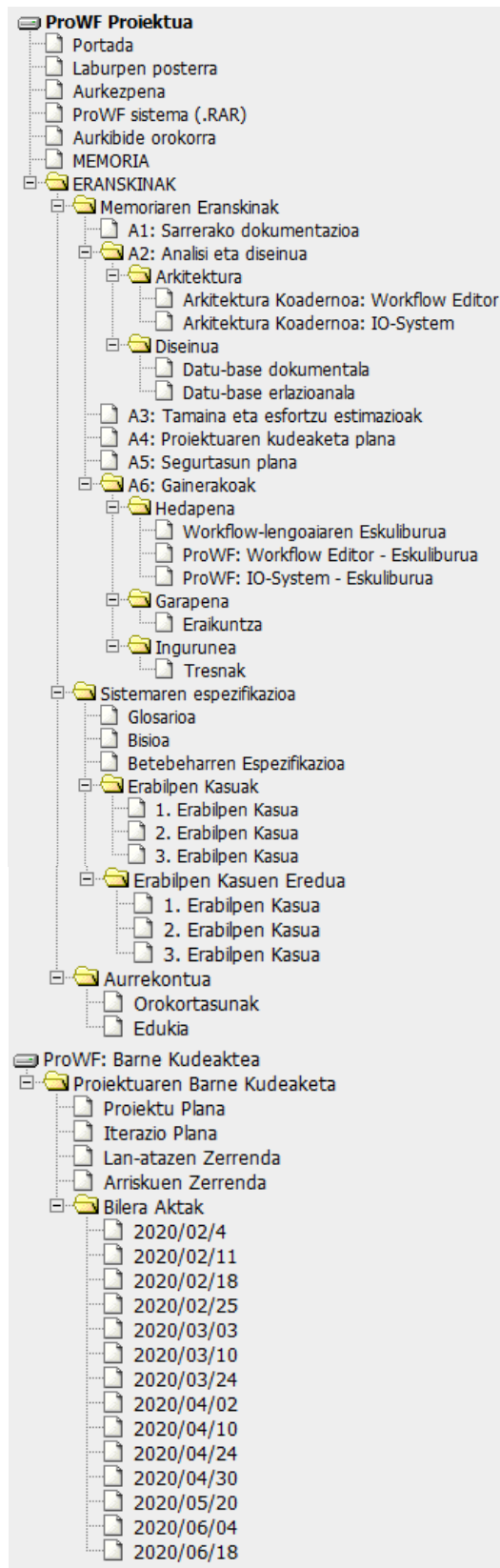
Gora 

© Copyright Juan M. Pikatza, Lengoaia eta Sistema Informatiko Saila (UPV/EHU). 2006, 2008. Eskubide guztiak erreserbatuta.

1.1 irudia: ProWF proiektuaren webgunea

<sup>6</sup> CCII: Siglak(gazteleraz), *Consejo de Colegios de Ingeniería Informática*. Estatu-mailan informatika ingeniari guztiak ordezkatzeko eta bateratzen dituen erakunde ofiziala da.

<sup>7</sup> OpenUP: Siglak(ingelesez), *Open Unified Process*. RUP(Rational Unified Process) metodologiaren azpimultzo bat da. Proiektu informatiko batean kokatzeko eta informazioa antolatuta izateko balio du.



1.2 irudia: Proiektuaren webgunearen ezkerreko menua

## 1.4 Arauak, erreferentziak eta definizioak

Proiektu hau egiteko erabili izandako baliabideak, aztertutako bibliografia, aplikatutako estandarrak eta beste erreferentzia batzuk **17. orrialdean** aurkitzen dira, 6. kapitulan eta dokumentu honen termino guztiak ondo ulertzeko beharrezkoak diren definizioak **25. orrialdean** aurkitzen dira, 7. kapitulan.



## 2. Proiektuaren Xedea

---

Proiektuaren izenburu ofizialak dioen moduan:

*“Software proiektuen elaboraziorako workflowetan oinarritutako sistemaren sorkuntza eta bizi-zikloa definitzeko metodologia baten ezarpena”.*

Horiek izango dira bi helburu nagusiak. Lehenengoa, lengoia grafiko bat, *workflow*-lengoia eta lengoia horretan oinarritutako sistema kudeatzaile bat sortzean datza; bigarrena, ordea, definitutako *workflow*-lengoia hori baliatuz *softwarearen* bizi-zikloa definitzeko metodologia bat ezartzea du helburu.

Enpresa edo proiektu informatikoen garatzaileen ikuspegitik hurrengoak dira helburuak: alde batetik, ekoizpen-prozesu<sup>8</sup> sistematiko bat izatea, ezinbestekoa etengabeko hobekuntza gauzatu eta kalitatezko produktuak sortzeko; bestetik, ekoizpen-prozesu hori sistematizatzeko baliabideak metodologia, arau eta estandarretatik ateratzea; azkenik, ekoizpen-prozesuko artefaktu guztiak berrerabili ahal izateko azpiegitura teknologiko bat sortzea.

Informatika Ingeniaritza Graduko ikasle bezala, Software Ingeniaritzan espezializatuta, lan honen egilearen helburua proiektuaren motibazioarekin bat dator: *software* enpresa bat sortzeko ezinbestekoak diren aspektuak bereganatzea, adibidez, gaur egungo bezeroen eskakizun ez funtzionalak betetzea, *softwarearen* kalitateari dagozkionak. Beraz, hori lortzeko proiektu honetan ukitzen diren aspektu garrantzitsu guztiak baliagarriak izango dira.

Orokorrean, esan daiteke helburu nagusiak izenburu ofizialean daudenak direla, besteak, ordea, epe luzekoak edo helburu ez-zuzenak dira, *software* enpresa munduari eta lan honen egilearen etorkizun profesionalari buruzkoak.

---

<sup>8</sup> Ekoizpen-prozesu: prozesu teknologiko bat aplikatuz eta lehengaiak edota produktu erdilanduak erabiliz, produktuak eta zerbitzuak eskaini ahal izateko egindako eragiketen edo faseen multzoa da.



## 3. Aurrekariak

---

Kapitulu honetan, proiektua burutu ahal izateko garrantzia izan duten iraganeko elementu esanguratsuak jasotzen dira.

Hasteko, gaur egungo *software* munduko bezeroen eskakizun gogorrak aztertuko dira, hori argi ikusteko *OESIA NETWORKS, S.L.* enpresak 2016an jarritako erreklamazioa eta dagokion administrazioaren erantzuna ebidentzia legez hartuko da. Ondoren, bezeroen eskakizunengatik agertu diren beharrak azalduko dira: *software* proiektuak aurkezteko arauak eta metodologiak. Hori azaldu eta gero, *BPM*<sup>9</sup> oinarritutako sistemek eskaintzen dituzten abantailak eta sortzen dituzten menpekotasunak aztertuko dira. Amaitzeko, inferentzia motorrak eta haien ahalmena komentatuko da, *CLIPS* eta *EHSIS* inguruneak aztertuz.

### 3.1 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, dagozkien ziurtagiriak arauarekin bat datozen erakundeek emanak izan behar dutela ezarri zuen.

Ondoren, aurreko legearekin bateratuta, 2011ko azaroaren 14an Estatuko Aldizkari Ofizialean onartu zen legearen ostean, **BOE-A-2011-17887**, bezeroen eskakizunak sendo gogortu ziran, softwarearen kalitateari dagokionez. Lege horren **80. artikulua** kalitatea bermatzeko arauak betetzen direla egiaztatzea zuen helburu, horretarako erakunde independenteek emandako ziurtagiriak beharrezkoak ziran merkatuak exijitzen bazituen. Erakunde horiek kalitatea bermatzeko Europako arau jakin batzuei erreferentzia egin edo baliokideak izan behar ziran. 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 (ikus 3.1 taula).

*CMMI*ren 3. heldutasun-maila lortzea ezinbestekoa zan merkatuan mantentzeko, hori dela eta, hainbat enpresa kexatu ziran 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 lortzear zegoela erreklamatzeko merkatuan mantentzeko, tramitazioaren faltan zegoen. Hala ere, administrazioaren erantzuna errekurtsua baiestearen aurka egotea izan zen, lehen aipatutako 80. artikulua ez betetzeagatik. Arrazoia enpresak oraindik ez zuela exijitutako kalitate maila bermatzen zuen ziurtagirik edo ziurtagiriaren baliokiderik izan zen.

---

<sup>9</sup> *BPM*: Sigla(ingelesez), *Business Process Management*. Enpresei prozesuak automatikoki modelatzeko, inplementatzeko eta exekutatzeak aukera ematen dien software teknologia da.

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

3.1 taula: CMMIren heldutasun-mailak

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.

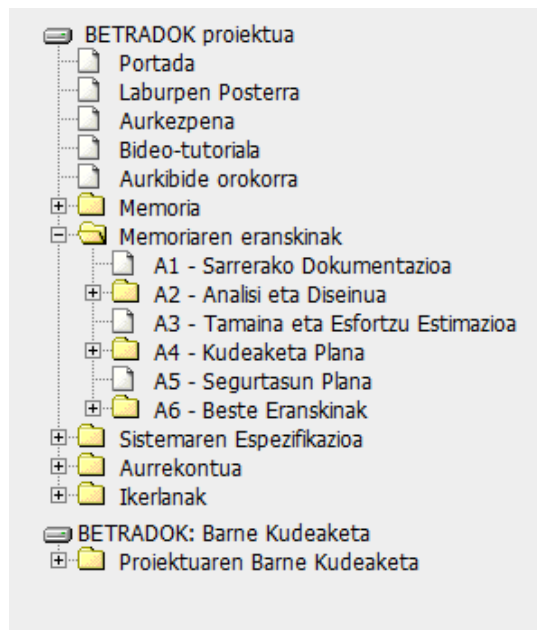
## 3.2 Proiektuak aurkezteko arauak

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.

Espanian eta nazioartean araurik finkatuenak *UNE*<sup>10</sup> eta *CCII* erakundeek aurkeztutakoak dira: **CCII-N2016-02**. Zehatz-mehatz deskribatzen du nola egin behar den Informatika Ingeniaritzako Proiektu baten dokumentu-zehaztapena (ikus 3.1 irudia). Proiektuko dokumentazioa egiteko ereduak eta dokumentazioaren antolamendua deskribatzen du ere. Erreferentziatzeko 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".

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.

<sup>10</sup> UNE: Sigla(gaztelaraz), *Una Norma Española*. Comités Técnicos de Normalización (CTN) batzordeak sortutako arauen, arau esperimentalen eta txostenen (estandarrek) multzoak dira.



3.1 irudia: CCII-2016N-02 araua betetzen duen proiektuaren webgune baten antolamendua. BETRADOK proiektua, Jon Legarda.

### 3.3 Metodologiaren beharra

Gaur egun ezinezkoa 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.

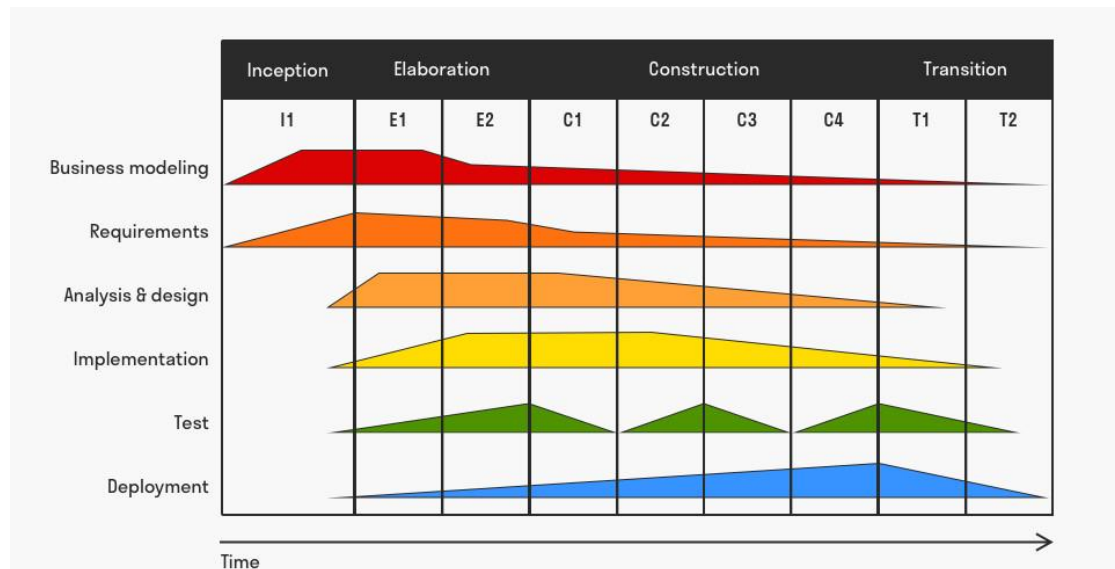
Gehien erabiltzen direnak metodologia arinak dira. “Arina” kontzeptua softwarea sortzeko urratsak arindu behar direlako sortzen da. Giza interakzioetan zentratzen da, aldean arteko elkarrikeria-fluxuari eutsi ahal izateko, garapen dinamikoagoa eta parte-hartzaileagoa ahalbidetzeko. Metodologia bizkorrek 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, ezin baititu ulertu bezeroaren beharrak.

*Software* garapenaren metodologia arinen artean RUP<sup>11</sup> aurkitzen da. *Softwarea* garatzeko prozesu iteratiboa bat da (ikus 3.2 irudia), *Rational Software Corporation* erakundeak sortua, *IBMren*<sup>12</sup> dibisio bat. RUP ez da zehatz-mehatz jarraitu behar den prozesua, baizik eta prozesu

<sup>11</sup> 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.

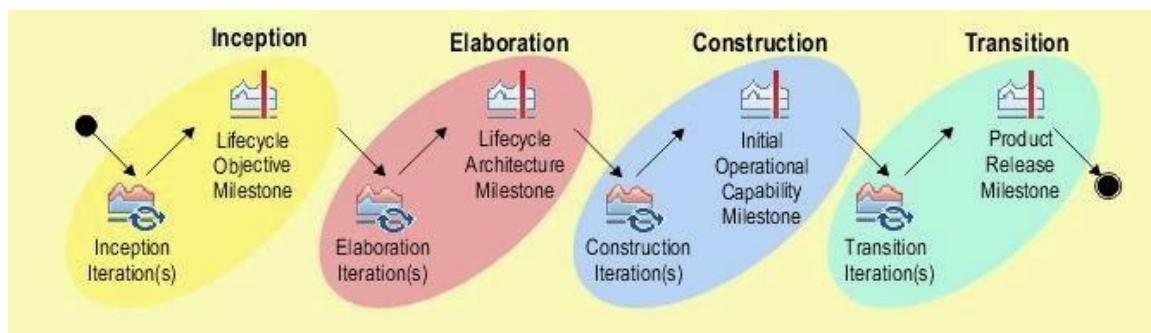
<sup>12</sup> IBM: Sigla(ingelesez), International Business Machines. Informatikarekin lotutako tresnak, programak eta zerbitzuak ekoiztu eta merkaturatzen dituen enpresa multinazionala da.

moldagarria da, garapen-erakundeek eta software-proiektuen taldeek egokitzeko asmoarekin, bakoitzaren premietarako egokiak diren elementuak hautatuz.



3.2 irudia: RUPen prozesu iteratiboa, goian bizi-zikloaren faseak agertzen dira, ezkerrean proiektuaren garapenean zehar egin beharreko jarduerak eta erdian iterazioen banaketa, jardura bakoitzaren lan-karga grafikoki adieraziz.

RUPen oinarritutako metodologia sinple eta erabiliena OpenUP da. Metodologia horrek RUPen funtsezko ezaugarriak gordetzen ditu, garapen iteratiboa (ikus 3.3 irudia), 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. Emaiza prozesu askoz sinpleagoa da, eta RUP printzipioekiko leiala izaten jarraitzen du.



3.3 irudia: OpenUP metodologiaren bizi-zikloa eta iterazioak

### 3.4 BPM: abantailak eta menpekotasunak

BPM 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. Lau koadrantetan banatutako grafiko gisa aurkezten da (ikus 3.4 irudia). 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*.

BPM merkatua aztertzeke, 2018 urteko BPM teknologien hornitzaile nagusien *Gartnerren* Koadrante Magikoa ikus dezakegu, 3.4 irudia. Antzeko informazioa aurkitu dezakegu *Forrester Wave* txostenean, 3.5 irudia. Bi txosten ezagun hauek, ondo kokatutako hornitzaileek komertzialki erabiltzen dituzte.



3.4 irudia: 2018ko BPM Suiten Koadrante Magikoa, Gartner <https://www.gartner.com>

### THE FORRESTER WAVE™

Software For Digital Process Automation For Deep Deployments

Q2 2019



3.5 irudia: 2019ko The Forrester Wave txostena, Forrester <https://www.forrester.com>



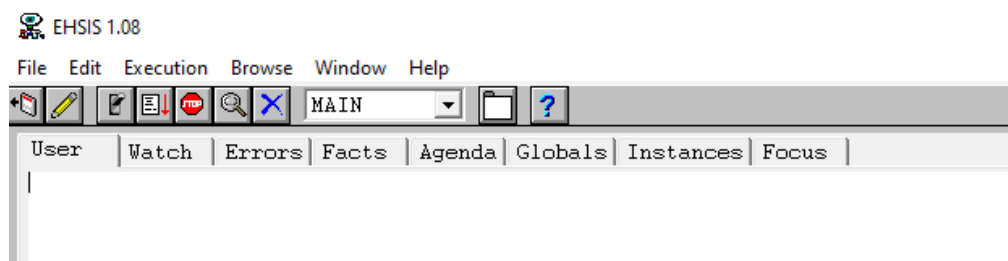
*Bizagi*<sup>13</sup> bi txostenetan agertzen da, ondo kokatuta. *Bizagi* bezalako teknologia erabiliz arkitektura konplexuko web-aplikazioa inplementatu daiteke, prozesu, erregela eta *workflowentzako* motorrarekin barne. Apustu hori garestia izan daiteke eta proiektu edo enpresaren menpekotasun teknologikoa handituko da. Beste apustu merkeago bat, metodologia eta estandarretara hurbiltzen joateko, teknologia propioa sortzea da, *workflow*-lengoaia propioa sortuz eta lengoaia horrekiko inferentzia motorra eraikiz.

### 3.5 Inferentzia motorrak: CLIPSetik EHSISera

*CLIPS* (*C Language Integration Production System*) sistema adituak ekoizteko eta exekutatzeko garapen-ingurunea ematen duen tresna da, *NASAK* 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 erabiltza *BPM* edo beste edozein sistemekin integragarria da.
- **Interakzio edo disziplinarreko garapena.** Formakuntza ezberdinetako pertsonen ideiak azkar inplementatzeko aukera lengoaia sinple eta ahaltsu bat erabiltza, 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 leihoetan 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 erabiltza, ezagutzen oinarritutakoak ere sor ditzake sistemaren arkitektura egoki batekin baliatuz (ikus 3.8 irudia).

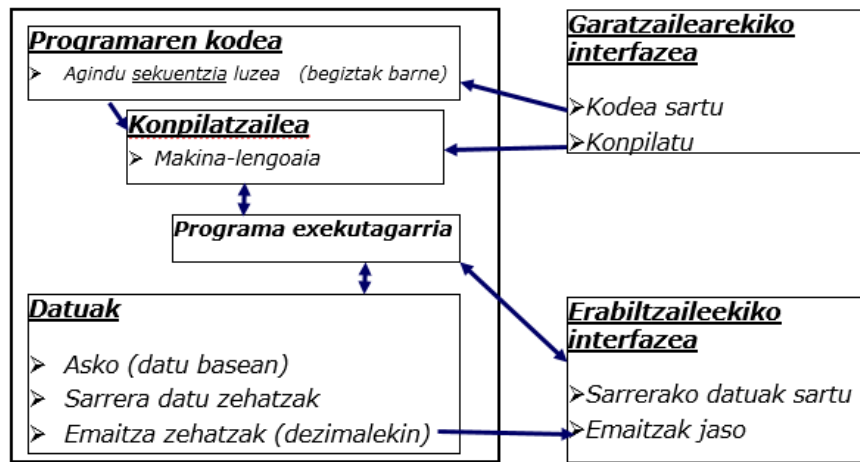


3.6 irudia: EHSISen garapen ingurunea

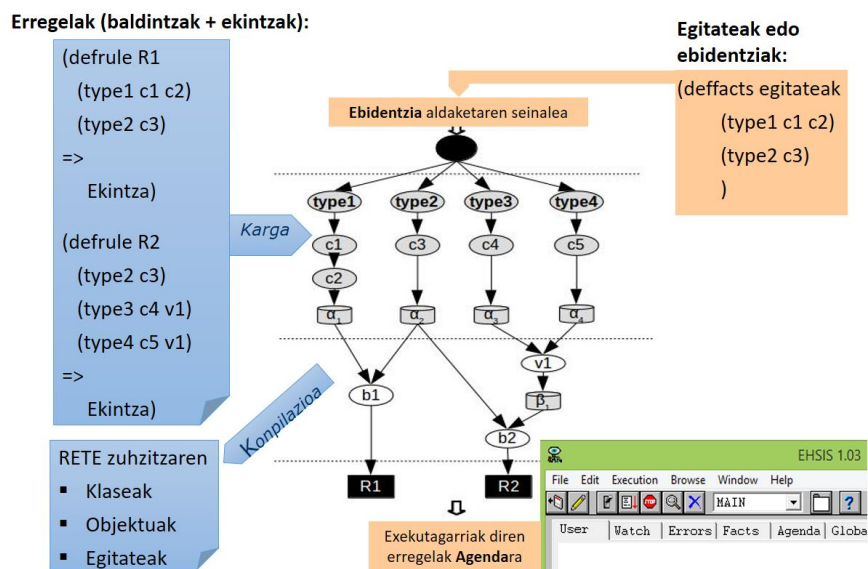
<sup>13</sup> *Bizagi*: Bi produktu osagarri dituen softwarea da, prozesuen modelatzaile bat eta *BPMren* suite bat.

<sup>14</sup> *FuzzyCLIPS*: Egitate eta erregela lausoen (*fuzzy*) errepresentazioa gehitzen dio *CLIPS*eri, The National Research Council of Canada taldeak sortutakoa.

<sup>15</sup> *COOL*: Sigla (ingelesez). *CLIPS Object Oriented Language*. *CLIPS*ek erabiltzen duen objektuetara bideratutako lengoaia.



3.7 irudia: Programazio tradizionalaren eskema



3.8 irudia: Erregelatan oinarritutako sistemen eskema: Klase, objektu eta erregelak kargatzearekin RETE zuhaitza sortzen da eta ebidentziak tratatzeko prest egongo da. RETE sareak gordeta karga azkartzen da.

