

## Informatika Fakultatea

### Informatika Ingeniaritzako Gradua

#### ▪ Gradu Amaierako Lana ▪

Software Ingeniaritza



#### ProMeta

Metaereduetan oinarritutako softwarearen garapenerako  
prozesuen definizio eta ezarpenerako sistema

---

Egilea: Julen Etxaniz Aragoneses

Tutorea: Juan Manuel Pikatza Atxa

Data: 2021

## Laburpena

Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema.

## Aurkibide Orokorra

Laburpena.....	2
Irudien Aurkibidea .....	8
Taulen Aurkibidea.....	9
1 Sarrera .....	10
1.1 Testuingurua .....	10
1.1.1 Arazoaren Planteamendua .....	10
1.1.2 Produktuaren Planteamendua .....	11
1.2 Proiektuaren Webgunea .....	11
2 Helburua .....	12
3 Aurrekariak .....	13
4 Egungo Egoera .....	14
4.1 Egungo Egoeraren Deskribapena .....	14
4.2 Identifikatutako Hutsuneen Deskribapena .....	14
5 Arauak eta Erreferentziak .....	16
5.1 Aplikatutako Legedia eta Araudia .....	16
5.2 Bibliografia .....	16
5.3 Metodoak, Tresnak, Ereduak, Metrikak eta Prototipoak.....	16
5.3.1 Metodoak eta Tresnak.....	16
5.3.2 Ereduak, Metrikak eta Prototipoak .....	16
5.4 Idazketaren Kalitatearen Kudeaketa Plana .....	17
5.5 Beste Erreferentziak.....	17
6 Definizioak eta Laburdurak .....	18
7 Hasierako Betekizunak.....	19
7.1 Betekizun Funtzionalak .....	19
7.2 Betekizun Ez-Funtzionalak .....	19
8 Irismena .....	20
9 Hipotesiak eta Murritzapenak .....	21
10 Aukeren Azterketa eta Egingarritasuna .....	22
10.1 CMS aukerak .....	22
10.1.1 Wordpress .....	23
10.1.2 Drupal.....	23
10.1.3 Joomla .....	23
10.2 Dokumentazioa Hosting Aukerak.....	24
Memoria .....	2021/05/21
	3 / 36

10.2.1	GitHub Pages .....	24
10.2.2	Netlify .....	24
10.3	Drupal Hosting Aukerak .....	24
10.3.1	000webhost.....	24
10.3.2	Heroku .....	24
10.3.3	Acquia.....	25
10.3.4	Pantheon .....	25
10.3.5	Platform.sh .....	25
10.4	Datu-base kudeaketa sistema .....	25
10.4.1	MySQL .....	25
10.4.2	PostgreSQL .....	25
10.5	Datu-basearen sorrera .....	25
10.5.1	Inferentzia motorra .....	25
10.5.2	Teneo.....	25
10.5.3	Xtext .....	25
10.6	Lanerako ingurunea .....	26
10.6.1	Makina birtuala .....	26
10.6.2	Ordenagailu pertsonala.....	26
10.7	Bertsio kontrola .....	26
10.7.1	GitHub .....	26
10.7.2	GitLab .....	26
10.8	Metaereduentzako tresnak.....	26
10.8.1	Eclipseren tresnak .....	26
10.8.2	SPEM metaeredua.....	26
10.9	Metodologia aukeratu .....	26
10.9.1	OpenUp .....	26
10.9.2	RUP .....	26
10.9.3	ABRD.....	26
10.10	Metodologia definitu .....	26
10.10.1	EPF Composer .....	26
10.10.2	Rational Method Composer .....	27
10.10.3	Editore grafikoa .....	27
10.10.4	Testu editorea .....	27
10.11	Prozesua bistaratu .....	27