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

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

## 4. Egungo Egoera

---

Kapitulu honetan, proiektuaren abiapuntua eta proiektua egin baino lehen jasotako prestakuntzak deskribatuko dira.

### 4.1 Proiektuaren abiapuntua

Jarraian, abiapuntua finkatzeko, proiektua egin baino lehen zeuden urritasunak deskribatzen dira:

- **Ibilbide metodologikoan zerotik hasi beharra.** *Software* proiektu baten elaborazioan metodologia baten jarraipenak ikasketa eta prestakuntza prozesu bat behar du. Prestakuntza prozesu hori zerotik hasi beharra denbora-kostu handia izaten da proiektuko partaide guztientzat, proiektuaren elaborazioan atzerapenak sortuz.
- Ibilbide metodologikoan aurrera egiteko, software prozesuaren euskarri izango den **azpiegitura teknologiko bat beharrezkoa da** eta bere arkitektura konplexua izanik, osagai garrantzitsuak integratu behar dira.
- *BPM* edo prozesuetan oinarritutako sistemak erabiltzen duten **azpiegitura teknologikoek menpekotasun handia eragiten dute**. *Software* ireki edo libreko osagaiak integratzea komenigarria izan daiteke.

Urritasun horiek direla eta proiektu hau aurrera eramatea eta *ProWF* sistema sortzea erabaki zen.

### 4.2 Prestakuntza

Proiektu honen egileak bazituen software proiektuen, *softwarearen* bizi-zikloaren, metodologia zein estandarren eta *BPMren* arloko oinarritzko ezagutzak, Informatika Ingeniaritzako Graduko Softwarearen Kalitatea irakasgaian ikasitakoak.

Izan ere, Softwarearen Kalitatea irakasgaian proiektu honetan interesa duten hurrengo ekintzak jorratu ziran:

- *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 ziran metodologiako artefaktu guztiak bete, baina bai hasierako fasekoak, betekizunen ingeniaritzari buruzkoak.
- Proiektu bat aurkezteko webgunea sortu eta antolatu *CCII-2016N-02* araua jarraituz.

Beste alde batetik, proiektua hasi baino lehen hurrengo gaietan prestakuntza jaso behar izan da:



- Ezagutzan oinarritutako sistemen garapen sistematizazioa. Garapen-prozesu baten ezagutza adierazi eta exekutatzeko.
- Lengoaia-grafiko baten modelizazioa eta hobekuntza. Garapen-prozesu baten bertsio ezberdinak adierazteko.
- Inferentzia motorren ezagupena eta erabilera, haien algoritmoek eskaintzen dituzten abantailak aprobetxatzeko.
- *CMS*<sup>16</sup> sistemen ezagupena eta erabilera, sistemaren interfaze bezala erabiltzeko.

---

<sup>16</sup> CMS: Sigla(ingelesez), Content Management System. Dokumentuak eta bestelako edukiak antolatu eta kudeatzeko softwarea da, normalean web-aplikazioa.

## 5. Arauak eta Erreferentziak

---

Kapitulu honetan, beste gauza askoren artean, proiektuan zehar aplikatutako araudia, erabilitako erreferentziak, metodoak zein tresnak zerrendatu eta deskribatuko dira.

### 5.1 Xedapen legalak eta aplikatutako araudia

Informatika Ingeniaritzako Graduko edo Ingeniaritza Teknikoko titulazioak bete beharreko kompetentzia profesionalak eta Gradu Amaierako Lanen izaera profesionala ezartzen duen Errege Dekretua:

- *BOE-A-2009-12977*

Administrazioa Publikoak ezarritako Sektore Publikoko Kontratuen legeak eta aurkeztutako kexak:

- *BOE-261-2007-18874*
- *BOE-A-2011-17887*
- 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.  
[https://www.boe.es/boe\\_euskera/dias/2017/11/09/pdfs/BOE-A-2017-12902-E.pdf](https://www.boe.es/boe_euskera/dias/2017/11/09/pdfs/BOE-A-2017-12902-E.pdf)
- *1\_Recurso 0006-2016 (Res 100) 05-02-16*

Kalitate-eredu eta giden inguruko informazio eta baliabideak:

- *2010-CMMI-DEV\_1.3\_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 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 antolaketa eta bere aurkezpena zehazten du. Memoria eta bere eranskinak estandar honen arabera antolatu dira, baita memoriarekin batera entregatu den webgunearen antolaketa.

## 5.2 Bibliografia

Jarraian, proiektuan zehar informazioa bilatzeko eta datuak lortzeko erabili diren erreferentziak zerrendatuko dira, ordena lexikografikoan:

1. *Algoritmos de búsqueda en grafos I* (2018). Dr. Eduardo A. RODRÍGUEZ TELLO. CINVESTAV-Tamaulipas. Hemendik eskuratuta: [DFS ALGORITMOA](#)
2. BETRADOK proiektua: *Betekizunen trazabilitate inpaktu-analisi automatikoa eta dokumentazio formalaren sorkuntza automatikoa modeloetan oinarritutako ekosistemetan* (2019), Gradu Amaierako Lana. Jon Legarda Gonzalez. Juan Manuel Pikatzak, GrAL honen tutoreak eskuratutakoa.
3. *CMMiren webgunea (Capability Maturity Model Integration)*. Hemendik eskuratuta: <https://cmmiinstitute.com/>
4. COOL lengoaiaren eskuliburua. EHSIS erramintan eskuragarri.
5. *Drupal Forums. Drupal™, for all versions.* Hemendik eskuratuta: <https://www.drupal.org/forum>
6. *Drupal: Getting Started. Drupal™, Drupal 8.* Hemendik eskuratuta: <https://www.drupal.org/docs/8/modules/search-api/getting-started>
7. *Drupal: Installing Drupal. Drupal™, Drupal 8.* Hemendik eskuratuta: <https://www.drupal.org/docs/installing-drupal>
8. GrAL eredu. Euskal Herriko Unibertsitatea (EHU), Informatika Fakultatea. Hemendik eskuratuta: [GrAL Eredua.docx](#)
9. *Guía de los Fundamentos para la Dirección de Proyectos, PMBOK, Quinta Edición* (2013). Project Management Institute, Inc. Hemendik eskuratuta: [PMBOK 5th Spanish](#)
10. *Innovation of Software Development Process across Hitachi Group* (2019). *Foundations of Next-generation MONOZUKURI Transformed by Digital Technology*. Hemendik eskuratuta: [http://www.hitachi.com/rev/archive/2019/r2019\\_03/03b05/index.html](http://www.hitachi.com/rev/archive/2019/r2019_03/03b05/index.html)
11. *Norma CCII-N2016-01: Norma de Visado de Proyectos y Actuaciones Profesionales en Ingeniería Informática* (2016). Describe el proceso de los servicios de “Visado” y de Revisión de la integridad documental”. Consejo de Colegios de Ingenieros en Informática. Hemendik eskuratuta: <https://www.cci.es/norma>. <https://www.cci.es/estandares-normalizacion-ingenieria-informatica>
12. *Norma CCII-N2016-02: Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática* (2016). Consejo de Colegios de Ingenieros en Informática. Hemendik eskuratuta: <https://www.cci.es/norma>
13. *OpenUP methodology* (2008). Eclipse Foundation – Open Unified Process. Hemendik eskuratuta: <http://www.utm.mx/~caff/doc/OpenUPWeb/index.htm>
14. *PlantUML: Getting Started. PlantUML Team.* Hemendik eskuratuta: <https://plantuml.com/es/starting>

15. *Proyecto EvDiscover: Sistematización del soporte a la gestión de estudios clínicos* (2017), *Trabajo de Fin de Grado*. Ioritz Cabero Llama. Juan Manuel Pikatzak, GrAL honen tutoreak eskuratutakoa.
16. *StackOverFlow Forum*. *Stack OverFlow*. Hemendik eskuratuta: <https://es.stackoverflow.com/>

## 5.3 Metodoak, tresnak, ereduak, metrikak eta prototipoak

Atal honetan, proiektuan zehar jarraitutako metodoak, ereduak zein metrikak, erabilitako tresnak eta sortutako prototipoak azalduko dira.

### 5.3.1 Metodoak

#### *OpenUP* metodologia

*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.

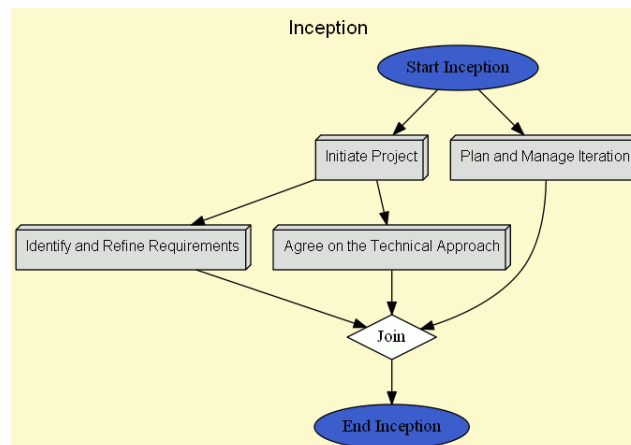
Metodologia honek garrantzi handia izan du proiektu osoan zehar. Batetik, proiektuaren helburuetako bat bizi-zikloa definitzeko metodologia baten ezarpena da eta aukeratu den bizi-zikloa *OpenUP* metodologiak definitzen duena izan da. Bestetik, proiektuaren elaborazio prozesurako *OpenUP* metodologia jarraitu da, dokumentazioa bilduz eta proiektuaren kontrola eramanez.

### 5.3.2 Tresnak

#### *Graphviz* 2.38

*Graphviz* grafikoak bistartzeko kode irekiko softwarea da. Egiturazko informazioa grafikoaren eta sare abstraktuen diagrama gisa irudikatzeko modu bat da. Aplikazio garrantzitsuak ditu sareetan, bioinformatikan, software-ingeniaritzan, datu-baseen eta webguneen diseinuan, ikaskuntza automatikoan eta beste domeinu tekniko batzuetarako interfazeetan.

Proiektu honen *workflow editorerako* erabili da, *DOT* lengoaian idatzitako *workflow*-ereduak irudi bihurtzeko (ikus 5.1 irudia).

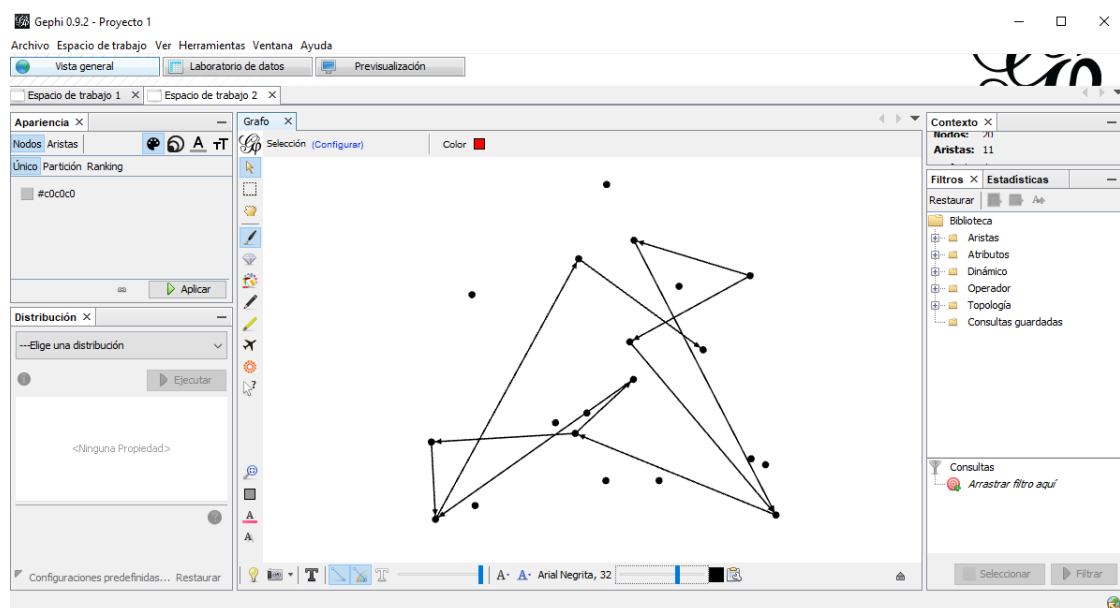


5.1 irudia: Graphviz tresnaren bitartez sortutako workflow-eredua

## Gephi v 0.9.2

Gephi mota guztietako grafiko eta sareetarako bistaratze eta esplorazio software liderra da. Kode irekiko eta doako softwarea da.

Proiektu honen *workflow editorerako* erabili da, *DOT* lengoaiaren bitartez sortutako *workflow-ereduak graphml* formatura bihurtzeko.



5.2 irudia: Gephi softwarearen interfaze grafikoa

## Protégé v 3.5

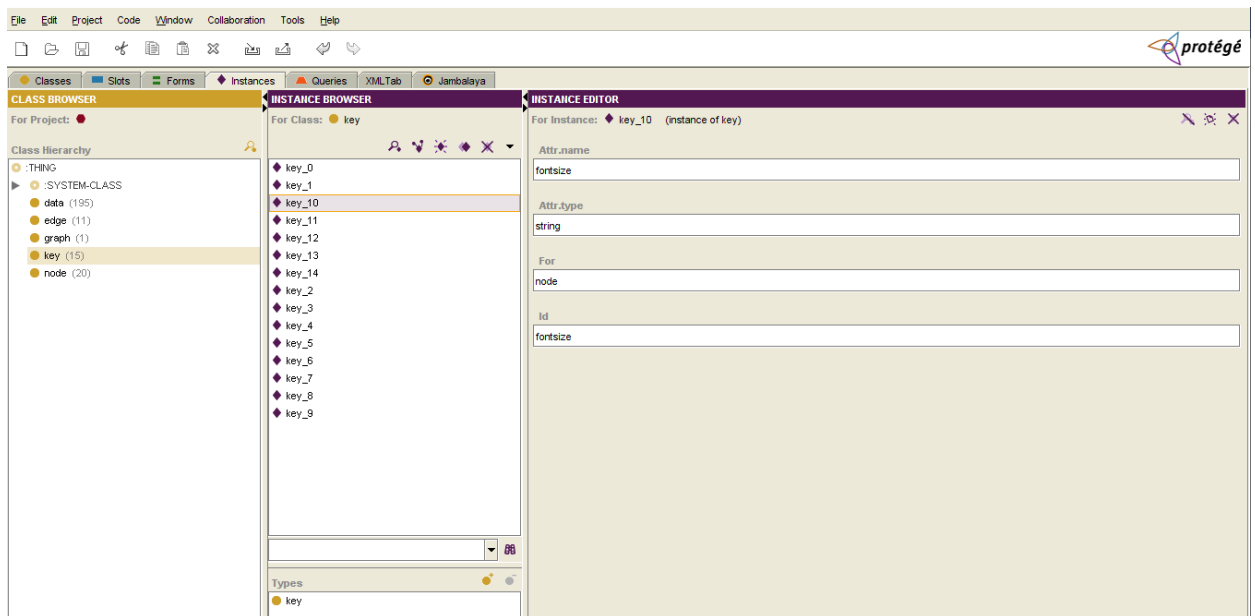
Objektu (Frame) eta ontologia esparruko kode irekiko editorea da, sistema adimendunak eraikitze softwarea.

*Protégé*k erabiltzaile akademikoek, gobernuen eta korporatiboen komunitate sendo baten babesa du. Komunitate horiek *Protégé* erabiltzen dute hainbat arloko ezagutzetan



oinarritutako irtenbideak eraikitzeko, hala nola biomedikuntzan eta merkataritza elektronikoan.

Proiektu honen *Workflow Editor*erako erabili da, *Graphml* formatuan zeuden *workflow*-ereduak *COOL* lengoaiako klase eta instantziatan bihurtzeko.

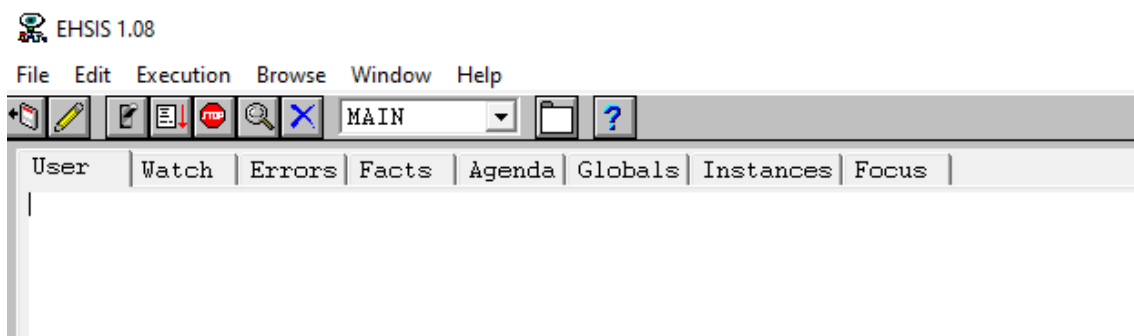


5.3 irudia: Protégé softwarearen interfaze grafikoa

### EHSIS v 1.08

Euskal Herriko Unibertsitateko (EHU) *ERABAKI* taldeak hedatutako ingurunea da, *CLIPS* 6.04, *FuzzyCLIPS* 6.04, gertaeretara orientatutako programazioa, interfazeen garapena, komunikazioa eta leihoetan oinarritutako ingurunea integratzen duena.

Proiektu honetan, bere produkzio bertsioa erabili da, *EHSIS\_RT*, workflow motorra implementatzeko.



5.4 irudia: EHSISen garapen ingurunea



## Drupal 8

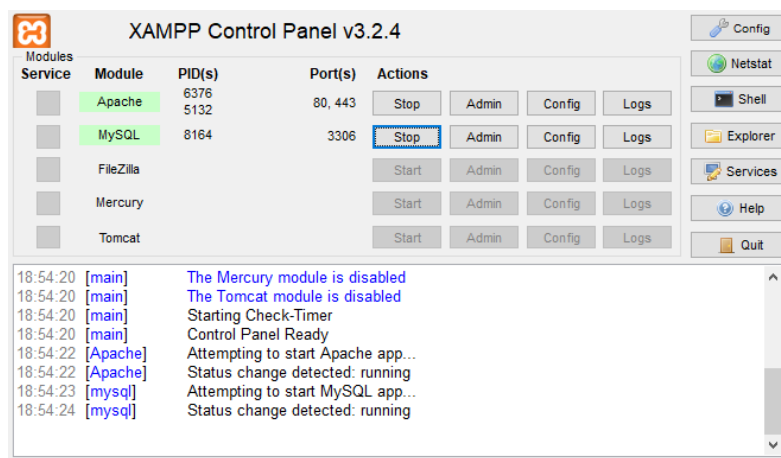
*Drupal* edukiak kudeatzeko sistema edo CMS librea, modularra eta oso konfiguragarria da. Sistema dinamikoa da, hau da, zerbitzariaren edukiak modu finkoan biltegitratu beharrean, orrien testu-edukia eta beste konfigurazio batzuk datu-base batean biltegitratzen dira eta web-ingurune bat erabiliz editatzen dira.

Proiektu honetan, *ProWF* sistemaren interfaze moduan, datuen sarrera/irteerak kudeatzeko erabili da.

## XAMPP v 3.2.4

XAMPP software libreko pakete bat da, nagusiki *MySQL* datu-baseen kudeaketa sistema eta *Apache* web zerbitzaria integratzen duena.

Proiektuan honetan *Drupal* exekutatzeke eta kudeatzeko erabili da.



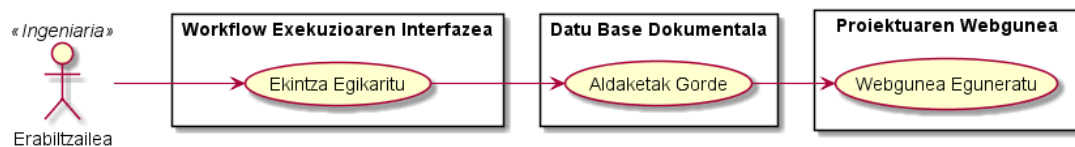
5.5 irudia: XAMPP programaren interfaze grafikoa

## PlantUML

Software irekia da. Testu laua *UML*<sup>17</sup> diagrametan bihurtzeko balio du.

Proiektu honetan, OpenUP metodologiak eskatzen dituen erabilpen kasuen ereduak sortzeko erabili da (ikus 5.6 irudia).

<sup>17</sup> UML: Sigla(ingelesez), Unified Modeling Language. Sistemak zehaztu, diseinatu eta eraikitzeke lengoaia da, printzipioz objektuei orientatutako programaziorako prestatuta dagoena.



5.6 irudia: PlantUML softwarearen bitartez sortutako diagrama

### Programazio-lengoaiak

Xehetasunetan sartu gabe, jarraian proiektu guztian zehar erabili diren programazio-lengoaiak guztiak zerrendatuko dira:

- PHP (Drupal)
- DOT (Workflow editorea)
- CLIPS (Workflow motorra)
- HTML (Drupal, proiektuaren webgunea)
- CSS (Drupal, proiektuaren webgunea)
- JavaScript (Drupal, proiektuaren webgunea)

### 5.3.3 Ereduak

#### CCII-N2016-02

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

#### OpenUP

Metodologia hau jarraitzeko, bere webgunean artefaktu bakoitzaren txantiloia dago eskuragarri. Txantiloiei horiek jarraituz *OpenUP* metodologiaren bitartez sortutako artefaktu guztiak idatzi dira.

#### Graphml ontologia

Grafo zuzendu zein ez-zuzenduen definiziorako klaseak, *CLIPSen COOL* lengoaiara itzuli daitezkeenak. <http://graphml.graphdrawing.org>

#### Grafoen sakonerako bilaketarako algoritmoa

Prozesuen grafoa sakoneran korritzeko *Depth First Search* (DFS) algoritmoa.

#### SWEBOK v3

*Software Engineering Body of Knowledge*, Software Engineering Coordinating Committee sortutako dokumentu bat da, *IEEE Computer Society* erakundeak sustatuta, *software*

ingeniaritzaren arloan dagoen ezagutzaren gida gisa definitzen dena. 2005eko bertsioa ISO/IEC TR 19759:2005 estandar gisa argitaratu zen.

### 5.3.4 Metrikak

#### Prozesuaren zuzentasuna neurtzeko sistema

Grafo eta azpigraphoetan sakonerako algoritmoa aplikatuz, sakonerako azterketa zuzen egin dela eta garapen-prozesuaren tamaina neurtzeko. Garapen prozesua *graphml* formatuan adierazita sortzen diren COOL lengoaiako objektu kopurua oso handia izan daiteke, proposatu nahi den sistemaren arkitektura definitzen denean kopuru hori kontutan hartu beharrekoa da.

### 5.3.5 Prototipoak

#### *ProWF: Workflow Editor + IO-System*

Workflow Editorea eta *Drupal* CMSaren bidez kudeatzen den *ProWF* sistema proiektu honen helburuak eta betekizunak betetzen dituen prototipoak dira.

Workflow Editoreari dagokionez, hobekuntza prozesuan egon da proiektu osoan zehar. Beraz, eredu hobeagoak lortu nahian aldagarrria izan daiteke. Sarrera/irteera sistema, ordea, ez da behin betikoa, nahiz eta proiektuaren helburuak betetzeko balio duen, oraindik funtzionalitate gehigarri asko inplementatu daitezke. Hurrengo kapituluetan aipatuko dira zeintzuk diren hobekuntza posibleak.

## 5.4 Idazkerako kalitatearen kudeaketa plana

Dokumentuen idazkera eta antolamenduaren kalitatea bermatzeko *CCII-N2016-01* eta *CCII-N2016-02* arauen gomendioak eta *OpenUP* metodologia jarraitu da, garapen prozesuan eta ezagutza bezala.

## 5.5 Beste erreferentzia batzuk

Atal honetan interesa izan duten, baina proiektuaren emaitzetan lekurik izan ez duten erreferentziak agertuko dira. Batetik, pasadan ikasturtean Softwarearen Kalitatea irakasgaian erabilitako *Bizagi* softwarea eta bestetik, software proiektuen bizi-zikloan oinarritutako beste metodologia bat.

- *Bizagi Modeler: Documentación. Bizagi.* Hemendik eskuratuta: [http://help.bizagi.com/process-modeler/es/index.html?intro\\_welcome.htm](http://help.bizagi.com/process-modeler/es/index.html?intro_welcome.htm)
- RUP methodology, *Rational Unified Process. IBM.* Hemendik eskuratuta: <https://metodoss.com/metodologia-rup/>

## 6. Definizioak eta Laburdurak

---

Kapitulu honetan, memorian eta proiektuko dokumentazioan zehar agertutako kontzeptu, gako-hitz, akronimo edo laburdurak definituko dira, haien esanahia argitzeko asmoarekin.

### 6.1 Termino eta kontzeptuen definizioak

*OpenUP* metodologiari dagokion artefaktuen terminoak bere glosarioan definitu dira, eranskinetan. Hurrengo zerrendan memorian zehar agertutako terminoei azalpena emango zaie, ordena lexikografikoan:

- **Bizagi:** Bi produktu osagarri dituen softwarea da, prozesuen modelatzaile (*Bizagi Modeler*) bat eta *BPMren suite* ofimatiko bat (*Bizagi Studio*).
- **BPM:** Sigla(ingelesez), *Business Process Management*. Enpresei prozesuak automatikoki modelatzeko, implementatzeko eta exekutatzeak aukera ematen dien software teknologia da.
- **CCII:** Sigla(gazteleraz), *Consejo de Colegios de Ingeniería Informática*. Estatu-mailan informatika ingeniari guztiak errepresentatu eta bateratzen dituen antolakundea da. Ikus, gainera: [CCII, webgunea](#).
- **CMMi:** Sigla(ingelesez), *Capability Maturity Model Integration*. Software-sistemak garatzeko, mantentzeko eta erabiltzeko, prozesuak hobetzeko eta ebaluatzeko eredua da, *CMMi* Institutuak administratutakoa. Ikus, gainera: [CMMi, webgunea](#).
- **CMS:** Sigla(ingelesez), *Content Management System*. Dokumentuak eta bestelako edukiak antolatu eta kudeatzeko softwarea da, normalean web-aplikazioa.
- **DOT:** Testu lauan idatzitako lengoaia deskriptiboa da. Grafoak deskribatzeko modu sinple bat eskaintzen du, gizakiek eta konputagailuek ulertzeko modukoa.
- **Drupal:** Edukiak kudeatzeko sistema edo *CMS* librea, modularra eta oso konfiguragarria. Ikus, gainera: [Drupal, webgunea](#).
- **Ekoizpen-prozesu:** Prozesu teknologiko bat aplikatuz eta lehengaiak edota produktu erdilanduak erabiliz, produktuak eta zerbitzuak eskaini ahal izateko egindako eragiketen edo faseen multzoa da.
- **Gephi:** Mota guztietako grafiko eta sareetarako bistaratze eta esplorazio software liderra da, kode irekikoa eta doakoa. Ikus, gainera: [Gephi, webgunea](#).
- **Graphviz:** Grafikoak bistartzeko kode irekiko softwarea da. Egiturazko informazioa grafikoen eta sare abstraktuen diagrama gisa irudikatze modu bat da. Ikus, gainera: [Graphviz, webgunea](#).
- **IBM:** Sigla(ingelesez), *International Business Machines*. Informatikarekin lotutako tresnak, programak eta zerbitzuak ekoiztu eta merkaturatzen dituen enpresa multinazionala da.



- **OpenUP:** Sigla(ingelesez), *Open Unified Process*. *RUP (Rational Unified Process)* metodologiaren azpimultzo bat da. Proiektu informatiko batean kokatzeko eta informazioa antolatuta izateko balio du. Horretaz aparte, proiektuaren elaborazio fasean produktua zein izan daitekeen edo zein bidetik lortu daitekeen definitzeko balio du. Ikus, gainera: [OpenUp, webgunea](#).
- **PlantUML:** Testu laua *UML* diagrametan bihurtzeko balio duen software irekia. Ikus, gainera: [PlantUML, webgunea](#).
- **Protégé:** Ontologia esparruko kode irekiko editorea da, sistema adimendunak eraikitze softwarea. Ikus, gainera: [Protege, webgunea](#).
- **ProWF:** Ingelesko hitzetatik eratutako hitz-jokoa da. “Pro” *professional* hitzetik datorren laburdura da eta profesionala esan nahi du, “WF” *workflow* hitzetik datorren laburdura da eta lan-fluxu esan nahi du.
- **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.
- **Softwarearen bizi-zikloa:** Software-produktu baten garapenari aplikatutako egitura da. Softwarea garatzeko prozesu bat ezartzeko hainbat eredu daude, eta horietako bakoitzak ikuspegi ezberdin bat deskribatzen du prozesuan zehar egiten diren jarduerak ezberdinetarako.
- **SPICE:** Akronimoa(ingelesez), *Software Process Improvement and Capability Determination* edo *ISO/IEC 15504*. Garapen-prozesuak hobetzeko, ebaluatze, informazio-sistemak eta software-produktuak mantentzeko eredu da. Ikus, gainera: [SPICE, webgunea](#).
- **UML:** Sigla(ingelesez), *Unified Modeling Language*. Sistemak zehaztu, diseinatu eta eraikitze lengoaia da, printzipioz objektuei orientatutako programaziorako prestatuta dagoena.
- **UNE:** Sigla(gaztelaraz), *Una Norma Española*. *Comités Técnicos de Normalización (CTN)* batzordeak sortutako arauen, arau esperimentalen eta txostenen (estandarrek) multzoak dira. Ikus, gainera: [UNE, webgunea](#).
- **Workflow, lan-fluxu:** 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.

## 7. Hasierako Betekizunak

---

Kapitulu honetan, proiektuaren hasierako betekizunak jasotzen dira, funtzionalak eta ez-funtzionalak.

Hala eta guztiz ere, sistemaren betekizunak *OpenUP* metodologiaren bitartez, proiektuaren garapen prozesuan zehar harrapatu dira. Betekizun horien inguruko argibide gehiago izateko hurrengo dokumentuak eskuragarri daude proiektuaren webgunean:

- *Glossary* (Glosarioa)
- *Vision* (Bisioa)
- *System-Wide Requirements Specification* (Betekizunen espezifikazioa)
- *Use Case 1, Use-Case Model 1* (Lehenengo erabilpen kasua eta bere eredua)
- *Use Case 2, Use-Case Model 2* (Bigarren erabilpen kasua eta bere eredua)
- *Use Case 3, Use-Case Model 3* (Hirugarren erabilpen kasua eta bere eredua)

### 7.1 Betekizun funtzionalak

Hasieran identifikatutako betekizun funtzionalen artean, honako hauek aurki ditzakegu:

- **Software proiektuen elaboraziorako bizi-zikloa definitzeko metodologia bat aukeratzea.** Aukeratutako metodologia horren bizi-zikloa definituko da *workflow*-lengoiaren bitartez, bizi-zikloa jarraitzeko gidak sortuz.
- **Metodologiak definitzen dituzten softwarearen bizi-zikloa adierazteko lengoia bat asmatzea, *workflow*-lengoia.** Metodologiaren ezagutza *workflow* batean jarri ahal izateko eta ondoren, automatikoki inferentzia motor baten kode bihurtzeko.
- **Prozesuetan edo *workflowetan* oinarritutako sistema baten bideragarritasuna aztertzea.** Azken finean, *BPMn* oinarritutako teknologiekin egin daitekeena *workflowetan* eta software irekitan oinarrituta egitea, menpekotasun teknologikoa ekiditeko asmoarekin.
- ***Workflowak* kudeatzeko sistema bat, prozesu edo *workflowsen* sorkuntzarako editore ezagun bat eta sistematizazioa zein automatizazioa bultzatzen duten estandarren bilaketa.**
- **Prozesuak edo *workflowak* prozesatzeko motorraren aukeraketa eta ezagutzapena,** inferentzia motorra sistemaren arkitekturan integratzea.
- **Datu eta informazioaren sarrera/irteera integratuta edo *CMS* bidezkoa izatearen erabakia hartzea.** *CMS* ezberdinen bideragarritasuna aztertu eta ondoren, *CMS* bat aukeratu edo sistema bat integratzearen erabakia hartzea da.
- **Datuen biltegitratze eta kudeaketarako datu-basen aukeraketa eta diseinua egitea.** Batetik, *workflowsaren* edukia biltegitratzeko datu-base erlazionala eta bestetik, metodologiak sortutako artefaktuak biltegitratzeko datu-base dokumentala behar da.

- **Erabiltzaileek izango duten rolak zehaztu.** Gutxienez *workflowak* sortzeko/editatzeko eta *workflowak* exekutatzeke rolak behar dira. Proiektuaren zuzendaria eta kalitatea bermatzeko arduradunaren rolak ere erabili daitezke.

## 7.2 Betekizun ez-funtzionalak

Hasieran identifikatutako betekizun ez-funtzionalen artean, honako hauek aurki ditzakegu:

- ***OpenUP* metodologia jarraitu eta metodologia horren bitartez sortutako dokumentuen antolaketarako eta proiektuaren aurkezpenerako CCII-2016N-02 estandarra betetzen duen webgunea sortu behar da.** Informazioa partekatu, proiektuak ikuskatu eta bidezko modu batean ebaluatzeko oso lagungarria delako.
- **Soluzioaren garapenean zehar urratsez-urrats *CMMIra* begira egon beharko da.** *CMMIren* heldutasun-mailak etorkizunean lortu ahal izanez, erronka epe luzean mantenduko da.
- **Soluzioa berrerabilgarria izan behar da softwarearen bizi-zikloa definitzen duen beste metodologiaren bat inplementatzeko.** Bezeroek erabakiko dute zein izango den proiektu baten aplikatu beharreko garapen metodologia.
- **Garatutako sistema hobekuntzak izateko prest egongo behar da, bere eraginkortasuna hobetzeko helburuarekin.** Metodologia aldaketak eta beraien bertsioen hobekuntzek garapen-prozesuan eta sistema osoan aldaketak ekar ditzake.

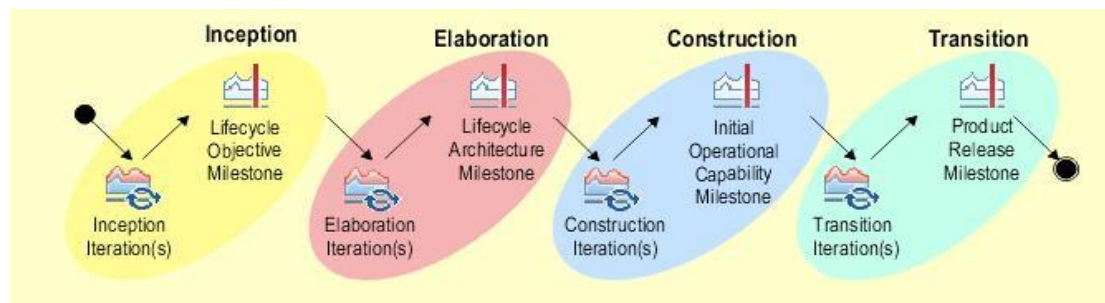


## 8. Irismena

Kapitulu honek proiektuaren irismena definitzea, proiektuak sortutako entregagarriak zerrendatzea eta kanpoan gelditutako aspektuak azaltzea du helburu.

### 8.1 Proiektuaren irismena

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 **elaborazio** fasera helduko da, bertan produktuaren lehenengo prototipoa ateraz, bere probekin. Denbora mugatuko proiektua izanik, eraikuntza eta trantsizio faseak kanpoan geratu dira.

Irismena finkatzeko hurrengoak hartu da kontuan:

1. **OpenUp metodologiak** eskatutako dokumentuak betetzea. Horretarako OpenUP metodologiak bere webgunean eskaintzen dituen txantiloak jarraituz.
2. **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.
3. **ProWF sistema** inplementatzea eta probatzea. "*Proposatutako Sistemaren Deskribapend*" kapituluaz azalduko da, instalazio eta erabiltzailearen eskuliburuaren dokumentuekin batera.
4. **Proiektuaren memoria** idaztea. Dokumentu hau eta bere eranskinak.
5. **Proiektuaren defentsa** prestatzea. Horretarako, memorian idatzitako aspektu guztiak laburbiltzen dituen aurkezpen bat prestatuz.

## 8.2 Proiektuak sortutako entregagarriak

CCII-N2016-02 araua jarraituz, jarraian proiektuan zehar sortuko diren entregagarriak agertuko dira:

- Proiektuaren memoria, dokumentua.
- Proiektuaren defentsa, aurkezpena.
- Proiektuaren webgunea
  - Proiektuaren memoria eta defentsa barne
  - Memoriaren eranskinak
  - Proiektuaren posterra
  - OpenUP metodologiako artefaktuak eranskin bezala (ikus 8.1 taula)
  - Proiektuaren barne kudeaketa

Diziplina	Artefaktua
Arkitektura	Arkitektura Koadernoa
Garapena	Diseinua Eraikuntza Inplementazioa
Kudeaketa	Iterazio Plana Proiektu Plana Lan-atazen zerrenda Arriskuen zerrenda
Betekizunak	Glosarioa Bisioa Betebeharren Espezifikazioa Erabilpen Kasuak Erabilpen Kasuen Eredua

8.1 taula: OpenUP metodologiako artefaktuak

## 9. Hipotesiak eta Murritzapenak

---

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

### 9.1 Hipotesiak

Hurrengoak dira proiektuaren hasierako hipotesiak:

- Inferentzia motorraren erabilerak azkartasuna ematen du prozesuen exekuzioan, kode ahaltzu eta erraza erabiliz. Definitu daitekeen garapen prozesua, testu formatutik COOL objektuetara automatikoki itzuli daiteke klase hierarkia finkoa eta berrerabilgarria mantenduz. Gainera, prozesuaren edo lan-fluxuaren definitze lengoaia aldatu ezean, prozesuaren exekuziorako erregelak ere berrerabil daitezke. Inferentzia motor bat baino gehiago erabiliz, sistema paralelo bat sortu daiteke.
- Definitutako garapen prozesuak software proiektuen elaborazioa gidatu eta kontrolatuko du. Prozesua aldatzeak sistemaren portaera eta datu-basea automatikoki aldatzea ekarriko du.
- CMS baten erabilera datuen sarrera/irteerarako irtenbide egokiena da. Webgune bat sortzeko aukera ematen duen tresna erabilerraza izateaz aparte, ez da baliabide tekniko aurreratuetara etengabe jo behar. Kudeaketa, administrazioa eta mantentze-lanak egiteko laguntza ematen du kanpoko baliabiderik erabili gabe.
- Datu-base erlazionalak prozesu edo *workflow* 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. Proiektu honetan hurrengoak izan dira hasierako murrizketa horiek:

- Betekizunen ingeniartza eta bizi-zikloa definitzen duen metodologia bat jarraitzea proiektuaren elaborazio eta garapenerako: *OpenUP*.
- Proiektuaren aurkezpena eta dokumentuen antolaketarako estandar bat erabiltzea: *CCII-2016N-02*.
- Sortuko den software proiektuen elaboraziorako sistema web bidez atzigarria izan behar du.



## 10. Aukeren Ikerketa eta Egingarritasuna

---

Kapitulu honetan, proiektuko soluzioaren alternatibak, balorazio-irizpideak eta hautatutako zein kanpoan gelditutako aukeren justifikazioak agertzen dira.

### 10.1 Arkitekturaren erabakia

Soluzioaren arkitekturari dagokionez, bi aukera aztertu dira: *Bizagi* bezalako softwarearen bitartez prozesuetan (BPM) oinarritutako web-aplikazioa eraikitzea edo arkitektura propioa sortzea.

#### a) Bizagi

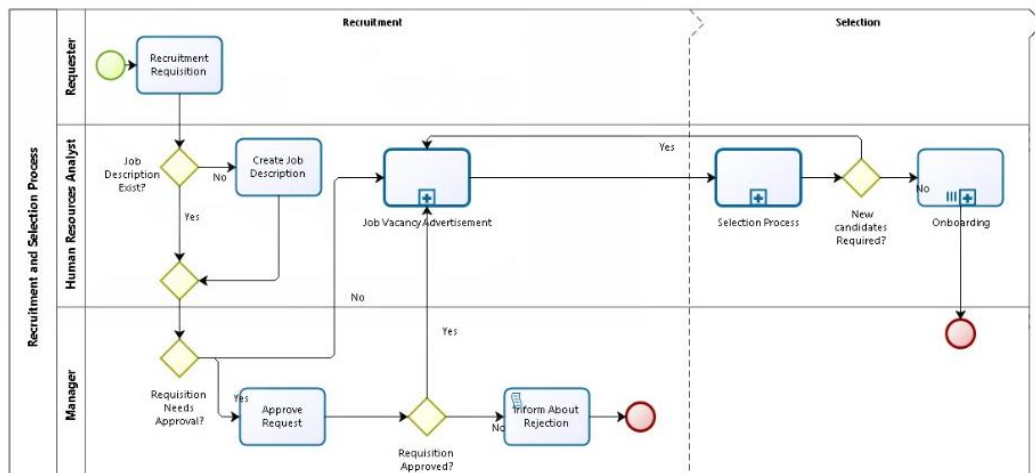
*Bizagi* pasadan ikasturtean, 3. mailan, Softwarearen Kalitatea irakasgaian erabilitako softwarea da. Bertan, *Bizagi Modeler* editorearen bitartez prozesuak sortu eta ondoren, *Bizagi Studio* tresnan negozio-erregelak adieraziz, datu-basea konfiguraturaz, formularioak definituz, web-zerbitzuak integraturaz 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 dituzte eta editore horren bitartez horiek definitu daitezke (ikus 10.1 irudia). Beraz, *Bizagi Modeler* erabiliz, bere lengoaiaren (BPMN) arauak errespetaturaz, *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

## b) Arkitektura propioa

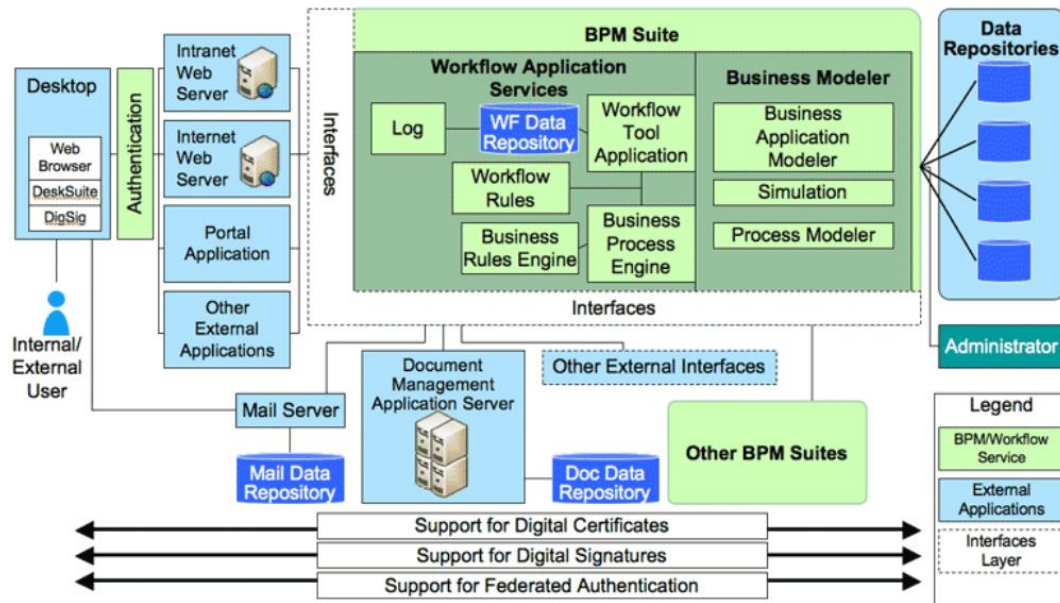
*Workflow*etan oinarritutako web-aplikazio bat zerotik eraikitzea apustu handia zen, denbora gehiago eman beharko litzateke arkitekturaren garapenean, baina, ordea, pisuzko arazoak zituen:

- *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.

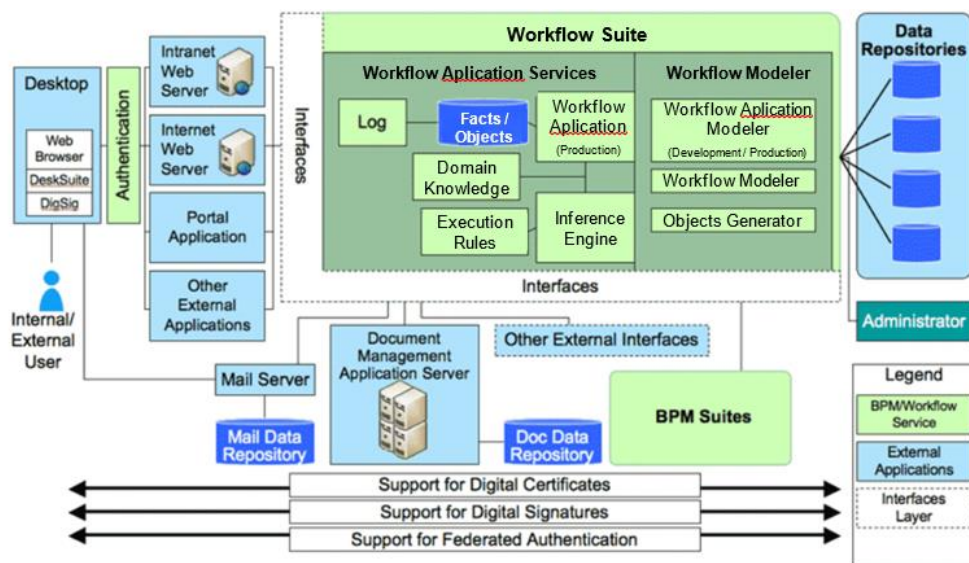
Hori dela eta, erabakia **arkitektura zerotik eraikitzea** izan zen. Hurrengo kapituluaren sartuko gara soluzioaren deskribapenean.