10.11.1	DOT.....	27
10.11.2	XPDL .....	27
10.11.3	Webgunea .....	27
10.12	Proiektuaren kudeaketa .....	28
10.12.1	ClickUp.....	28
10.12.2	Quire.....	28
10.12.3	Trello .....	28
10.13	Denboraren kontrola .....	28
10.13.1	Clockify .....	28
10.13.2	Toggl.....	28
10.13.3	Waketime .....	28
10.14	Gantt diagrama .....	28
10.14.1	Teamgantt .....	28
10.14.2	Elegantt .....	28
10.14.3	Ganttproject .....	28
11	Proposatutako Sistemaren Deskribapena .....	29
12	Arriskuen Analisia .....	31
13	Proiektuaren Antolamendua eta Kudeaketa .....	32
13.1	Proiektuaren Antolamendua .....	32
13.1.1	Proiektuaren Aktoreak eta Erlazioak.....	32
13.1.2	Barne Egitura.....	32
13.1.3	Kanpo Interfazeak .....	32
13.1.4	Rolak eta Ardurak.....	32
13.2	Proiektuaren Kudeaketa .....	32
13.2.1	Integrazioaren Kudeaketa .....	32
13.2.2	Irismenaren Kudeaketa .....	32
13.2.3	Epeen Kudeaketa.....	32
13.2.4	Produktuaren Kostuen Kudeaketa .....	32
13.2.5	Kalitate Kudeaketa .....	32
13.2.6	Giza Baliabideen Kudeaketa .....	32
13.2.7	Komunikazioen Kudeaketa .....	32
13.2.8	Arriskuen Kudeaketa .....	32
13.2.9	Erosketen Kudeaketa .....	32
13.2.10	Interesatuen Kudeaketa .....	32

14	Denbora Planifikazioa .....	33
14.1	Proiektu Planaren Eboluazioa .....	33
14.2	Hornitzailearen Proiektu Planaren Eboluzioa .....	33
15	Aurrekontuaren Laburpena .....	34
16	Dokumentuen Lehentasun Ordena .....	35
17	Memoriaren Eranskinak.....	35
17.1	Sarrerako Dokumentazioa .....	35
17.2	Analisia eta Diseinua .....	35
17.2.1	Arkitektura Kuadernoa .....	35
17.2.2	Analisiaren Eredua .....	35
17.2.3	Diseinuaren Eredua .....	35
17.3	Tamaina eta Esfortzu Estimazioa .....	35
17.4	Kudeaketa Plana .....	35
17.4.1	Integrazioaren Kudeaketa .....	35
17.4.2	Irismenaren Kudeaketa .....	35
17.4.3	Epeen Kudeaketa.....	35
17.4.4	Produktuaren Kostuen Kudeaketa .....	35
17.4.5	Kalitate Kudeaketa .....	35
17.4.6	Giza Baliabideen Kudeaketa .....	35
17.4.7	Komunikazioen Kudeaketa.....	35
17.4.8	Arriskuen Kudeaketa .....	35
17.4.9	Erosketen Kudeaketa .....	35
17.4.10	Interesatuen Kudeaketa .....	35
17.5	Segurtasun Plana .....	35
17.6	Beste Eranskinak .....	35
17.6.1	Hedapena .....	35
17.6.2	Garapena .....	35
17.6.3	Ingurunea .....	35
17.6.4	Proba .....	35
18	Sistemaren Espezifikazioa.....	35
18.1	Glosategia .....	35
18.2	Ikuspegia .....	35
18.3	Betebeharren Espezifikazioa.....	35
18.4	Erabilpen Kasuak.....	35

18.5	Erabilpen Kasuen Eredua .....	35
19	Aurrekontua.....	36
20	Ikerlanak .....	36

Irudien Aurkibidea

Irudia 1. Prometa logoa. .... 10



Taulen Aurkibidea

Taula 1. Arazoaren Planteamendua. .... 11

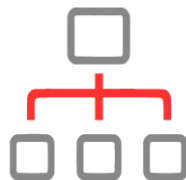
Taula 2. Produktuaren Planteamendua..... 11

## 1 Sarrera

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

Dokumentu honetan ProMeta proiekturen inguruko aspektu guztiak azalduko dira. ProMeta izena prozesu edo profesional eta metaeredu hitzetatik dator. Irudia 1 ikusten den proiektuaren logoak metaeredu bat irudikatzen du. Proiektuaren izenburua **Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema** da. Beraz, proiektuak bi helburu nagusi ditu: garapenerako prozesuen definizioa eta prozesuaren ezarpena.

Proiektua OpenUP metodologia erabili gaizatuenez, bertako pausoak jarraitu dira proiektua gauzatzeko. Gainera, dokumentazio formala eta profesionala lortzeko asmoz, ezaguna den CCII-N2016-02 arau estandarra eta Informatika Fakultateko GrAL ereduak erabili dira dokumentu honen eta proiektuaren webgunearen atalak definitzeko.



Irudia 1. ProMeta logoa.

### 1.1 Testuingurua

Proiektuaren testuingurua ulertu ahal izateko, konpondu nahi den arazoa eta sortuko den produktua ulertu behar dira.