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 eta 10.3 irudietan, *Bizagiren* arkitekturaren eta *workflow*etan oinarritutako arkitektura propioaren arteko alderaketa ikus daiteke. Esan bearra dago 10.3 irudian agertzen diren funtzionalitate guztiak ez direla inplementatuko, *BPM* eta *workflow* sistema kudeatzailea alderatzea bakarrik du helburu.

<sup>18</sup> BPMN: sigla (ingelesez), *Business Process Model and Notation*. Notazio grafiko estandarizatu bat da, BPMak lan-fluxuko formatu batean modelatzeko aukera ematen duena.



10.2 irudia: Bizagiren arkitektura



10.3 irudia: Workflowetan oinarritutako arkitektura

## 10.2 Bizi-zikloa definitzen duen metodologiaren aukeraketa

Proiektu honen helburuetako bat sortutako *workflow*-lengoaia erabiliz bizi-zikloa definitzen duen metodologia bat ezartzea da. Metodologiaren aukeraketak ez du denbora askorik eraman, hasieratik argi zegoen zein zen egokiena: **OpenUP**.

*OpenUP* aurreko urteko, hirugarren mailan, Softwarearen Kalitatea irakasgaian erabili zen metodologia da, beraz proiektuaren hasieratik ezaguna izan da. Gainera, aldi berean proiektu honen bizi-zikloa finkatzeko eta dokumentuen sorkuntzarako *OpenUP* metodologia jarraitzea

ezarri zen, beraz, oso komenigarria zen metodologia berdina definitzea *workflow*-eredua sortzerako orduan.

**Beste aukera bat *RUP (Rational Unified Process)* metodologia arina erabiltzea zen**, honek bizi-ziklo definitzen du ere, baina, azken finean *OpenUP* metodologiak *RUP*en funtsezko ezaugarriak hartzen ditu, garapen prozesua sinplifikatuz eta bere printzipioei leial izaten jarraituz.

Beste alde batetik, *OpenUP* metodologiaren [webguneak](#) pisu handia izan zuen erabakian. Webgunean bete behar diren dokumentuen/artefaktuen txantiloak egoteaz gain, bizi-zikloaren eta bizi-zikloan parte hartzen duten elementu guztien deskribapen zehatzak eta diagramak daude. Faseak, aktibitateak, jarduerak, rolak, artefaktuak, iterazioak eta haien arteko loturak deskribatzen dira besteak beste, beraz, webgunean agertzen dena *workflow*-lengoaia baliatuz eraldatzea ez zan zaila izango metodologia honekin.

### 10.3 *Workflow*-lengoaia bilatu

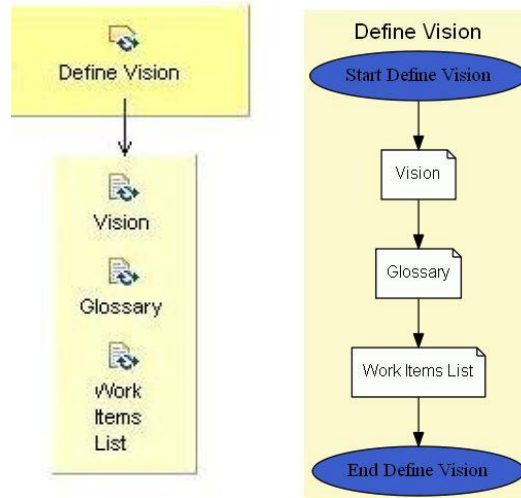
*Workflow*-lengoaiak bi betebeharrak ditu, ulergarria izatea eta inferentzia motorrak exekutatzeko prest egotea. Lengoaia definitiboa aurkitzea proiektuko ataza garrantzitsuenetarikoa izan da eta denbora gehien eman duena.

*Workflow*-lengoaia *DOT* lengoaia deskriptiboarekin definitzea erabaki zen. Ondoren, *DOT* lengoaiaren bitartez sortutako testu sinplea, batetik, *GraphViz* erabiliz irudi bihurtu eta, bestetik, inferentzia motorrak exekutatu ahal izateko EHSIS inguruneak erabiltzen duen *COOL* lengoaiara bihurtu behar zen. Bestetik, *workflows* nabigagarria izatea erabaki zen, bizi-zikloa *workflow* eta azpi-*workflow*etan banatuz, prozesuaren irudia ulergarriagoa eta garbiagoa izanik.

Hurrengoa izan da *workflow*-lengoaiaren azkenengo bertsioa sortzeko bidea:

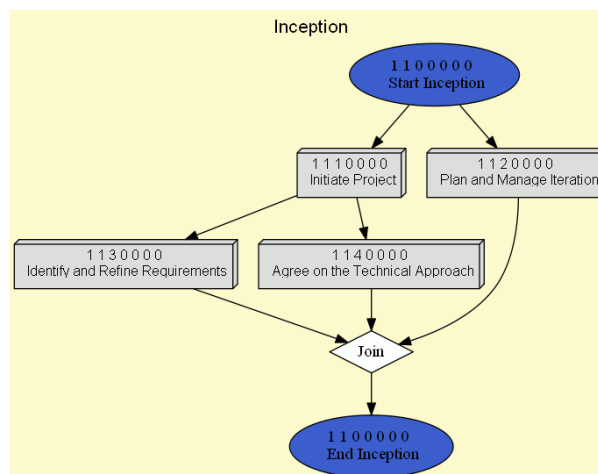
1. **Ikusgarritasuna.** Hasiera baten, aukeratutako metodologiaren (*OpenUP*) bizi-zikloa definitzeko ikusgarritasuna bakarrik bilatu zen, inferentzia motorra eta bere exekuzioa kontuan izan gabe. Testu sinplea *workflows*en irudi bihurtzean *OpenUP* metodologiak bere webgunean erabiltzen dituen koloreak eta formak errespetatzea bilatu zen, 10.4 irudian ikus daitekeen bezala. Ikusgarritasunaren aldetik ez da ezer aldatu, bertsio finalean horrela mantendu da.





10.4 irudia: ezkerrean, OpenUPen webgunean agertzen den “Define Vision” jardura, eskuinean, workflow-lengoiaren bitartez sortutako irudia

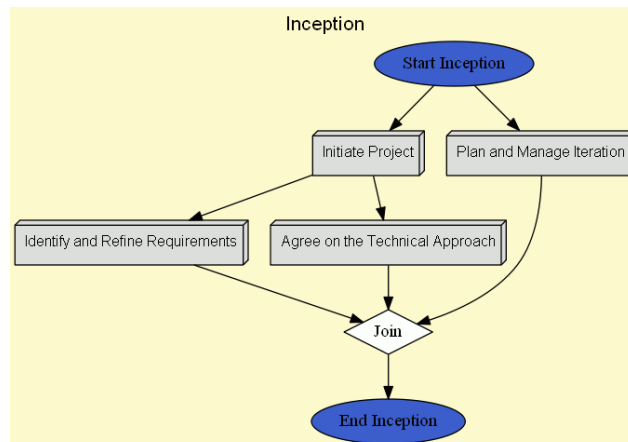
- Exekuzioan pentsatzen.** Hurrengo bertsioetan *workflowaren* informazioa datu-base erlazionalean gordetzea eta inferentzia motorrak exekutatu ahal izatea bilatu zen. Horretarako, informazioa datu-basean gordetzeko kode numerikoak erabiltzea erabaki zen, beraz kode numeriko horiek *workflowan* jarri ziran, datu-basean gorde beharreko informazioa ere irudian agertuz (ikus 10.5 irudia). Bertsio hauetan ikusgarritasuna alde batera utzi eta ulergarritasuna galdu zen.



10.5 irudia: Datu-basean gorde beharreko kode numerikoak workflowan agertzen dira

- Irudia eta negozio-logikaren banaketa.** Urrats garrantzitsuena izan zela esan daiteke, *workflowaren* irudian ikusten denaren eta exekuzioarekin zein datu-basearekin zerikusia duenaren arteko banaketa egitea zuen helburu. *Workflowaren* informazioa harrapatzeko modu hobea aurkitu zen: metodologiaren inguruko informazio guztia nodoen izenetan jartzea. *DOT* lengoiaren bitartez, nodoaren izena eta etiketaren arteko ezberdintasuna egin zen, nodoaren izenean negozio-logikarekin zerikusia duena jarri eta etiketan, ordea, *workflowaren* irudian agertuko dena. Nodoaren izenean jarritako guztia

ez zen irudian ikusiko, hori dela eta, *workflow* garbiagoak eta nabigagarritasuna ulergarriagoa lortu zen (ikus 10.6 irudia).



10.6 irudia: Negozio-logika ez da lan-fluxuan agertzen

4. **Optimizazioa eta akatsen zuzenketa.** Azkenengo bertsioa lortzeko funtsezkoa izan zen, exekuzioan zehar agertutako arazoak konpondu eta exekuzioa arintzea bilatuz.

*Workflow*-lengoaia hobekuntzan egon da proiektu osoan zehar, esan daiteke lengoaiaren hobekuntza-prozesuak ez duela amaierarik. Seguruenik, prozesuaren exekuzioa arintzeko oraindik hainbat aldaketa egin daitezke.

## 10.4 CMSak probatu

Soluzioaren datu zein informazioaren sarrera/irteerak kudeatzeko web-aplikazioa bat sortzea erabaki zenez, web-aplikazio hori erraz sortzeko *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 ziran ere. Hiru *CMS* aztertu ziran nagusiki: *Wordpress*, *Joomla* eta *Drupal*.

10.1 taulan hiru *CMS* horien konparazio bat ikus daiteke erabakia hartzeko gehien nabarmentzen diren puntuekin.

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 ziran.

- 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*<sup>19</sup> enpresak *Drupal* erabiltzen du oinarri bezala.

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

10.1 taula: CMS ezberdinen konparazio-taula

## 10.5 Inferentzia-motorraren erabakia

Inferentzia-motorraren bitartez *workflow*-lengoian idatzitakoa exekutatu da, goitik behera, horretarako beharrezkoa da *workflow*-lengoiaren eta motorrak erabiliko duen lengoiaren arteko eraldaketa bat egitea.

Inferentzia-motor bezala *CLIPS* erabiltzea erabaki zen. Sistema adituentzako beste motor batzuk bezala, *CLIPSe* egitate (ebidentzia) eta erregelak erabiltzen ditu elementu nagusi bezala. Egitateak erabilpen kasuaren inguruko informazioa adierazteko erabiltzen dira. Erregelak, ordea, erabilpen kasuaren informaziotik informazio berria inferitzeko. Honek, sistema eboluzionatzea ahalbidetzen du, normalean egitateak gehitu edo aldatuz. Erregelen baldintza aldeak betetzean, eta ondorioz, beraien ekintzak exekutatzean, emaitzak egitate-basean geratuko dira. Egitate-basearen informazioa datu-base edo fitxategietatik hartu daiteke eta alderantzizkoa egin ere.

*Workflow*-lengoaia eta *EHSIS* inguruneak erabiltzen duen *COOL* lengoiaren arteko eraldaketa egiteko hurrengo prozesua egitea adostu zen:

1. *DOT* lengoian idatzitako eredua *graphml* formatura eraldatzea, *Gephi* softwarea erabiliz. XML lengoian oinarrituta, *Graphml* lengoiak grafiko edo diagramen datuak partekatze balio du.

<sup>19</sup> Acquia: Software enpresa bat da, Drupal kudeatzeko enpresa-produktuak, zerbitzuak eta euskarri teknika ematen dituena. <https://www.acquia.com/es>

2. Graphml formatuan eraldatutako eredu hori *Protégé* v3.5 softwarearen bitartez klase eta instantzietan bihurtzea. Azken finean, *workflowan* agertzen den informazioa COOL lengoaiaren objektu bihurtzea du helburu.
3. Sortutako klase eta instantziak EHSIS inferentzia motorrarekin exekutagarriak izateko egokitu.

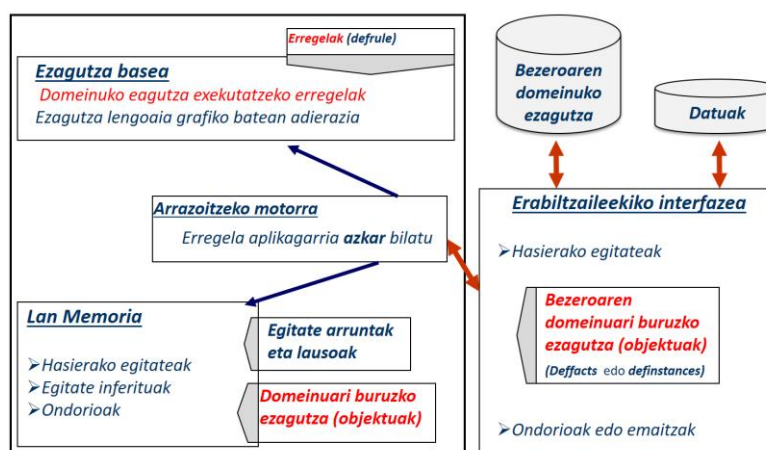
Definitu behar dugun *workflowa* nabigagarria izatea erabaki zenez eraldaketa-prozesu hori azpi-*workflow* bakoitzarekin errepikatu behar da, prozesu oso errepikakorra eta neketsua bihurtuz. Hori ekiditeko metaeredu bat sortzeko ideia agertu zen.

*Model-driven Engineering (MDE)* softwarea garatzeko metodologia bat da, domeinu-ereduak sortu eta ustatzen dituen, arazo espezifiko batekin lotutako gai guztien eredu kontzeptualak. Metaeredua eredu bat definitzeko erregela, murriztapen, teoria eta eskemen bilduma gisa definitzeko lengoia mota da. Metaeredu batek eredu bat definitzen du, eredu horrek beharrezkoak dituen ezaugarri guztiak azalduz.

Inferentzia motorraren kasuan metaeredu bat ezartzeak produktibitatea handitzea du helburu, *DOT* lengoia deskriptiboaren eta *COOL* lengoaiaren arteko eredu bat sortuz, urrats bakar baten bidez eraldaketa eginez eta kanpoko softwarerik erabili gabe.

Hala ere, metaeredu bat definitzeak denbora-kostu eta esfortzu handia ekarriko zuen eta horregatik, **proiektu honen irismenetik kanpo geratu zen**. Baina, oso interesgarria eta onurak dituen hobekuntza denez, ez da etorkizunerako baztertzen.

Metaeredu bidez edo *Gephi* eta *Protégé* erremintekin, definitutako prozesua COOL lengoaiara itzultzeaz gain, prozesu eta azpiprosesuen informazio daramaten objektuak exekutatzeak kode sinple, azkar eta mantengarri bat gehitu behar zaio. Inferentzia motorrak lagungarriak dira honetarako baina OPenUP metodologiari buruzko ezagutza guztia objektu bezala adieraztea oso komenigarria da (ikus 10.7 irudia).



10.7 irudia: Domeinuari (OpenUP) buruzko ezagutza objektu bezala jarrita, informazio hori exekutatzeak erregela gutxi batzuekin nahikoa da.

## 11. Proposatutako Sistemaren Deskribapena

---

Kapitulu honetan planteatutako arazoa konpontzeko proposatzen den soluzioa, bere osagaiak eta bere ezaugarriak deskribatzen dira.

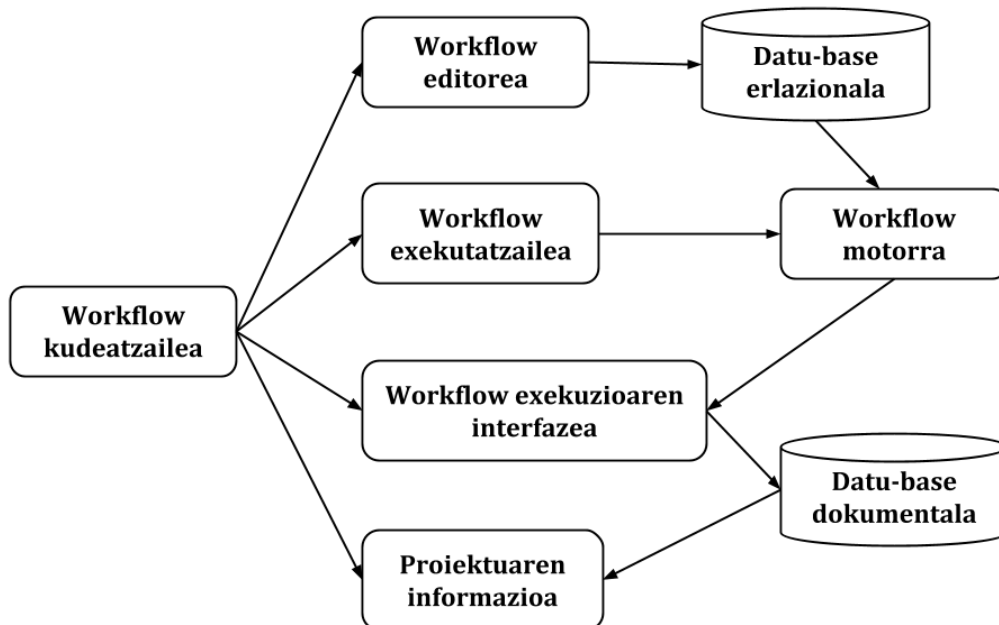
### 11.1 Sistemaren testuingurua

Proposatutako sistemaren testuinguru osoaz jabetzeko, arazoaren eta sistemaren kokapena ondo ulertu behar da, horretarako memoriaren eranskinetako “Sistemaren Espezifikazioa” kapituluaren dokumentuak irakurtzeak ikuspegi tekniko sakonagoa emango du.

Planteatutako arazotik hurrengo beharrak eta ezaugarriak ateratzen 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 rolak esleituiko 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 (ikus 11.1 irudia):
  - **Workflow editorea:** sortutako *workflow*-lengoaia erabiliz *workflow*-ereduak sortzeko balio du, ondoren, eredu horiek *workflow* motorrak exekutatzeke eraldatuko ditu.
  - **Workflow motorra:** bere lengoiara 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 biltegiratuko du.

- **Datu-base dokumentala:** workflowaren 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.

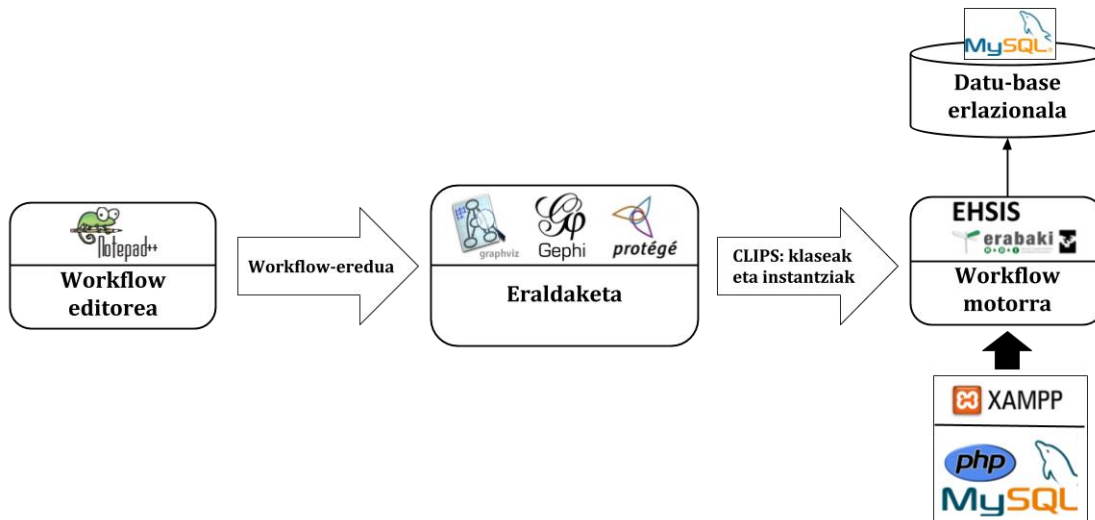


11.1 irudia: ProWF sistemak behar dituen osagaiak eta haien arteko erlazioak

- *ProWF* sistema bi azpisistema ezberdinetan bananduta egongo da: **Workflow Editor** eta **IO-System**.
  - **Workflow Editor:** sortutako *workflow*-lengoaia erabiliz *workflow*-ereduak sortzeko balio du, ondoren, eredu horiek *workflow* motorrak exekutatzeke eraldatuko ditu. Testu-editore bat izango da. Rolak: Prozesu sortzailea.
  - **IO-System:** *worklow* kudeatzailearen sarrera/irteerak maneiatzea du helburu. *CMS* baten bitartez kudeatutako web-aplikazioa izango da. Rolak: Proiektuko zuzendaria, analista eta kalitate arduraduna.

## 11.2 ProWF: Workflow Editor

### 11.2.1 Arkitektura



11.2 irudia: Workflow Editor azpisistemaren arkitektura

Jarraian, arkitekturaren osagai eta elementu guztiak gainetik deskribatuko dira:

- Workflow editorea: definitutako lengoia grafiko bat erabilita, testu editore batekin, DOT lengoiaren bitartez *workflow*-ereduak sortzeko balio du. “*Workflow-lengoiaren Eskuliburua*” dokumentuan jarri dira lengoia grafikoaren xehetasunak.
- Eraldaketa-prozesua: DOT lengoian idatzitako *workflow*-eredua CLIPS lengoiako klase eta instantzietan eraldatzea du helburu. Horretarako, “*Workflow Editor - Eskuliburua*” dokumentu jarraitu behar da.
- Datu-base erlazionala: sortutako *workflow*-ereduen informazio gordetzeaz arduratzen da.
- Workflow motorra: COOL lengoiara eraldatutako *workflow*-ereduaren klaseak eta instantziak baliatuz, *workflows* exekutatzeko erregela batzuen bidez, informazio guztia datu-base erlazionalean gordeko du.

### 11.2.2 Erabilitako teknologiak

Atal honetan, *Workflow Editor* azpisistemak behar dituen teknologiak zerrendatuko dira.

#### Notepad++

DOT lengoiaren bitartez *workflow*-ereduak sortzeko erabili den testu-editorea. Edozein testu-editore erabili daiteke, adibidez, Windowseko *Bloc de Notas* edo *WordPad*.



### Graphviz

*Dot* komandoen bidez, DOT lengoian definitutako workflow-ereduak irudi garbian bihurtzeko erabili den kotsola-tresna da. DOT lengoiaren arauak errespetatu behar dira irudia sortzeko.

### Gephi

DOT lengoian definitutako workflow-ereduak *graphml* (XML) formatura eraldatzeko erabili den softwarea.

### Protégé

*Graphml* formatuan eraldatutako *workflow*-ereduak *COOL* lengoiako klase eta instantzia bihurtzeko erabili den softwarea.

### XAMPP → Apache, MySQL

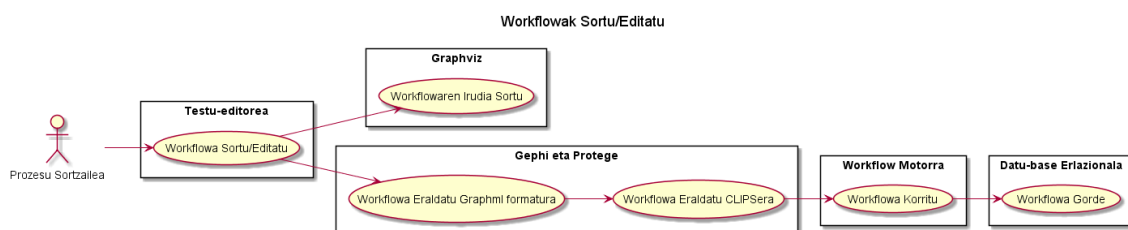
*Workflow* motorrak datu-base erlazionalaren atzipena izateko erabili da, MySQLren bitartez.

### EHSIS\_RT

EHSIS ingurunearen produkzio bertsio da. *COOL* lengoiara eraldatutako *workflow*-ereduaren klaseak eta instantziak baliatuz, erregelen bitartez informazio guztia datu-base erlazionalan gordetzeko erabili den inferentzia motorra da. Berarekin lotutako EHSIS garapen ingurunea ere erabili da probak egiteko.

## 11.2.3 Analisia

Azpisistema honi dagozkion erabilpen-kasuak 11.3 irudiko erabilera-kasuen diagraman ikus daitezke.



11.3 irudia: Workflow Editor azpisistemaren erabilpen kasuen diagrama. PlantUML.

### 11.2.3.1 Workflowaren irudia sortu

**Deskribapena:** Erabilpen-kasu honek testu-editorean sortutako *workflow*-ereduak irudi bihurtzea du helburu.

**Aktorea:** Prozesu Sortzailea

**Aurrebaldintzak:**

- Testu-editorean sortutako workflowa DOT lengoiaren eta workflow-lengoiaren arauak jarraitu behar izan ditu.



**Postbaldintzak:**

- Irudia sortuta egongo da, *GIF* formatuan. Beste formatu batzuk ere aukeratu daitezke.

**Gertakarien oinarrizko fluxua:**

1. Prozesu sortzaileak workflow-eredua sortu edo aldatuko du testu-editorean.
2. *Graphviz* kontsola-tresnaren bidez, *dot* komandoak erabilita, workflow-eredua irudi bihurtuko du.
3. *Workflow* nabigagarria denez, 1 eta 2 urratsak errepikatu *workflow* eta azpi-*workflow* bakoitzeko.

### 11.2.3.1 Workflowaren informazioa datu-base erlazionalean gorde

**Deskribapena:** Erabilpen-kasu honen helburua testu-editorean sortutako *workflow*-ereduen informazioa datu-base erlazionalean biltegiatzea da.

**Aktorea:** Prozesu Sortzailea

**Aurrebaldintzak:**

- Testu-editorean sortutako workflowa DOT lengoiaren eta workflow-lengoiaren arauak jarraitu behar izan ditu.
- *Workflow* motorrak datu-base erlazionalaren atzipena behar du.

**Postbaldintzak:**

- *Workflow*-ereduaren informazioa datu-base erlazionalean gordeta egongo da.

**Gertakarien oinarrizko fluxua:**

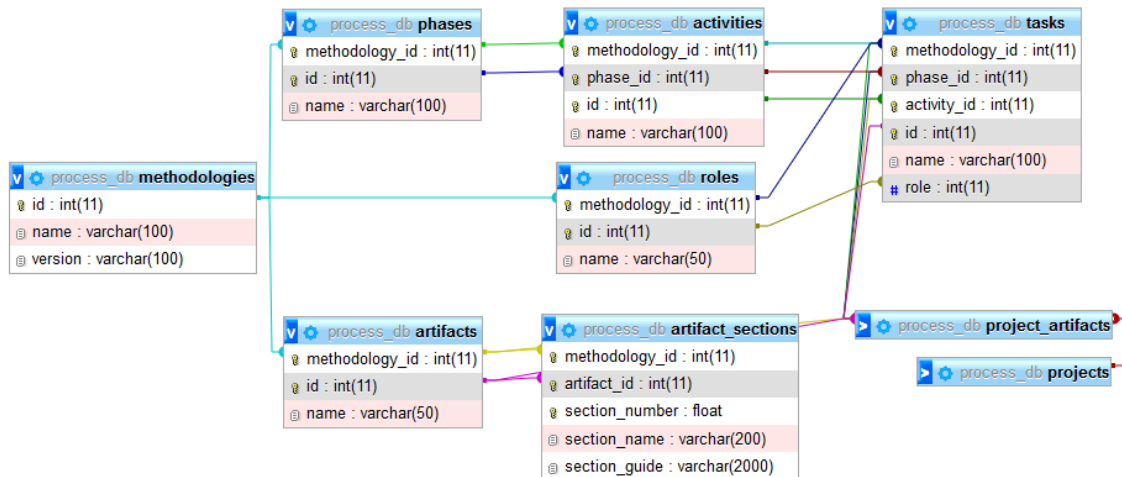
1. Prozesu sortzaileak workflow-eredua sortu edo aldatuko du testu-editorean.
2. *Gephi* softwarearen bidez, *workflow*-eredua *graphml* formatura eraldatuko du.
3. *Protégé* softwarearen bidez, *graphml* formatura eraldatutako *workflow*-eredua COOL lengoiako klase eta instantzietara eraldatuko du.
4. *Workflow* nabigagarria denez, 2 eta 3 urratsak errepikatu *workflow* eta azpi-*workflow* bakoitzeko.
5. *Workflow* motorraren bidez, klase eta instantzia guztiekin sare bat eratu, *workflow* guztia korritu eta informazioa datu-base erlazionalean gordeko du.

### 11.2.4 Diseinua

Atal honetan, azpisistema honek izan dituen diseinuak deskribatuko dira.

#### 11.2.4.1 Datu-base erlazionala

Jarraian, datu-base erlazionalaren diseinua (11.4 irudia) eta taula bakoitzaren deskribapena ikusten da:



11.4 irudia: Datu-base erlazionalaren diseinua (I)

- **methodologies (Metodologiak):** taula honetan aukeratu den metodologiaren izena eta *workflow*-ereduaren bertsioa adierazten da.
- **phases (Faseak):** metodologiak definitzen duen bizi-zikloaren faseen izenak gordetzen ditu.
- **activities (Aktibitateak):** faseen barruan dauden aktibitateen izenak gordetzen ditu.
- **tasks (Jarduerak):** aktibitateen barruan dauden jardueren izenak eta jardueri dagokien rolak gordetzen ditu.
- **roles (Rolak):** metodologiak erabiltzen dituen rolen izenak gordetzen ditu.
- **artifacts (Artefaktuak):** metodologiaren bitartez bete behar diren artefaktuen izenak gordetzen ditu.
- **artifact\_sections (Artefaktuen sekzioak):** artefaktuak duten sekzioen zenbakiak, izenak eta izan ahal duten gida edo laguntza-testuen kokalekuak gordetzen ditu.
- **projects\_artifacts** eta **projects** taulak ez dira azpisistema honetan erabiltzen.

Taula guztiek identifikatzaile numerikoa dute, kontsultak egiterako orduan transakzio-denborak murrizteko eta idazkera akatsak saihesteko.

#### 11.2.5 Implementazioa

Atal honetan, *Workflow Editor* azpisistema implementatzeko behar izan diren elementu esanguratsuenak agertzen dira.

#### 11.2.5.1 *Workflow*-lengoaia

Testu-editorean sortzen diren *workflow*-ereduak motorrak exekutatu ahal izateko eta bizi-zikloa definitzen duten metodologiaren ezaugarriak izateko, *DOT* lengoaiaren arauak errespetatzen dituen lengoaia propio bat asmatu da.

Definitu den *workflow*-lengoaia “*Workflow-lengoaiaren Eskuliburua*” dokumentuan idatzi da.

#### 11.2.5.2 Eraldaketa-prozesua

*Graphviz*, *Gephi* eta *Protégé* softwareak erabilita, *DOT* lengoia idatzitako *workflow*-eredua *COOL* lengoiaiko klase eta instantzietan eraldatzea du helburu.

*Eraldaketa*-prozesu hori “*Workflow Editor - Eskuliburua*” dokumentuan zehatz-mehatz deskribatzen da.

Laburpen moduan, hurrengo da egin beharreko eraldaketa-prozesua:

1. *DOT* lengoia idatzitako eredua *graphml* formatura eraldatzea, *Gephi* softwarea erabiliz.
2. *Graphml* formatuan eraldatutako eredu hori *Protégé* v3.5 softwarearen bitartez klase eta instantzietan bihurtzea.
3. Sortutako klase eta instantziak EHSIS inferentzia motorrarekin exekutagarriak izateko egokitu.

#### 11.2.5.3 *OpenUP* metodologiaren bizi-zikloko *workflow*-eredua

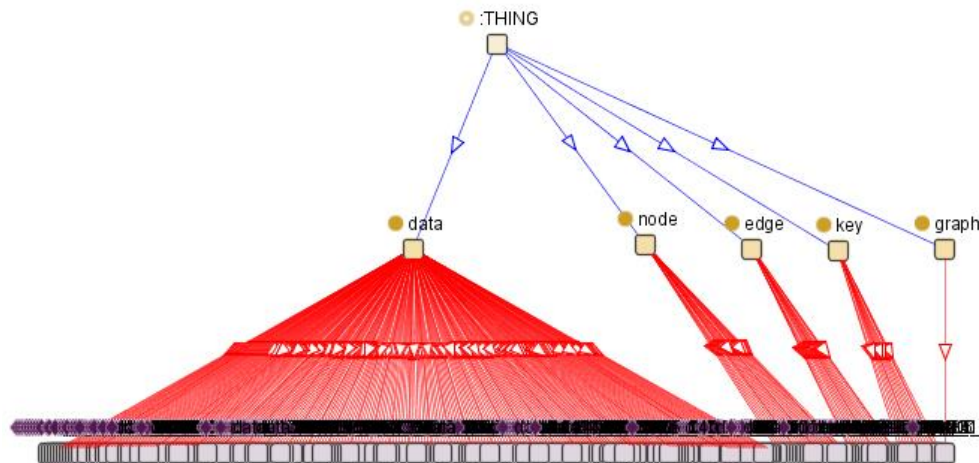
Aurretik komentatutako *workflow*-lengoaia baliatuz, *OpenUP* metodologiaren bizi-zikloaren *workflow*-eredua implementatu da. Lehenengo fase osoa eta hurrengoaren hasiera implementatzea erabaki zen (*inception* eta *elaboration*), beraz bizi-zikloaren %30a inguru implementatu da.

Guztira 71 *workflow* eta azpi-*workflow* daude, beraien erpin eta ertzen informazioa 450 objektuetan biltzen da. Orokorrean, *RUP* bezalako metodologia ahaltsuago baten kasuan, 5000 objektuetara igo daiteke. *Workflow* osoaren irudia erraz partekatu ahal izateko, ereduak nabigarriak dira.

#### 11.2.5.4 *Workflowa* exekutatzeko kodea

##### Klaseak

*Workflowak* *graphml* formatura bihurtu direnez, formatu honentzako definitutako klaseak berrerabili daitezke, bost baino ez dira (ikus 11.5 irudia): *graph* (grafo), *data* (datu), *node* (nodo), *edge* (ertz) eta *key* (kode).



11.5 irudia: Graphml formatuak definitutako klaseak eta grafo nagusiko instantziak

### Instantziak

Workflow baten informazio edo ezagutza guztia adierazteko aurreko klaseen instantzia asko behar dira, hala ere, automatikoki sortu daitezke.

### Erregelak

Workflow edo azpi-workflow bat hasieratik bukaerara exekutatzeko erregela kopuru txiki batekin nahikoa da eta beraien baldintza aldean idatzi beharreko kode patroia berrerabilgarria da.

Hona hemen erregela bat definitzeko erabili den kode zatia, denetan errepikatzen den patroia:

```
(defrule AztertuGrafoaErpinBatetikHasita
  (declare (salience 31))
  ?O<-(GrafoBereanJarraitu ?oraingoErpina ?proiektua) ;; lehentasuna
  ?erp<-(object (is-a node) ;;pausua sakoneran
    (name ?erpinObjektua) ;; erpin bat
    (_id ?erpinIzena) )
    (test(eq ?oraingoErpina ?erpinIzena)) ;; erpin identifikatzailea
    (not (Aztertuta ?proiektua ?oraingoErpina ?bisita)) ;; oraingo pausukoa?
    ?ertz<-(object (is-a edge) ;;ertza
      (_source ?erpinIzena) ;; oraingotik ateratzen dena?
      (_target ?hurrengoErpina)) ;; helburua
=>
(bind ?*BisitaOrdena* (+ ?*BisitaOrdena* 1))
(assert (Aztertuta ?proiektua ?erpinIzena ?*BisitaOrdena*))
```

#### 11.2.5.5 Workflow motorra

EHSIS\_RT inferentzia motorraren bitartez *workflow* eta azpi-*workflow* guztiak korritzeko, *DFS* (*Depth First Search - Sakonerako Bilaketa*) grafoen bilaketa algoritmoa erabili da. Azken finean, *workflowek* grafoen egitura berdina dute, nodo eta ertzez osatuta daude, beraz grafoen bilaketa algoritmoak aplikatu daitezke.

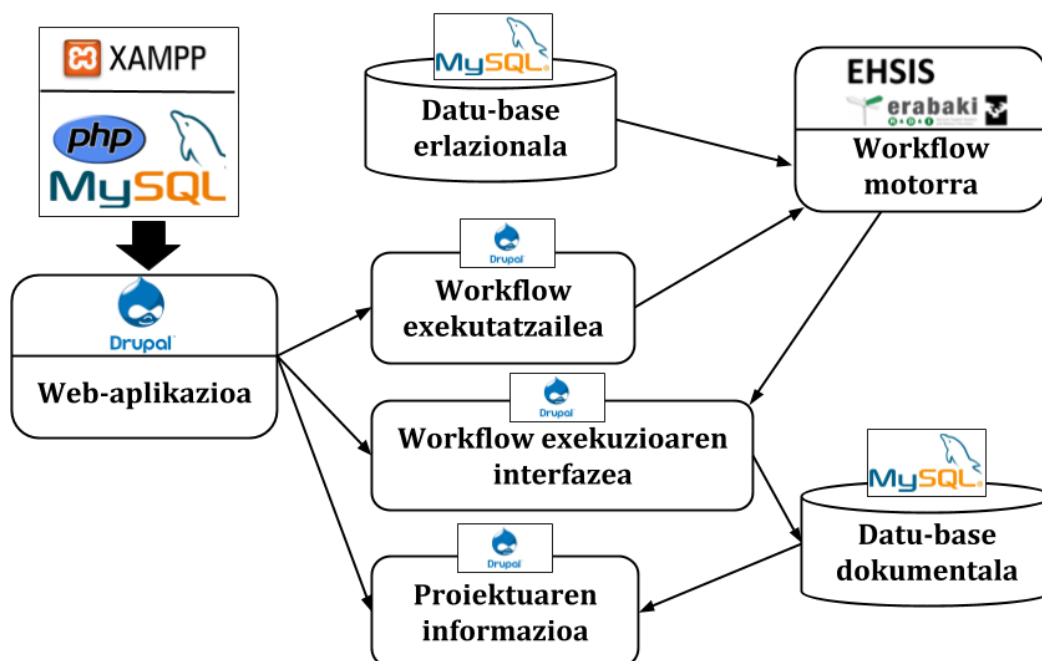
Gainera, motorrak *workflow*aren klase, instantzia eta erregela guztiak arinago prozesatu dezan, *CLIP*Sek inplementatzen duen *RETE* algoritmoan<sup>20</sup> oinarritzkoa den *RETE* sarean egokitzen da informazio guztia eta fitxategi bakar batean gorde edo “konpilatu” daiteke. *RETE* algoritmoa objektu eta erregela askorekin ere eraginkorra da, errendimendu hori lortzeko memoria gehiago erabiltzen du. Memoria eta abiaduraren oreka hobetzen duten inplementazio komertzialak garestiak dira.

Hurrengo komandoaren bidez, *workflow*-ereduko instantzia guztien informazioa datu-base erlazionalean gordeko da:

```
EHSIS_RT.exe -b _DFS_SakonerakoBilaketaGrafoan_bin-exe.bch
```

## 11.3 ProWF: IO-System

### 11.3.1 Arkitektura



11.6 irudia: IO-System moduluaren arkitektura

Jarraian, arkitekturaren osagai eta elementu guztiak gainetik deskribatuko dira:

- **Web-aplikazioa:** CMS baten bitartez, lokalean erabiltzeko (zerbitzari batean ere kokatu daiteke) pentsatuta dagoen sistema kudeatzailea. Hurrengo osagai nagusiak integratuko ditu:

<sup>20</sup> *RETE* algoritmoa: arauak ekoizten dituzten sistemak inplementatzeko patroiak bilatzen dituen algoritmo eraginkorra da. Gaur egun, izen handiko sistema aditu askoren oinarria da, *CLIPS* barne.

- **Workflow exekutatzailea:** *workflow* motorrari aginduak emango dizkio eta ondoren, bere irteerak jasoko ditu.
- **Workflow exekuzioaren interfazea:** erabiltzaileak exekuzioan zehar ikusiko duena da, *workflow* motorraren irteeran adierazitako ekintzak betetzeko interfazea da.
- **Proiektuaren informazioa:** *CCII-2016N-02* araua jarraituz, erabiltzailearen dokumentuak antolatuta dituen webgune antzeko bat da. Informazioa datu-base dokumentaletik eskuratuko du.
- **Workflow motorra:** Datu-base erlazionalari kontsultak eginez, *workflow*-eredua prozesatuko du eta jardueraz-jarduera joango da egin beharreko ekintzak irteera lez bueltatuz.
- **Datu-base erlazionala:** sortutako *workflow*-ereduen informazioa gordetzeaz aparte, erabiltzaileen proiektu eta artefaktuen informazioa biltegitratuko du.
- **Datu-base dokumentala:** erabiltzaileek idatzitako dokumentuen kokapena eta sekzio bakoitzaren edukia gordeko du.

### 11.3.2 Erabilitako teknologiak

Atal honetan, *IO-System* azpisistemak behar dituen teknologiak zerrendatuko dira.

#### XAMPP → Apache, MySQL

Web-aplikazioak datu-base dokumentala eta erlazionalaren atzipena eta *Workflow* motorrak datu-base erlazionalaren atzipena izateko erabiltzen da, *MySQL*ren bitartez. Gainera, web-aplikazioak *PHP* programazio lengoia eta *MySQL* behar du funtzionatzeko.

#### Drupal 8

Web-aplikazioa sortu eta kudeatzeko erabili den *CMSa*. XAMPP behar du funtzionatzeko. Moduluen bitartez inplementatutako funtzionalitateez gain, erabiltzaileen kudeaketarako eta datu-baseen kontsulta dinamikoak egiteko erabiltzen da ere.

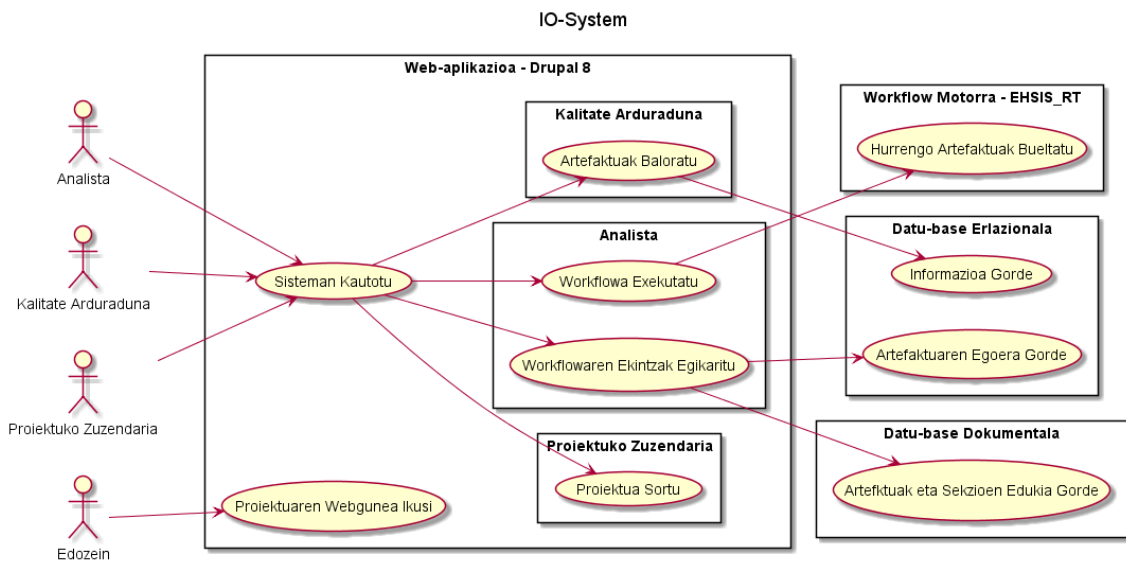
Web-aplikazioak lokalean funtzionatzen du bakarrik, baina zerbitzari batera eraman daiteke.

#### EHSIS\_RT

EHSIS motorraren produkzio bertsio da. CLIPS lengoaiara eraldatutako *workflow*-ereduaren klaseak eta instantziak baliatuz, erregelen bitartez eta datu-base erlazionalari kontsultak eginez, prozesua urratsez urrats exekutatzeko erabiltzen da.

### 11.3.3 Analisia

*IO-System* azpisistemari dagokion erabilpen-kasuen diagrama 11.7 irudian ikus daiteke.



11.7 irudia: IO-System azpisistemaren erabilpen-kasuen diagrama. PlantUML.

### 11.3.3.1 Proiektua sortu

**Deskribapena:** Erabilpen-kasu honek proiektu bat sortzea eta datu-base erlazionalan gordetzea du helburu.

**Aktorea:** Proiektuko Zuzendaria

**Aurrebaldintzak:**

- Erabiltzailea web-aplikazioan erregistratuta egon behar da, proiektuko zuzendariaren rolaekin.
- *Workflow Editor* azpisistemaren bitartez *workflow*-eredu bat sortuta egon behar du, eta bere informazioa datu-base erlazionalan egon behar du gordeta.
- Gutxienez, analista eta kalitate arduradunaren rola duten erabiltzaile bat egongo behar da web-aplikazioan erregistratuta.