#### 1.1.1 Arazoaren Planteamendua

Proiektuaren testuingurua ulertu ahal izateko, lehenik konpondu nahi dugun arazoa ulertu behar da. Honakoa jakin behar dugu: arazoa zein den, nori eta nola eragiten dion eta gure soluzioa zein den. Ikusi Taula 1.

<b>Arazoaren Deskribapena</b>	Proiektu informatikoen elaboraziorako ezagutza erabilgarria pilatzen da zenbait jardura modu automatikoan egiteko, adostutako gida bati jarraituz. Gida hauek softwarea garatzeko prozesuetan, metodologietan eta estandarretan oinarritzen dira. Beharrezkoa da gida hauek definitu eta kudeatu ahal izatea, mantendu eta hobetu ahal izateko. Horrek giden edukia eta horiek exekutatzeko dituen sistema definitzea eskatzen du.
<b>Interesatu Kaltetuak</b>	Proiektu informatikoak garatzen dituzten erakundeak.
<b>Arazoaren Eragina</b>	Estandarizazio maila baxua software proiektuen garapen prozesuan. Desadostasunak, atzerapenak eta akatsak eragiten ditu honek.

<b>Soluzioaren Abantailak</b>	Softwarea garatzeko prozesua definitu, kudeatu, mantendu eta hobetzeko erraztasuna. Garapen denbora asko gutxituko du eta bizi-ziklorako metodologia bat erabiliz estandarizazio-maila handituko du.
-------------------------------	--

*Taula 1. Arazoaren Planteamendua.*

### 1.1.2 Produktuaren Planteamendua

Arazoa ulertu ondoren, produktuaren inguruko planteamendua azalduko dugu. Bezeroa eta beharra, gure produktua, alternatibak eta horiekiko abantaila eta hobekuntzak zein diren jakin behar dugu. Ikusi Taula 2.

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

*Taula 2. Produktuaren Planteamendua.*

## 1.2 Proiektuaren Webgunea

Proiektuaren webgunea honakoa da: <https://juletx.github.io/ProMeta/>. 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. Gainera, webguneak CCII N2016-02 estandarra betetzen duenez, irakurle adituak oso azkar identifikatuko du behar duen dokumentua. Behin menuko aukera batean klikatzen denean, zati nagusian PDF formatu gisa irekiko da dokumentua.

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. Eskuinean, ordea, nabigazio menuan aukeratutakoa agertuko da.

## 2 Helburuak

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

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

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

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

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

### 3 Aurrekariak

## 4 Egungo Egoera

### 4.1 Egungo Egoeraren Deskribapena

### 4.2 Identifikatutako Hutsuneen Deskribapena

ProWF proiektuan honako urritsunak identifikatu ziren:

- **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.

### 4.3 Prestakuntza

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

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

Softwarearen Garapen Industrialak irakasgaien beste gai hauek landu ziren:

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

Web Sistemak irakasgaien ikasitakoak:

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

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

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

## 5 Arauak eta Erreferentziak

### 5.1 Aplikatutako Legedia eta Araudia

### 5.2 Bibliografia

### 5.3 Metodoak, Tresnak, Ereduak, Metrikak eta Prototipoak

#### 5.3.1 Metodoak

##### **OpenUP**

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

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

##### **ABRD**

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

#### 5.3.2 Tresnak

- **Git**
- **GitHub**
- **GitHub Pages**
- **Toggle Track**
- **Java SE**
- **Eclipse IDE**
- **Eclipse Process Framework**
- **EPF Composer**
- **Eclipse Modelling Framework**
- **Xtext**
- **XSLT**
- **Drupal**
- **Pantheon**
- **XAMPP**
- **MySQL**
- **PHP**

#### 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. Txantilo horiek jarraituz OpenUP metodologiaren bitartez sortutako artefaktu guztiak idatzi dira.

## UMA

Unified Method Architecture metaeredua erabili da ereduak definitzeko. Metaeredu honen helburua edozein metodologia modelatu ahal izatea da.

### 5.3.4 Metrikak

Denboraren kontrola egiteko ataza bakoitzean pasatako denbora neurtu da, Toggle Track aplikazioaren kronometroa erabiliz.

### 5.3.5 Prototipoak

#### **ProMeta ModelEditor**

Proiektuko metaeredu, eredu eta editoreak biltzen dituen prototipoa.

#### **ProMeta IO-System**

Proiektuko Drupal webgunea eta datu-basea biltzen dituen prototipoa.

## 5.4 Idazketaren Kalitatearen Kudeaketa Plana

Dokumentuen idazkera eta antolamenduaren kalitatea bermatzeko CCII-N2016-02 araua, GrAL eredu eta OpenUP metodologia jarraitu dira.