**Postbaldintzak:**

- Proiektua sortuta egongo da eta bere informazio guztia datu-base erlazionalan gordeta.

**Gertakarien oinarrizko fluxua:**

1. Proiektuko zuzendaria sisteman kautotuko da.
2. Proiektua sortuko du, proiektuaren rola esleituz, metodologia aukeratuz eta bestelako informazio gehigarria idatziz.
3. Proiektuaren informazioa datu-base erlazionalan gordetzen da.

### 11.3.3.2 Workflowa exekutatu

**Deskribapena:** Erabilpen-kasu honetan, egungo proiektuan aukeratutako metodologiaren *workflowa* exekutatu da, egungo jardueraren artefaktuak edo ekintzak bueltatuz.

**Aktorea:** Analista

**Aurrebaldintzak:**

- Erabiltzailea web-aplikazioan erregistratuta egon behar da, analistaren rolaekin.
- Analistaren erabiltzailea proiektuan esleituta egon behar izan da.

**Postbaldintzak:**

- Inferentzia motorrak CSV formatuko fitxategi bat bueltatuko du, proiektuaren egungo jardueraren ekintza eta artefaktuekin.

**Gertakarien oinarrizko fluxua:**

1. Analista sisteman kautotuko da.
2. *Workflowa* exekutatu du.
3. *Workflow* motorrak hurrengo jarduerari dagokion artefaktu eta ekintzak bueltatuko ditu CSV formatuko fitxategi baten.

### 11.3.3.3 Workflowaren ekintzak egikaritu

**Deskribapena:** Erabilpen-kasu honetan, aurreko erabilpen-kasuan jasotako CSV fitxategia irakurri eta dagokion ekintzak egikaritu eta aldaketak datu-base dokumental zein erlazionalean gordeko dira.

**Aktorea:** Analista

**Aurrebaldintzak:**

- Erabiltzailea web-aplikazioan erregistratuta egon behar da, analistaren rolaekin.
- Analistaren erabiltzailea proiektuan esleituta egon behar izan da.
- *Workflowa* gutxienez lehenengo aldiz exekutatu behar izan da.

**Postbaldintzak:**

- Egikaritutako artefaktuaren egoera datu-base erlazionalean gordeta egongo da.
- Artefaktua eta bere sekzioen edukiak datu-base dokumentalean gordeta egongo dira.

**Gertakarien oinarrizko fluxua:**

1. Analista sisteman kautotuko da.
2. Egungo jardueran egiteke dauden artefaktuetatik bat aukeratu.



3. Artefaktuari dagokion rola duen taldekideari jakinarazi eta ekintza egikaritu. Hemen hiru aukera daude:
  - a. Artefaktuaren sekzioa idatzi behar da.
  - b. Artefaktuaren sekzioa idatzi behar da eta txantiloia edo laguntza-testua du.
  - c. Irudi, *UML* diagrama edo graforen bat igo behar da.
4. Artefaktuari sartutako ordu kopuruak, bere bertsioa eta artefaktua beten duenaren nota jarri.
5. Artefaktuaren egoera datu-base erlazionalean gordeko da.
6. Artefaktua eta bere sekzioak datu-base dokumentalean gordeko dira.

#### 11.3.3.4 Artefaktuak baloratu

**Deskribapena:** Erabilpen-kasu honen helburua, *workflowaren* exekuzioan zehar betetzen diren artefaktuak balioztatzea da, haien kalitatea bermatzea.

**Aktorea:** Kalitate Arduraduna

**Aurrebaldintzak:**

- Erabiltzailea web-aplikazioan erregistratuta egon behar da, kalitate arduradunaren rolarekin.
- Kalitate arduradunaren erabiltzailea proiektuan esleituta egon behar izan da.
- *Workflowa* gutxienez lehenengo aldiz exekutatu behar izan da.

**Postbaldintzak:**

- Artefaktua balioztatuta egongo da eta balorazioa datu-base erlazionalean gordeko da.

**Gertakarien oinarrizko fluxua:**

1. Kalitate arduraduna sisteman kautotuko da.
2. Egungo jardueran balioztatzeke dauden artefaktuetatik bat aukeratu.
3. Artefaktuaren kalitatea aztertu eta nota eman.
4. Balorazioa datu-base erlazionalean gordeko da.
5. Artefaktua balioztatzeke dauden artefaktuen zerrendatik kenduko da.

#### 11.3.3.5 Proiektuaren webgunea ikusi

**Deskribapena:** Erabilpen-kasu honetan, aukeratzen den proiektuaren informazioa eta bere dokumentuak, *CCII-2016N-02* araua jarraitzen duen webgune antzeko baten antolatuta ikusiko dira.

**Aktorea:** Edozein rol, erregistratuta ez dauden erabiltzaileak ere.

**Aurrebaldintzak:** Ez dago aurrebaldintzarik.

**Postbaldintzak:** Ez dago postbaldintzarik.

Gertakarien oinarrizko fluxua:

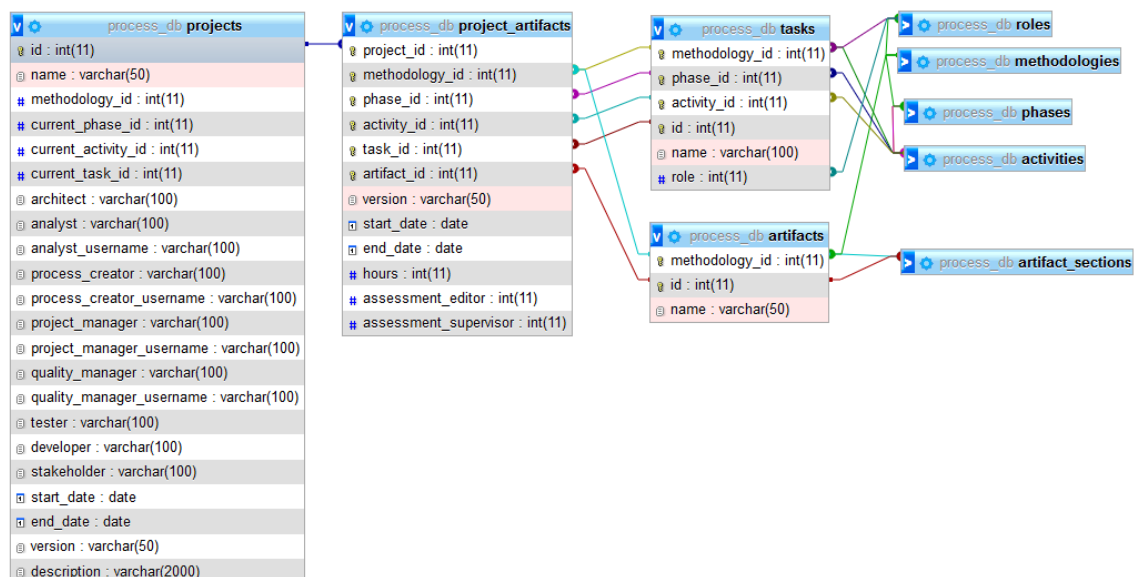
1. Erabiltzaileak ikusi nahi duen proiektua aukeratzen du.
2. Proiektuaren dokumentuak/artefaktuak eta bere informazioa ikusten du, CCII-2016N-02 araua betetzen duen webgune antzeko baten.

### 11.3.4 Diseinua

Atal honetan, azpisistema honek izan dituen diseinuak deskribatuko dira.

#### 11.3.4.1 Datu-base erlazionala

Jarraian, datu-base erlazionalaren diseinua (11.8 irudia) eta taula bakoitzaren deskribapena ikusten da:



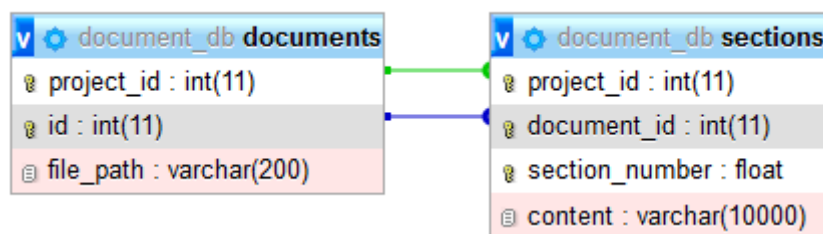
11.8 irudia: Datu-base erlazionalaren diseinua (II)

- methodologies, phases, activities, tasks, roles, artifacts eta artifact\_sections: Workflow Editor azpisistemaren diseinu atalean azaldu dira.
- projects (Proiektuak): erabiltzaileek sortzen duten proiektuen beharrezko informazio guztia gordetzen du. Hurrengo informazioa gordetzen du:
  - Proiektuaren izena.
  - Erabilitako metodologia.
  - Proiektuko zuzendariaren, analistaren eta kalitate arduradunaren erabiltzaileen izenak.

- Proiektuko zuzendariaren, analistaren, kalitate arduradunaren, prozesu sortzailearen, arkitektoaren, garatzailearen, probatzailearen eta interesdunaren izenak.
- Proiektuaren hasiera eta bukaera datak.
- Proiektuaren bertsioa. Adibidez: Pre-Alpha, Alpha, Beta...
- Proiektuaren deskribapena.
- Egungo fase, aktibitate eta jardueraren identifikatzaileak.
- **project\_artifacts (Proiektuaren artefaktuak):** proiektuan bete diren artefaktuen egoera biltegitatzeko erabiltzen da. Hurrengo informazioa gordetzen du:
  - Proiektu eta metodologiaren identifikatzailea.
  - Artefaktuaren identifikatzailea.
  - Artefaktua zein fase, aktibitate eta jardueran garatu den.
  - Artefaktuaren hasiera eta bukaera datak, egungo fase, aktibitate eta jardueran.
  - Artefaktuaren bertsioa, egungo fase, aktibitate eta jardueran. Adibidez: Pre-Alpha, Alpha, Beta...
  - Artefaktuari sartutako ordu kopurua, egungo fase, aktibitate eta jardueran.
  - Artefaktua bete duen arduradunaren nota (0-10), egungo fase, aktibitate eta jardueran. Betetze-maila adierazteko.
  - Kalitate arduradunaren nota (0-10), egungo fase, aktibitate eta jardueran.

#### 11.3.4.2 Datu-base dokumentala

Jarraian, datu-base dokumentalaren diseinua (11.9 irudia) eta taula bakoitzaren deskribapena ikusten da:



11.9 irudia: Datu-base dokumentalaren diseinua

- **documents (Dokumentuak/Artefaktuak):** proiektuaren eta artefaktuaren identifikatzaileak eta dokumentuaren kokapena gordetzen du.
- **sections (Sekzioak):** proiektuaren, artefaktuaren eta sekzioaren identifikatzaileak eta sekzioaren edukia gordetzen du.

### 11.3.5 Implementazioa

Atal honetan, *IO-System* azpisistema inplementatzeko behar izan diren elementu esanguratsuenak agertzen dira.

#### 11.3.5.1 Drupal 8 eta eduki modularra

*Drupalen* guneko edukia datu-base batean biltegitzen denez, komenigarria da edukia modularra izatea; horrek esan nahi du, gunean orrialde oso gisa editatu beharrean, guneko zenbait orrialde automatikoki sortzen direla beste eduki-elementu edo modulu batzuetatik abiatuta.

Hori dela eta, behar izan diren funtzionalitate bakoitzetik modulu bat edo bi inplementatu dira. Era berean, web-aplikazioaren beharretara egokitzen den moduluren bat aurkitu da ere, *Drupalen* webgunean eskuratutakoa, eta ondoren, sisteman integratu da.

#### 11.3.5.2 Workflowaren exekuzioa eta EHSIS\_RT inferentzia motorra

*Workflow Editor* azpisistema erabiliz sorutako *workflow*-eredua exekutatzeko EHSIS\_RT motorra erbili denez, web-aplikazioak hurrengo komandoa exekutatu behar du:

```
EHSIS_RT.exe -b _DFS_Hurrengo_Artefaktuak_Sakoneran_bin-exe.bch
```

Komando hori exekutatzean, datu-base erlazionalean dauden proiektu bakoitzaren CSV formatuko fitxategi bat bueltatuko da, proiektuaren egungo fase, aktibitate eta jarduerak kontuan hartuz (datu-baseko *current\_phase*, *current\_activity*, *current\_task*). Hori guztia CLIPS lengoaiako erregelen bidez inplementatu da.

*Workflowa* exekutatzeko, *workflow* eta azpi-*workflow* guztiak korritu behar dira ere, *DFS* (*Depth First Search* - *Sakonerako Bilaketa*) grafoen bilaketa algoritmoa berriro erabiliz. Sakonerako bilaketa grafo konexu bakoitzean erabiltzen da eta azpi-grafoetara heltzeko ere modu berean jotzen da.

Gainera, motorrak *workflowaren* klase eta instantzia guztiak arinago prozesatu dezan, aurretiaz instantzia eta erregela guztiekin RETE sarea eratu eta fitxategi "konpilatu" bezala gorde behar da. Alde honen kodeari begiratuta, helburuetan jarritako erronka kontutan hartuta, mantengarria eta hobekuntza berriagoen oinarria izatea ezinbestekoa da. Klaseak berrerabiltzea, objektuak automatikoki sortzea eta erregelak, gutxi izateaz gain, patroik berekoak eta berrerabilgarriak izateak kodea automatikoki sortzeko aukera handiak ematen ditu, programazio tradizionalak baino gehiago.

## 11.4 Sistemaren abantailak eta desabantailak

Sistemaren ezaugarriak nabarmentzeko, bereziki sistemaren funtzionalitate nagusiei erreparatuz, planteatutako soluzioaren abantaila eta desabantaila batzuk azaltzen atal honetan.

#### 11.4.1 Abantailak

Sistema honen 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 CMS*ari esker, erabiltzaileen erregistro eta kudeaketa erraza du.
- *Workflow* motorra, *workflow*-lengoaia 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.

#### 11.4.2 Desabantailak

Soluzioaren desabantailei dagokienez, hurrengoak aurkitu dira:

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

## 11.5 Etorkizunerako hobekuntzak

Proposatutako irtenbidea, *ProWF* sistema, prototipo bat denez, oraindik bere kalitate eta fintze maila igo daiteke. Prototipoaren bidez lortutako emaitzak oinarri lez hartuta, sistemaren eta proiektuaren 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>21</sup> motako testu-editore bat implementatzea. Softwarearen bizi-zikloa definitzen duten metodologia askotan taulak eta Excel orriak bete behar dira, prototipo honetan, ordea, ez dago taulak txertatzeko aukerarik.
- 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.

---

<sup>21</sup> WYSIWYG: sigla (ingelesez), What You See Is What You Get. Testu-prozesadoreei eta beste testu-editore batzuei aplikatutako esaldi bat da, azkenengo emaitza zuzenean erakutsiz dokumentu bat idazteko aukera ematen duena.

## 11.6 Ondorioak

Aurretik ikusitako guztiaren ondorioz eta kapitulu honetan azaldutakoarekin, jarraian proiektu honi buruzko ondorio batzuk azalduko dira.

Oro har, esan daiteke sistemak proiektu honen helburuak bete dituela, hasieran ezarritako betekizunak bete dira ere. Beraz, denbora gehiago, proiektuko partaide eta finketa-prozesu egokiekin, produktu bideragarri baten bihur daiteke, gainera, aurreko atalean komentatutako hobekuntzekin, sistemaren kalitateari dagokionez, maila bat igoko luke.

Proiektuaren hasierako hipotesi eta betekizunetatik abiatuz, hurrengo ondorioak atera dira:

- **Inferentzia motorraren erabilerak azkartasuna ematen du prozesuaren exekuzioan, kode ahaltsu eta erraza erabiliz.** Prozesuaren edo lan-fluxuaren lengoaia grafikoa aldatu ezean, prozesuaren exekuziorako erregelak berrerabili daitezkeen patroien bidez definitu daitezke. *DFS* grafoen sakonerako bilaketa algoritmoa eta *RETE* algoritmoaren erabilerarekin, prozesaketa-denbora laburrak lortu dira. Esan beharra dago, sortutako *workflow*-eredua (*OpenUP* metodologiaren bizi-zikloa) ez dagoela osatuta, beraz, *workflow* eta azpi-*workflow* gehiago gehituz, prozesaketa-denbora handitu daiteke, baina ez da espero asko handitzea.
- **Definitutako garapen prozesuak proiektuen elaborazioa kontrolatzen du.** Prozesua aldatzeak sistemaren portaera eta datu-basea automatikoki aldatzea ekarriko du. Beraz, sortzen diren bizi-zikloen *workflow*-ereduek pisu handiena dute sisteman, haien arabera funtzionatuko du.
- **CMS baten erabilera datuen sarrera/irteerarako irtenbide egokiena da.** Kanpoko baliabiderik erabili gabe, kudeaketa, administrazioa eta mantentze-lanak egiteko laguntza ematen duen web-aplikazioa sortu da, itxura profesionalarekin.
- **Datu-base erlazionalak prozesu edo lan-fluxu baten ezagutza gordetzeko modurik egokiena izan da.** Garatutako sistemari atomikotasuna, datuen independentzia, emaitzen arteko koherentzia eta produktibitatea eman dizkio. Gainera, oso lagungarria izan da inferentzia motorraren erregelak eta sarrera/irteera sistemaren funtzionalitateak inplementatzeko.
- **Estandarretan oinarritutako garapenak sistemaren mantenua eta hedapena errazten ditu.** Sistema iteratiboki hobetzen joan denez, bere mantenua eta hedapena ez da proiektuaren bukaerarako utzi, oster, proiektu osoan zehar egon da prozesu horren barruan.





## 12. Proposatutako Sistemaren Arrisku Analisia

Kapitulu honetan, proiektuaren hurrengo faseetara begira, sortu daitezkeen arriskuen analisia egin da. Arriskuak harrapatzeko OpenUP metodologiako “Arriskuen Zerrenda” dokumentuan oinarritu gara, modu horretan, arriskuak identifikatu, bere larritasuna neurtu eta arriskua arintzeko estrategiak bilatu dira.

### 12.1 Arriskuen identifikazioa

Hurrengo taulan, proiektuaren elaborazioan zehar agertu daitezkeen arriskuak identifikatu eta zerrendatu dira, bere identifikazioa kodea, izena eta deskribapena adieraziz:

Arriskuaren IDa	Arriskua	Deskribapena
SR01	<i>Drupal CMSaren eta ProWF sistemaren bilakaera bideragarria ez izatea.</i>	<i>Drupal CMSaren funtzionalitateak ez izatea ProWF sistemak behar dituen bestekoak.</i>
SR02	Prozesuak definitzeko lengoia grafikoak hobekuntza gehiegi behar izatea.	Prozesuak definitzeko workflow-lengoia inplementatzen dituen baliabideak prozesu zehatzagoak definitzeko nahikoa ez izatea.
SR03	<i>Clips</i> dohainezko inferentzia motorra ordezkatu beharra.	<i>Clips</i> dohainezko inferentzia motorra produktu komertzial batekin ordezkatu beharra.
SR04	<i>MySQL</i> ordezkatu behar izatea.	Beste datu-base kudeatzaile batera migratu behar izatea.

12.1 taula: Arriskuen identifikazio-zerrenda

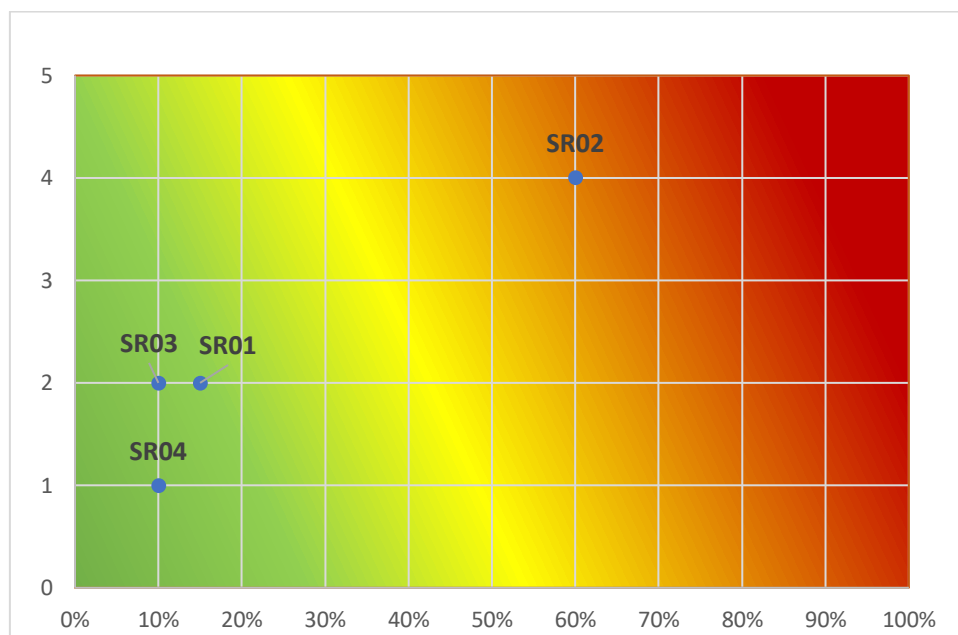
### 12.2 Arriskuen analisi kuantitatibo eta kualitatiboa

Hurrengo taulan aurreko atalean identifikatutako arriskuen tamaina edo larritasuna neurtuko da, horretarako arriskuaren inpaktua eta bera agertzeko probabilitatearen arteko biderketa egingo da:

Arriskuaren IDa	Inpaktua (1-5)	Probabilitatea (%)	Larritasuna
SR01	2	%15	$2 * 0.15 = 0.3$
SR02	4	%60	$4 * 0.6 = 2.4$
SR03	2	%10	$2 * 0.1 = 0.2$
SR04	1	%10	$1 * 0.1 = 0.1$

12.2 taula: Arriskuen identifikazio-zerrenda

Jarraian arriskuen larritasuna adierazten duen grafikoa ikusi daiteke (ikus 12.1 irudia), X ardatzean arriskua agertzeko probabilitatea eta Y ardatzean arriskuaren inpaktua adierazten da. Kolore berdea larritasun gutxiko arriskua dela adierazteko erabili da, horia neutroa eta gorria larritasun handia adierazteko.



12.1 irudia: arriskuen larritasuna adierazten duen grafikoa

## 12.3 Arriskuak arintzeko estrategiak

Aurreko analisi kuantitatibo eta kualitatiboa kontuan hartuz, hurrengo taulan larritasun handiko arriskuei erantzuteko plangintza deskribatuko da:



Arriskuaren IDa	Larritasuna	Arriskua arintzeko estrategia
SR02	2.4	Metodologia baten ezarpena lortzeko denbora urteetakoa izan daiteke, software garatzaileen praktika onak ere beraien bikaera izan behar dutelako. Arriskuak gutxitzeko lengoaiaren sinpletasuna mantendu eta bere interpretazioa Inferentzia motorraren erregela sinpleetan inplementatu behar da.

12.3 taula: Arriskuak arintzeko estrategiak



## 13. Proposatutako Sistemaren Antolaketa eta Kudeaketa

---

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

### 13.1 Proiektuaren antolaketa

Proiektuaren antolaketa *OpenUP* metodologiaren bitartez definitu da, beraz gomendatzen da eranskinetako “Proiektu Plana” dokumentua irakurtzea ikuspegi zabalagoa izateko.

#### 13.1.1 Taldekideak

- Proiektuko zuzendaria
- Analista
- Arkitektoa
- Garatzailea
- *Testerra*
- Interesduna

#### 13.1.2 Ardurak eta betekizunak

- Proiektuko zuzendaria: proiektuaren plangintza zuzentzen du, alderdi interesdunekiko elkarrekintzak koordinatzen ditu eta proiektuaren helburuak betetzea ziurtatzen dio lantaldeari.
- Analista: bezero eta erabiltzaileen kezkak adierazten ditu; horretarako, alderdi interesdunen ekarpenak biltzen ditu, konponduko den arazoa ulertzeko, eskakizunak atzemateko eta lehentasunak ezartzeko.
- Arkitektoa: softwarearen arkitektura definitzeaz arduratzen da, proiektuaren diseinu orokorra eta implementazioa mugatzen duten funtsezko erabaki teknikoak hartzeaz gain.
- Garatzailea: sistemaren zati bat garatzeaz arduratzen da, arkitekturara egokitze diseinua barne. Erabiltzaile-interfazearen arteko elkarrekintza gauzatzeko prototipoak sortzeko eta, ondoren, soluzioa osatzen duten osagaiak inplementatu, banan-banan probatu eta integratzen ditu.
- *Testerra*: sistemaren proba jarduera nagusien arduraduna da. Jarduera horien artean probak identifikatzea, definitzea, inplementatzea, proben emaitzak erregistratzea eta konponbidearen parte diren emaitzak aztertzea daude.

- Interesduna: proiektuak asebate behar dituen interes-taldeak dira. Proiektuaren emaitzak materialki eragiten dion (edo eragin diezaiokeen) edozein pertsonak bete dezake eginkizun hori.

### 13.1.3 Lan-ingurunea

*OpenUP* eta proiektuaren webgunea erabilita lan egingo da.

### 13.1.4 Informazio-sistema

Proiektu honen dokumentuak eta fitxategiak zuzendariaren makinan gordetzea eta segurtasun-kopia moduan zerbitzari propio bat erabili daiteke. Zuzendariak eta bere taldekideek bakarrik izango dute zerbitzarirako sarbidea.

### 13.1.5 Komunikazio-kanalak

Proiektuko taldekideen arteko harremanak normalean bileren bidez gauzatuko dira. Arazo edo beste konturen bat izatekotan posta elektronikoaren bitartez komunikatuko da. Interesdunekin komunikatzeko, ordea, posta elektronikoa lehenetsiko da, baina bilera birtualak eta presentzialak egin daitezke ere.

Taldekideen arteko fitxategien transferentziak egiteko zerbitzaria erabiltzea zehaztu da.

## 13.2 Proiektuaren kudeaketa

Kudeaketa, antolaketaren moduan, *OpenUP* metodologiaren bitartez egin da. Metodologia horrek aurretik finkatutako helburuak dituzten iterazio batzuk ezartzea proposatzen du. Iterazioak hasieran definitu edo proiektuan zehar gehitu daitezke.

Iterazio bakoitzaren hasieran, proiektuaren kudeaketari buruzko hurrengo artefaktuak eguneratu behar dira:

- Proiektu plana (*Project Plan*): egungo iterazioan bete beharreko mugarriak eta helburuak gehitu.
- Iterazio plana (*Iteration Plan*): iterazioa deskribatu eta lan-atazak zerrendatu. Iterazioa amaitzean ebaluazioa egin.
- Lan-atazen zerrenda (*Work Items List*): iterazioan zehar egin beharreko jarduerak eta bere estimazioak gehitu, ondoren, iterazioa amaitzean sartutako orduak erregistratu.
- Arriskuen zerrenda (*Risk List*): iterazioan identifikatutako arriskuak gehitu.

Hurrengo kapituluan proiektuko lan-atazak eta iterazioak deskribatuko dira, bere denbora-estimazioen eta sartutako denboraren arteko alderaketa eginez.

## 14. Proposatutako Sistemaren Denboraren Planifikazioa

Kapitulu honetan, proposatutako sistemaren proiektua bideragarria izateko denbora-plangintza aurkeztuko da. Aurreko kapituluak aipatu den moduan, *OpenUP* metodologia jarraituz **iterazioen bidezko plangintza** definitu da.

### 14.1 Mugarri garrantzitsuak

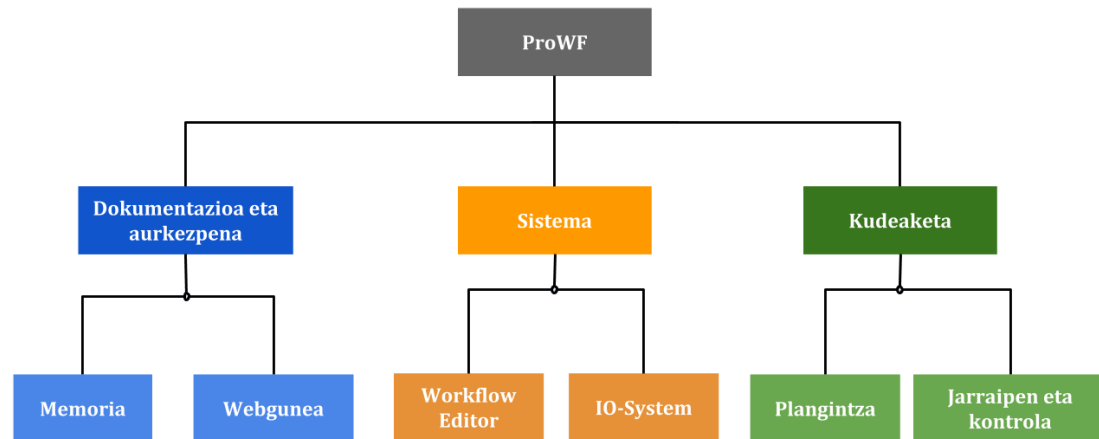
14.1 taulan proiektuaren mugarri garrantzitsuenak eta haien deskribapenak biltzen dira. Mugarrien data, proiektuaren taldekideak erabaki beharko dute.

Mugarria	Deskribapena
Data 1	Eraikuntzari hasiera eman.
Data 2	Proiektuaren dokumentazioa aztertu: Memoria eta webgunea.
Data 3	1. iterazioaren bukaera.
Data 4	2. iterazioaren bukaera.
Data 5	3. iterazioaren bukaera.
Data 6	Trantsizio fasearen bukaera.
Data 7	Proiektuaren entrega eta aurkezpena: memoria, eranskinak eta web gunea.
Data 8	Proiektuari buruzko aholkularitza.

14.1 taula: Proiektuko mugarri garrantzitsuak

### 14.2 Lan-ataza nagusien diagrama (LDE)

Jarraian, aurreko taularen lan-ataza nagusiak antolatu eta multzokatu dira:



14.1 irudia: Lan-ataza nagusien diagrama (WBS/LDE)

Azken mailan kokatutakoak proiektuko lan-pakete nagusiak dira, hona hemen lan-pakete horien deskribapenak eta bere azpian dituzten atazak:

- **Memoria (MM):** dokumentu hau eta bere eranskin guztiak.
  - **MM1:** memoriaren eranskinak idatzi.
  - **MM2:** memoria idatzi.
- **Webgunea (WG):** proiektuaren dokumentu guztiak biltzen dituen eta CCII-2016N-02 araua betetzen duen webgunea.
  - **WG1:** CCII-2016N-02 araua irakurri.
  - **WG2:** webgunea sortu eta hasierako dokumentuak jarri.
  - **WG3:** webgunea osatu.
- **Workflow Editor (WE):** workflow-lengoaia eta workflow editore hobetua.
  - **WE1:** workflow-lengoaia hobetu.
  - **WE2:** workflow-editorea inplementatu.
- **IO-System (IO):** sistemaren web-aplikazioa eta workflow motorra.
  - **IO1:** datu-base dokumentalaren diseinua hobetu.
  - **IO2:** datu-base erlazionalaren diseinua hobetu.
  - **IO3:** workflow motorra hobetu.
  - **IO4:** workflow exekutatzailerak hobetu.
  - **IO5:** workflow exekuzioaren interfazea hobetu.
  - **IO6:** Proiektuen informazioa erakusteko interfazea hobetu.
  - **IO7:** guztia integratu.



- **Plangintza (PL):** *OpenUP* metodologiaren bitartez jarraitu beharreko plangintza. Proiektuaren barne-elaborazioa.
  - **PL1:** proiektuaren plangintza egin.
- **Jarraipen eta kontrola (JK):** *OpenUP* metodologiaren bitartez egin beharreko jarraipen eta kontrola eta bilera-aktak. Proiektuaren barne-elaborazioa.
  - **JK1:** iterazioen kontrola eraman eta bilera-aktak idatzi.

## 14.3 Lan-atazen denbora estimazioa

Aurreko atalean definitutako paketeak eta atazak kontuan hartuz, 14.2 irudian agertzen den hilabeteko lanaren estimazioa egin da. Estimazio hau egiteko proiektu honek izan dituen desbiderapenak kontuan hartu dira.

	1. Hilabetea	2. Hilabetea	3. Hilabetea	4. Hilabetea	5. Hilabetea	6. Hilabetea	7. Hilabetea	8. Hilabetea
<b>Memoria (MM)</b>								
MM1								
MM2								
<b>Webgunea (WG)</b>								
WG1								
WG2								
WG3								
<b>Workflow Editor (WE)</b>								
WE1								
WE2								
<b>IO-System (IO)</b>								
IO1								
IO2								
IO3								
IO4								
IO5								
IO6								
IO7								
<b>Plangintza (PL)</b>								
PL1								
<b>Jarraipen eta kontrola (JK)</b>								
JK1								

14.2 irudia: hilabete bakoitzean egin beharreko lanaren estimazioa

Proiektua bideragarria izateko **lan osoa 1000 ordukoa** izan behar da. Hona hemen lan-pakete eta ataza bakoitzari estimatutako denbora (orduak):



<b>Memoria (MM)</b>	<b>200</b>
MM1	100
MM2	100
<b>Webgunea (WG)</b>	<b>20</b>
WG1	5
WG2	5
WG3	10
<b>Workflow Editor (WE)</b>	<b>190</b>
WE1	150
WE2	40
<b>IO-System (IO)</b>	<b>510</b>
IO1	10
IO2	10
IO3	160
IO4	30
IO5	100
IO6	60
IO7	140
<b>Plangintza (PL)</b>	<b>50</b>
PL1	50
<b>Jarraipen eta kontrola (JK)</b>	<b>30</b>
JK1	30
<b>GUZTIRA</b>	<b>1000</b>

14.2 taula: lan-ataza bakoitzari estimatutako ordu kopurua

Ataza bakoitzaren formakuntza-prozesua 14.2 taulan agertzen diren denboren barne dago. Esan daiteke formakuntza-prozesua agertzen den denbora bakoitzaren %30a zela. Ataza bakoitzaren denbora estimazioak zehazteko, proiektu honek izan dituen desbiderapenak kontuan hartu dira, desbiderapenak izan dituzten atazetan denbora gehiago edo gutxiago sartuz.

## 14.4 Iterazio-prozesua

Atal honetan, *OpenUP* metodologiaren bitartez jaso diren atazak iterazioetan zehar nolako banaketa izango duten deskribatuko da.

Lan guztia **bost iteraziotan banatu da**, proiektu honen moduan, bostak proiektuaren hasieran definitutakoak, proiektuaren garapenean zehar gehiago gehitu daitezke. Jarraian, iterazioak eta iterazio bakoitzaren helburuak deskribatzen dira:

ID	Iterazioa	Helburuak
I0	Hasiera	Lan-ingurunearen prestakuntza Proiektuaren webgunearen prestakuntza (WG1) Betekizunen ingeniariatza (MM1) Proiektu eta iterazio plana (PL1, JK1) Workflow-lengoaia hobetu (WE1)
I1	1.Iterazioa	Datu-base dokumentalaren diseinua hobetu (IO1) Proiektuaren webgunea sortu eta antolatu (WG2) Proiektuaren informazio erakusteko modulua hobetu (IO6) Workflow-lengoaia hobetu (WE1) Workflow editorea hobetu (WE2)
I2	2.Iterazioa	Datu-base erlazionalaren diseinua hobetu (IO2) Workflow exekuzioaren interfazea hobetu (IO5) Workflow motorrarekin hasi (IO3) Workflow-lengoaia hobetu (WE1)
I3	3.Iterazioa	Memoria idatzi (MM2) Proiektuaren webgunea osatu (WG3) Workflow-lengoaia hobetu (WE1) Workflow exekutatzailerak hobetu (IO4) Workflow motorra bukatu (IO3) Modulu guztiak integratu eta probak egin (IO7)
I4	Itxiera	Memoria egokitu eta guztiz zuzendu (MM2) Proiektuaren webgunea osatu (WG3)

14.3 taula: Proiektuko iterazioen banaketa eta bakoitzaren helburuak

*OpenUP* metodologiako “Lan-atazen Zerrenda” dokumentuari esker, proiektuko atazen jarraipen eta kontrola egitea posible da, iterazioen amaieran ataza bakoitzari emandako ordu kopuruak eta gelditzen zitzaien orduen estimazioa adieraziz.



## 15. Proposatutako Sistemaren Aurrekontua

---

Kapitulu honetan, proposatutako sistema bideragarria izateko izango duen kostu ekonomikoa agertuko da.

### 15.1 Orokortasunak

Aurrekontua sortzeko *ALI (Asociación de Titulados Universitarios Oficiales en Informática)* elkarteak banatutako irizpide batzuk jarraitu dira. Batetik, giza baliabideen barne-kostuk eta kanpo-kostuak ateratzea bere ordu kopuruekin batera eta 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. Bertan, software proiektu baten kide bakoitzari hurrengo ordainketa egitea proposatzen 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.

Jarraian, 15.1 taulan, proposatutako sistema bideragarria izateko aurrekontua ikus daiteke:

PARTIDA		PARAMETROAK			TOTALA (BEZik gabe)	TOTALA (BEZa barne)
Giza Baliabideak		Rola	Ordu kopurua	Barne-kostuak		
		Zuzendaria	200	100€	20.000	24.200
		Arkitektoa	150	70€	10.500	12.705
		Analista	200	70€	14.000	16.940
		Garatzailea	350	50€	17.500	21.175
		Testerra	100	50€	5.000	6.050
		Giza Baliabideak - Totala			67.000	81.070
Erremintak		Lizentzia	Mantenua (urtero)			
1	XAMPP	0	0	0	0	
2	Drupal 8	0	0	0	0	
3	PlantUML	0	0	0	0	
4	EHSIS	0	0	0	0	
5	GraphViz	0	0	0	0	
6	Gephi	0	0	0	0	
7	Protégé	0	0	0	0	
8	Notepad++	0	0	0	0	
Erremintak - Totala					0	0
TOTALA					67.000	81.070

15.1 taula: proiektuaren aurrekontua

Ondorioz, proposatutako sistema aurrera ateratzeko **81.070€** (BEZa barne) behar dira.

## 16. Oinarritzko Dokumentuen Ordena

---

Kapitulu honetan memoria eta proiektuko beste dokumentazio guztiaren arteko ordena ezarriko du, hau da, zeinek duen lehentasuna koherentzia faltan kasuan.

Dokumentazio luze honetan inkongruentziak 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, nik, egileak, idatzitako **memoria izango da kontuan hartu beharreko informazioa inkongruentzia eta koherentzia faltan kasuan**. Hau da, dokumentu hau.

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. Nire kontaktuak bertan jartzen ditut:

- [julenrojo12@gmail.com](mailto:julenrojo12@gmail.com)
- [jrojo013@ikasle.ehu.eus](mailto:jrojo013@ikasle.ehu.eus)





# ERANSKINAK

---

Memoriaren eranskin guztiak ez daude dokumentu honetan, tamaina handiko dokumentu bat atera ez dadin eta dokumentu guztiak ondo antolatuta izateko batzuk proiektuaren webgunean daude.

Proiektu honen egileak egin duen barne kudeaketa ikusi ahal izateko, memoria honen “**IV. Proiektuaren Barne Kudeaketa**” eranskinera jo.

Hori dela eta, memoriaren eranskinak, sistemaren espezifikazioa eta aurrekontua aurreko kapituluetan aipatu den webgunera igo dira:

<https://prowfgral.000webhostapp.com/index.htm>

## I. Memoriaren Eranskinak

### A1: Sarrerako dokumentazioa

- *Sarrerako dokumentazioa*

### A2: Analisi eta Diseinua

- Arkitektura:
  - *Arkitektura Koadernoa: Workflow Editor*
  - *Arkitektura Koadernoa: IO-System*
- Diseinua:
  - *Datu-base dokumentalaren diseinua*
  - *Datu-base erlazionalaren diseinua*

### A3: Tamaina eta esfortzu estimazioak

- *Tamaina eta esfortzu estimazioak*

### A4: Proiektuaren kudeaketa plana

- *Kudeaketa plana*

## A5: Segurtasun plana

- *Segurtasun plana*

## A6: Gainerakoak

- Hedapena:
  - *Workflow-lengoiaren Eskuliburua*
  - *Workflow Editor - Eskuliburua*
  - *IO-System - Eskuliburua*
- Garapena
  - *Eraikuntza*
- Ingurunea
  - *Tresnak*

## II. Sistemaren Espezifikazioa

- *Glosarioa*
- *Bisioa*
- *Betebeharren Espezifikazioa*
- *Erabilpen Kasuak*
- *Erabilpen Kasuen Ereduak*

## III. Aurrekontua

- *Aurrekontua*

## IV. Proiektuaren Barne Kudeaketa

- *Arriskuen Analisia*
- *Proiektuaren Antolaketa eta Kudeaketa*
- *Aurrekontuaren Laburpena*
- *Denboraren Planifikazioa*
- *Bilera-aktak (Proiektuaren webgunean)*

Atal honetako dokumentuak hurrengo orrialdeetan agertzen dira.

## IV.1. Arriskuen Analisia

Kapitulu honetan proiektuaren garapena hasi baino lehen eta proiektuaren garapenean zehar identifikatu diren arriskuen analisia egingo da. Arriskuak harrapatzeko *OpenUP* metodologiako “*Arriskuen Zerrenda*” dokumentua bete da, hemen arriskua identifikatu, bere larritasuna neurtu eta arriskua arintzeko estrategiak bilatu dira.

### IV.1.1 Arriskuen identifikazioa

Hurrengo taulan, proiektuaren elaborazioan zehar identifikatutako arriskuak zerrendatzen dira, bere identifikazioa kodea, identifikatutako data, izena eta deskribapena adieraziz:

Arriskuaren IDa	Identifikatu zen data	Arriskua	Deskribapena
R01	2020/03/03	Lan ingurunearen prestakuntza	Lan ingurunearen instalazioan eta ikaskuntza prozesuan agertu daitezken arazoak eta denbora galerak.
R02	2020/03/03	CMSaren aldaketa	CMSren bitartez garatuko den sarrera/irteera web-aplikazioa aukeratu denean, garapenaren bitartean CMSez aldatzea denbora galera handiak ekarriko ditu.
R03	2020/03/03	Workflow-lengoiaren aldaketak	Workflow-eredua sortzea prozesu neketsua da, nabigagarria denez aldaketak egitean fitxategi asko aldatu behar dira.
R04	2020/03/03	Aukeratutako softwarearen bizi-zikloa ezartzen duen metodologiaren aldaketa	Behin proiektuaren garapenean zehar metodologiaz aldatzea osagai guztiak aldatu beharra ekarriko du.
R05	2020/04/20	Drupa CMSaren bertsio aldaketa	CMSa eguneratzean, orain arte inplementatutako moduluek funtzionatzeari utz diezaiokete edo sartutako datuak desagertu egin daitezke.
R06	2020/05/24	Drupal CMSaren eta datu-baseen migrazioa	CMSa eta datu-baseak beste konputagailu batera migratzean agertu daitezken arazoak. Informazioa eta moduluen erredundantzia galtzea adibidez.
R07	2020/07/01	Proiektuko memoriaren eta bere eranskinen arteko inkongruentziak	CCII-2016N-02 eta OpenUP metodologia jarraitu denez, eranskinetan dokumentu asko daude eta koherentzi faltak agertu daitezke dokumentu guztien artean.

IV.1.1 taula: Arriskuen identifikazio-zerrenda

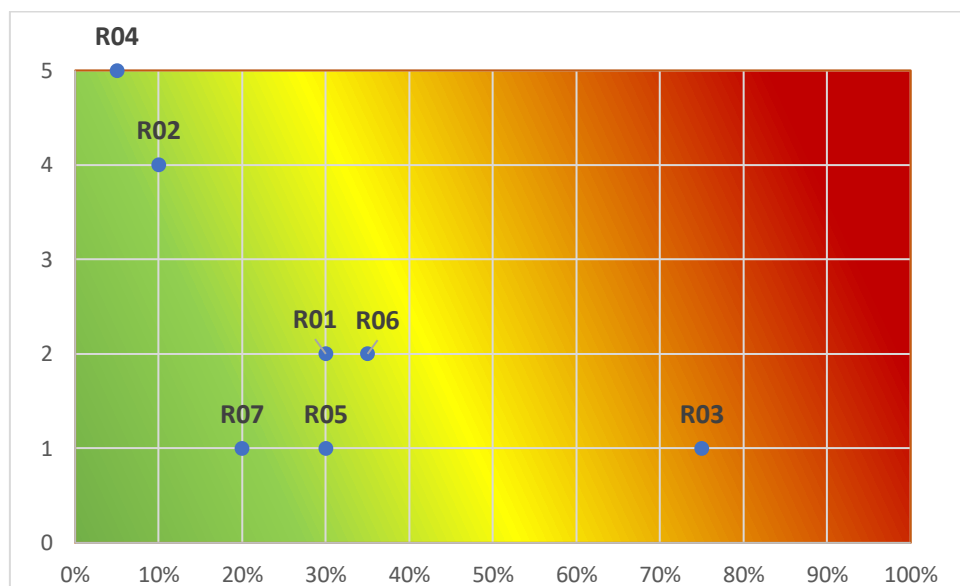
## IV.1.2 Arriskuen analisi kuantitatibo eta kualitatiboa

Hurrengo taulan aurreko atalean identifikatutako arriskuen tamaina edo larritasuna neurtuko da, horretarako arriskuaren inpaktua eta bera agertzeko probabilitatearen arteko biderketa egingo da:

Arriskuaren IDa	Inpaktua (1-5)	Probabilitatea (%)	Larritasuna
R01	2	%30	$2 * 0.3 = 0.6$
R02	4	%10	$4 * 0.1 = 0.4$
R03	1	%75	$1 * 0.75 = 0.75$
R04	5	%5	$5 * 0.05 = 0.25$
R05	1	%30	$1 * 0.3 = 0.3$
R06	2	%35	$2 * 0.35 = 0.7$
R07	1	%20	$1 * 0.2 = 0.2$

IV.1.2 taula: Arriskuen identifikazio-zerrenda

Jarraian arriskuen larritasuna adierazten duen grafikoa ikusi daiteke (ikus IV.1.1 irudia), X ardatzean arriskua agertzeko probabilitatea eta Y ardatzean arriskuaren inpaktua adierazten da. Kolore berdea larritasun gutxiko arriskua dela adierazteko erabili da eta gorria larritasun handia adierazteko.



IV.1.1 irudia: arriskuen larritasuna adierazten duen grafikoa

### IV.1.3 Arriskuak arintzeko estrategiak

Aurreko analisi kuantitatibo eta kualitatiboa kontuan hartuz, hurrengo taulan arriskuei erantzuteko plangintza deskribatuko da, arriskuak bere larritasunaren arabera zerrendatuz (goian larriena):

Arriskuaren IDa	Larritasuna	Arriskua arintzeko estrategia
R03	0.75	Workflow-lengoiaren xehetasun guztiak ondo pentsatu eta kontsultatu tutorearekin ezer egin baino lehen. Ondoren, xehetasun guztiak dokumentu batean idatzi. Workflow motorrak exekutatu behar duela pentsatu, beraz motorrari egokitu beharko da.
R06	0.7	Migratuko den konputagailuaren espezifikazioak aztertu eta migrazio prozesua ondo ikasi. Datu-base zein <i>Drupalen</i> kopiak egin eta <i>Drupal</i> mantenu moduan jarri.
R01	0.6	Instalazioan zehar egindakoa dokumentu batean idatzi, instalatuko dudan softwarearen espezifikazioak ondo irakurri.
R02	0.4	CMS ezberdinei buruzko azterketa sakona egin, aukeraketa egin baino lehen.
R05	0.3	<i>Drupal</i> bertsioz aldatzeko hartu behar diren arreta-neurriak eta egin behar diren prestakuntza guztiak ondo aztertu.
R04	0.25	Metodologia ezberdinei buruzko azterketa sakona egin, metodologia arina eta sinplea aukeratu.
R07	0.2	Eranskin guztiak berrikusi, batez ere, proiektuaren hasieran idatzitakoak. Gainera, memorian kapitulu bat gehitu, memorian idatzitakoa eranskinetan idatzitakoaren aurretik doala adieraziz.

IV.1.3 taula: Arriskuak arintzeko estrategiak



## IV.2. Proiektuaren Antolaketa eta Kudeaketa

---

Kapitulu honek proiektuaren antolaketa eta kudeaketa egiteko jarraitutako plana deskribatzea du helburu. Proiektu honen antolaketa eta kudeaketa egiteko *Project Management Insitutek* gomendatutako *PMBOK* gida jarraitu da.

### IV.2.1 Proiektuaren antolaketa

Proiektuaren antolaketa *OpenUP* metodologiaren bitartez definitu da, beraz gomendatzen da eranskinetako “Project Plan” dokumentua irakurtzea ikuspegi zabalagoa izateko.

#### IV.2.1.1 Taldekideak

- Julen Rojo Raño (JRR)
- Juan Manuel Pikatza (JMP)

#### IV.2.1.2 Ardurak eta betekizunak

- Zuzendaria: JRR
- Analista, arkitektoa, garatzailea, *testerra*: JRR
- Proiektuaren tutorea: JMP

#### IV.2.1.3 Lan-ingurunea

Normalean etxetik lan egitea zehaztu zen, baina posible zen EHUK eskaintzen duen Gradu Amaierako Lana egiteko gelatara joatea.

*Eguneraketa: COVID-19 birusaren egoera zela eta guztiz etxetik lan egin zen.*

#### IV.2.1.4 Informazio-sistema

Proiektu honen dokumentuak eta fitxategiak zuzendariaren ordenagailuan gordetzea eta segurtasun-kopia moduan tutoreak eskuratutako zerbitzari propio bat erabiltzea zehaztu zen. Tutoreak eta zuzendariak bakarrik dute zerbitzarirako sarbidea.

#### IV.2.1.5 Komunikazio-kanalak

Proiektuko tutorearekin harremanean jartzeko bilerak egitea adostu zen, normalean astero bat eginez. Arazo edo beste konturen bat izatekotan posta elektronikoaren bitartez komunikatuko zen. Hurrengo posta helbideak erabili ziran:

- JRR: [julenrojo12@gmail.com](mailto:julenrojo12@gmail.com) edo [vrojo013@ikasle.ehu.eus](mailto:vrojo013@ikasle.ehu.eus)
- JMP: [jm.pikatza@ehu.eus](mailto:jm.pikatza@ehu.eus)

Tutorearen eta zuzendariaren arteko fitxategien transferentziak egiteko zerbitzaria erabiltzea zehaztu zen.

Eguneraketa: COVID-19 birusaren egoera zela eta bilera birtualak egin ziran, Blackboard Collaborate web tresna erabiliz.

## IV.2.2 Proiektuaren kudeaketa

Kudeaketa, antolaketaren moduan, *OpenUP* metodologiaren bitartez egin da. Metodologia horrek aurretik finkatutako helburuak dituzten iterazio batzuk ezartzea proposatzen du. Iterazioak hasieran definitu edo proiektuan zehar gehitu daitezke.

Iterazio bakoitzaren hasieran, proiektuaren kudeaketari buruzko hurrengo artefaktuak eguneratu behar dira:

- Proiektu plana (*Project Plan*): egungo iterazioan bete beharreko mugarriak eta helburuak gehitu.
- Iterazio plana (*Iteration Plan*): iterazioa deskribatu eta lan-atazak zerrendatu. Iterazioa amaitzean ebaluazioa egin.
- Lan-atazen zerrenda (*Work Items List*): iterazioan zehar egin beharreko jarduerak eta bere estimazioak gehitu, ondoren, iterazioa amaitzean sartutako orduak erregistratu.
- Arriskuen zerrenda (*Risk List*): iterazioan identifikatutako arriskuak gehitu.

Hurrengo kapituluan proiektuko lan-atazak eta iterazioak deskribatuko dira, bere denbora-estimazioen eta sartutako denboraren arteko alderaketa eginez.



## IV.3. Denboraren Planifikazioa

Kapitulu honetan, proiektu hau egiteko diseinatutako denbora-plangintza aurkeztuko da. Aurreko kapituluak aipatu den moduan, *OpenUP* metodologia jarraituz **iterazioen bidezko plangintza** definitu da.

### IV.3.1 Mugarri garrantzitsuak

Proiektu honen bezero edo interesdun gisa EHUKo Informatika Fakultatea hartu daiteke, hori dela eta, mugarri garrantzitsuak aukeratzeko orduan fakultateak GrAlak matrikulatzeko, entregatzeko eta defendatzeko mugak kontuan hartu dira. Hala ere, tutorearekin adostutako mugarriak ere ezarri dira.

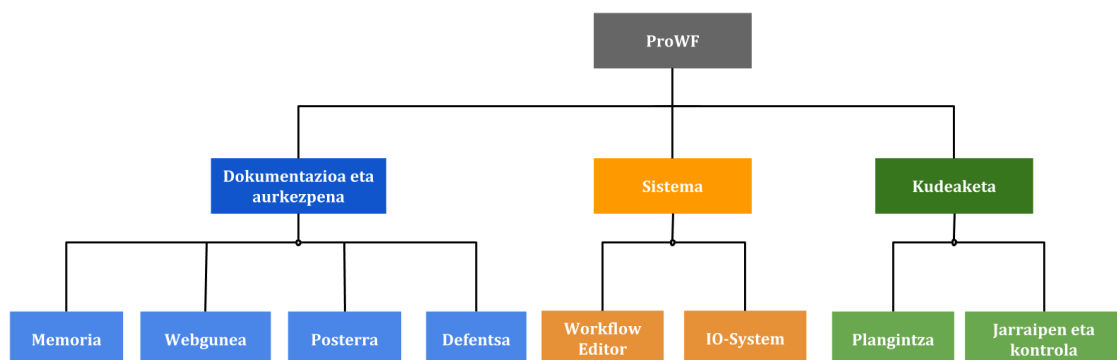
IV.3.1 taulan proiektuaren mugarri garrantzitsuenak eta haien deskribapenak biltzen dira, horiz azpimarratutakoak fakultateak ezarritakoak izanik.

Mugarria	Deskribapena
2020.02.04	Proiektuari hasiera eman.
2020.07.24	Proiektua matrikulatzeko azken eguna.
2020.08.01	Sistema eta memoria bukatu.
2020.09.06	Proiektuaren entrega: memoria eta posterra.
2020.09.14-18	Proiektuaren defentsa: aurkezpena.

IV.3.1 taula: Proiektuko mugarri garrantzitsuak

### IV.3.2 Lan-ataza nagusien diagrama (LDE)

Jarraian, proiektuaren lan-ataza nagusiak antolatu eta multzokatu dira:



IV.3.1 irudia: Lan-ataza nagusien diagrama (WBS/LDE)

Azken mailan kokatutakoak proiektuko lan-pakete nagusiak dira, hona hemen lan-pakete horien deskribapenak eta bere azpian dituzten atazak:

- **Memoria (MM):** dokumentu hau eta bere eranskin guztiak.
  - **MM1:** memoriaren eranskinak idatzi.
  - **MM2:** memoria idatzi.
- **Webgunea (WG):** proiektuaren dokumentu guztiak biltzen dituen eta CCII-2016N-02 araua betetzen duen webgunea.
  - **WG1:** CCII-2016N-02 araua irakurri.
  - **WG2:** webgunea sortu eta hasierako dokumentuak jarri.
  - **WG3:** webgunea osatu.
- **Posterra (PS):** proiektuaren ideia garrantzitsuenak biltzen dituen posterra.
  - **PS1:** posterra sortu.
- **Defentsa (DF):** proiektuaren defentsa egiteko prestatu beharreko aurkezpena.
  - **DF1:** aurkezpena sortu.
- **Workflow Editor (WE):** workflow-lengoaia eta workflow editorea.
  - **WE1:** workflow-lengoaia sortu.
  - **WE2:** workflow-editorea inplementatu.
- **IO-System (IO):** sistemaren web-aplikazioa eta workflow motorra.
  - **IO1:** datu-base dokumentalaren diseinua egin.
  - **IO2:** datu-base erlazionalaren diseinua egin.
  - **IO3:** workflow motorra sortu.
  - **IO4:** workflow exekutatzaileria inplementatu.
  - **IO5:** workflow exekuzioaren interfazea egin.
  - **IO6:** Proiektuen informazioa erakusteko interfazea egin.
  - **IO7:** guztia integratu.
- **Plangintza (PL):** *OpenUP* metodologiaren bitartez jarraitu beharreko plangintza. Proiektuaren barne-elaborazioa.
  - **PL1:** proiektuaren plangintza egin.
- **Jarraipen eta kontrola (JK):** *OpenUP* metodologiaren bitartez egin beharreko jarraipen eta kontrola eta bilera-aktak. Proiektuaren barne-elaborazioa.
  - **JK1:** iterazioen kontrola eraman eta bilera-aktak idatzi.

### IV.3.3 Lan-atazen denbora estimazioa

Aurreko atalean definitutako paketeak eta atazak kontuan hartuz, IV.3.2 irudian agertzen den hilabeteko lanaren estimazioa egin zen. Kontuan hartu behar da tutorearekin abuztuan atsedena hartzea eta irailerako bakarrik proiektuaren defentsa eta posterra uztea adostu zela.

	Otsaila	Martxoa	Apirila	Maiatza	Ekaina	Uztaila	Abuztua	Iraila
<b>Memoria (MM)</b>								
MM1								
MM2								
<b>Webgunea (WG)</b>								
WG1								
WG2								
WG3								
<b>Posterra (PS)</b>								
PS1								
<b>Defentsa (DF)</b>								
DF1								
<b>Workflow Editor (WE)</b>								
WE1								
WE2								
<b>IO-System (IO)</b>								
IO1								
IO2								
IO3								
IO4								
IO5								
IO6								
IO7								
<b>Plangintza (PL)</b>								
PL1								
<b>Jarraipen eta kontrola (JK)</b>								
JK1								

IV.3.2 irudia: hilabete bakoitzean egin beharreko lanaren estimazioa

Proiektuaren hasieran aurreikusi zen **lan osoa 500 ordukoa** izango zela. Hona hemen lan-pakete eta ataza bakoitzari estimatutako denbora (orduak):



<b>Memoria (MM)</b>	<b>125</b>
MM1	75
MM2	50
<b>Webgunea (WG)</b>	<b>20</b>
WG1	5
WG2	10
WG3	5
<b>Posterra (PS)</b>	<b>5</b>
PS1	5
<b>Defentsa (DF)</b>	<b>15</b>
DF1	15
<b>Workflow Editor (WE)</b>	<b>80</b>
WE1	40
WE2	40
<b>IO-System (IO)</b>	<b>210</b>
IO1	10
IO2	10
IO3	60
IO4	20
IO5	40
IO6	20
IO7	50
<b>Plangintza (PL)</b>	<b>25</b>
PL1	25
<b>Jarraipen eta kontrola (JK)</b>	<b>20</b>
JK1	20
<b>GUZTIRA</b>	<b>500</b>

IV.3.2 taula: lan-ataza bakoitzari estimatutako ordu kopurua

Ataza bakoitzaren formakuntza-prozesua IV.3.2 taulan agertzen diren denboren barne dago. Esan daiteke formakuntza-prozesua agertzen den denbora bakoitzaren %30a zela.

### IV.3.4 Iterazio-prozesua

Atal honetan, *OpenUP* metodologiaren bitartez jaso diren atazak iterazioetan zehar nolako banaketa izan duten deskribatuko da. Ataza horiek izan duten denbora-kostu erreala ere adieraziko da.

Proiektua **bost iteraziotan banatu da**, lehenengo biak proiektuaren hasieran definitu ziran, hurrengoak, ordea, proiektuaren garapenean zehar gehitu ziran. Jarraian iterazioak eta iterazio bakoitzaren helburuak deskribatzen dira:

ID	Iterazioa	Helburuak
I1	2020.02.04 - 2020.03.03	Lan-ingurunearen prestakuntza Proiektuaren webgunearen prestakuntza (WG1) Betekizunen ingeniariaritz (MM1) Proiektu eta iterazio plana (PL1, JK1) Workflow-lengoaia sortu (WE1)
I2	2020.03.04 - 2020.03.31	Iterazio berria gehitu (JK1) Datu-base dokumentala diseinatu (IO1) Proiektuaren webgunea sortu eta antolatu (WG2) Proiektuaren informazio erakusteko modulua implementatu (IO6) Workflow-lengoaia aldatu (WE1) Workflow editorea implementatu (WE2)
I3	2020.04.1 - 2020.05.30	Iterazio berria gehitu (JK1) Datu-base erlazionala diseinatu (IO2) Workflow exekuzioaren interfazea implementatu (IO5) Workflow motorrarekin hasi (IO3) Workflow-lengoaia aldatu (WE1)
I4	2020.06.01 - 2020.07.30	Iterazio berria gehitu (JK1) Memoria idatzi (MM2) Proiektuaren webgunea osatu (WG3) Workflow-lengoaia aldatu (WE1) Workflow exekutatzaila implementatu (IO4) Workflow motorra bukatu (IO3) Modulu guztiak integratu eta probak egin (IO7)
I5	2020.08.01 - 2020.09.18	Iterazio berria gehitu (JK1) Memoria egokitu eta guztiz zuzendu (MM2) Proiektuaren webgunea osatu (WG3) Proiektuaren posterra egin (PS1) Proiektuaren aurkezpena prestatu (DF1) Proiektuaren entrega egin Proiektuaren defentsa egin

IV.3.3 taula: Proiektuko iterazioen banaketa eta bakoitzaren helburuak

*OpenUP* metodologiako “Lan-atazen zerrenda” dokumentuari esker, proiektuko atazen jarraipen eta kontrola egitea posible izan da, iterazioen amaieran ataza bakoitzari emandako ordu kopuruak eta gelditzen zitzairen orduen estimazioa adieraziz. Horri esker, ataza guztiei emandako ordu kopuru totalak kalkulatu dira eta aurretik estimatutako denborekin alderatu izan ahal dira (ikus IV.3.4 taula).



<b>Memoria (MM)</b>	<b>110</b>	<b>15</b>
MM1	60	15
MM2	50	0
<b>Webgunea (WG)</b>	<b>5</b>	<b>15</b>
WG1	1	4
WG2	1	9
WG3	3	2
<b>Posterra (PS)</b>	<b>5</b>	<b>0</b>
PS1	5	0
<b>Defentsa (DF)</b>	<b>15</b>	<b>0</b>
DF1	15	0
<b>Workflow Editor (WE)</b>	<b>110</b>	<b>-30</b>
WE1	100	-60
WE2	10	30
<b>IO-System (IO)</b>	<b>246</b>	<b>-36</b>
IO1	1	9
IO2	5	5
IO3	80	-20
IO4	10	10
IO5	50	-10
IO6	30	-10
IO7	70	-20
<b>Plangintza (PL)</b>	<b>15</b>	<b>10</b>
PL1	15	10
<b>Jarraipen eta kontrola (JK)</b>	<b>10</b>	<b>10</b>
JK1	10	10
<b>GUZTIRA</b>	<b>516</b>	<b>-16</b>

IV.3.4 taula: ataza bakoitzari emandako ordu kopuru totala eta estimatutakoaren diferentzia (gorriz, estimatutakoak baino ordu gehiago sartu dira; berdez, estimatutakoak baino gutxiago)

Proiektuan guztira **516 ordu sartu dira**, kopuru horretatik **100 ordu inguru prestakuntzan** eman dira. kopuru hori ez da asko urruntzen aurreikusitako ordu kopuruetatik, beraz, esan daiteke denboraren planifikazio egokia egin dela.

### IV.3.5 Desbiderapenak

Nahiz eta proiektuko ordu kopuru totala ondo kalkulatu den, desbideratze aipagarri batzuk egon dira. Hurrengoak izan dira proiektuaren desbideratze nagusiak:

- **Workflow-lengoaia (WE1) eta workflow editorea (WE2).** Workflow-lengoaia sortzeak aurreikusitakoa baino askoz denbora gehiago behar izan du, 60 ordu. Gainera, lehenengo iterazioan hasi eta bigarren iterazioan bukatzea estimatu zen, eta oster, laugarren iterazioan bukatu izan da. Iterazio bakoitzean aldaketa asko jasan behar izan ditu, hobekuntza-prozesu baten egon delako proiektu guztian zehar. Beste alde batetik, workflow editorea inplementatzeak estimatutako baino denbora gutxiago behar izan du, 30 ordu. Esan daiteke *Workflow Editor* azpi-sistemaren atazarik garrantzitsuen workflow-lengoaia sortzearena izan dela.
- **Workflow motorra (IO3).** Inferentzia motorrak *workflow*-lengoiak izan duen antzeko desbiderapena izan du, *workflow*-lengoaia aldatuz inferentzia motorraren erregelak aldatu behar zirelako. Gainera, motorra apirilean hastea eta ekainean bukatzea estimatu zen, baina oster, uztailean arte luzatu zen.



- **Datu-baseen diseinua (IO1 eta IO2).** Datu-base erlazionala zein dokumentalaren diseinua aurreikusitako baino askoz errazagoa izan da. Bakoitzari 10 ordu sartzea estimatu zen, hala ere, datu-base dokumentala ordu baten diseinatu zen eta erlazionala 5 ordutan, bigarrenak aldaketak jasan zituelako. Beraz, guztira 20 ordu behar zuten atazek 6 ordutan egitea lortu zen.
- **Proiektuaren webgunea (WG1, WG2 eta WG3).** Datu-baseen diseinuen moduan, aurreikusitako baino azkoz errazagoa izan da proiektuaren webgunea antolatzea *CCII-2016N-02* araua jarraituz. Pasadan ikasturteko Softwarearen Kalitatea irakasgaiko proiektu baten arau hori jarraitzen zuen webgune bat sortu eta antolatu zen, beraz horrek asko lagundu du proiektu honen webgunea sortzeko prozesuan. Guztira estimatutako baino 15 ordu gutxiago behar izan dira proiektuaren webgunearen atazak garatzeko.





## IV.4. Aurrekontuaren Laburpena

---

Kapitulu honetan, proiektuaren kostu ekonomikoa aterako da.

### IV.4.1 Orokortasunak

Aurrekontua sortzeko *ALI (Asociación de Titulados Universitarios Oficiales en Informática)* elkarteak banatutako irizpide batzuk jarraitu dira. Batetik, giza baliabideen barne-kostuk eta kanpo-kostuak ateratzea bere ordu kopuruekin batera eta bestetik, proiektua garatzeko behar izan diren erraminten kostua kalkulatzeko. *Testing* tekniken eta auditoretza baten ziurtagiriaren kostua alde batera utzi da.

Giza baliabideen kostua Ekonomia eta Ogasun Ministerioaren 26/2010 Esparru Akordioan oinarrituta dago. Bertan, software proiektu baten kide bakoitzari hurrengo ordainketa egitea proposatzen da:

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

Beraz, giza baliabideen barne-kostuaren balio hurbildua lortzeko haien arteko batezbestekoa egin da eta **60€/orduko** parametroa atara da. Beste alde batetik, proiektuaren memoriak, posterrak eta defentsak behar izan duten ordu kopurua baztertu da barne-kostuen ordu kopurutik, **386 ordutan** geratuz.

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.

Jarraian, IV.4.1 taulan, proiektuaren aurrekontua ikus daiteke:

PARTIDA		PARAMETROAK				TOTALA (BEZik gabe)	TOTALA (BEZa barne)
Giza Baliabideak		Ordu kopurua	Barne-kostuak	Ordu kopurua	Kanpo-kostuak		
		386	60€	0	0	23.160	28.024
Erramintak		Lizentzia		Mantenua (urtero)			
1	XAMPP	0		0		0	0
2	Drupal 8	0		0		0	0
3	PlantUML	0		0		0	0
4	EHSIS	0		0		0	0
5	GraphViz	0		0		0	0
6	Gephi	0		0		0	0
7	Protégé	0		0		0	0
8	Notepad++	0		0		0	0
Erramintak - Totala						0	0
TOTALA						23.160	28.024

IV.4.1 taula: proiektuaren aurrekontua

Ondorioz, proiektu hau aurrera ateratzeko **28.024€** (BEZa barne) behar dira.