## 5.5 Beste Erreferentziak

## 6 Definizioak eta Laburdurak

## 7 Hasierako Betekizunak

Kapitulu honetan proiektuaren hasierako betekizun funtzionalak eta ez-funtzionalak azaltzen dira.

### 7.1 Betekizun Funtzionalak

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

### 7.2 Betekizun Ez-Funtzionalak

Betekizuna	Lehenetasuna	Entrega Data
Dokumentazioa eta proiektuaren webgunea CCII estandarren arabera.	Altua	2021/06/20
Proiektuan OpenUp metodologia jarraitzea.	Altua	2021/06/20
Garapen prozesua metodologia eta estandarrek esaten duten moduan eratzea.	Altua	2021/06/20
Erreminta, metodologia eta ezagutzaren aldetik eman daitezkeen aldaketen aurrean, soluzioa malgua eta egokigarria izatea.	Ertaina	2021/06/20
Proiektuaren garapenerako doakoak eta libreak diren tresnak erabiltzea.	Ertaina	2021/06/20
Proiektuko osagaien dokumentazioa eta eskuliburuak	Ertaina	2021/06/20

## 8 Irismena

## 9 Hipotesiak eta Murriztapenak

## 10 Aukeren Azterketa eta Egingarritasuna

### 10.1 CMS aukerak

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

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

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

Ezaugarria	<i>Wordpress</i>	<i>Joomla</i>	<i>Drupal</i>	Oharra
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	-

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* enpresak *Drupal* erabiltzen du oinarri bezala.

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

### 10.1.1 Wordpress

<https://wordpress.com/>

<https://wordpress.org/>

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

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

WordPress PHP hizkuntzan garatzen da MySQL eta Apache exekutatzen duten inguruneetarako, GPL lizentziapean eta software libre da.

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

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

### 10.1.2 Drupal

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

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

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

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

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

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

### 10.1.3 Joomla

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

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

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

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

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

## 10.2 Dokumentazioa Hosting Aukerak

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

### 10.2.1 GitHub Pages

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

GitHubekin integratuta, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan. Konfigurazio oso erraza, erreositoriotik bertatik egiten da. Webgune estatikoak bakarrik onartzen ditu, beraz, drupal-erako ez du balio, PHP erabiltzen baitu.

Hasieran aukera hau erabili da, sinpleena baita. Baina, ProWF proiektuaren fitxategiak gehitutakoan arazoak eman ditu, drupalen fitxategiengatik. Konprimatuta uzten badira ez dute arazorik ematen baina deskonprimatutakoan bai.

### 10.2.2 Netlify

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

GitHubekin integratu daiteke, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan. Webgune estatikoak bakarrik onartzen ditu, beraz, drupal-erako ez du balio, PHP erabiltzen baitu.

## 10.3 Drupal Hosting Aukerak

### 10.3.1 000webhost

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

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

### 10.3.2 Heroku

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

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



### 10.3.3 Acquia

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

### 10.3.4 Pantheon

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

### 10.3.5 Platform.sh

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

## 10.4 Datu-base kudeaketa sistema

### 10.4.1 MySQL

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

### 10.4.2 PostgreSQL

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

## 10.5 Datu-basearen sorrera

### 10.5.1 Inferentzia motorra

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

### 10.5.2 Teneo

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

### 10.5.3 Xtext

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

## 10.6 Lanerako ingurunea

### 10.6.1 Makina birtuala

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

### 10.6.2 Ordenagailu pertsonala

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

## 10.7 Bertsio kontrola

### 10.7.1 GitHub

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

### 10.7.2 GitLab

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

## 10.8 Metaereduentzako tresnak

### 10.8.1 Eclipse-ren tresnak

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

### 10.8.2 SPEM metaeredua

SPEM metaeredua cmof formatuan dago.

## 10.9 Metodologia aukeratu

### 10.9.1 OpenUp

### 10.9.2 RUP

### 10.9.3 ABRD

## 10.10 Metodologia definitu

### 10.10.1 EPF Composer

Metodologia definitzeko aukera argiena da, horretarako egindako tresna baita. Jadanik hainbat metodologia definituta daude tresna honekin: OpenUp, ABRD, Scrum... Metodologia horietako bat erabili nahi badugu ez daukagu ezer diseinatu beharrik. Horiek oinarri bezala hartuta ere errazagoa da beste

metodologia bat definitzea. Metodologiaren webgunea sortzeko aukera ematen du, dokumentazio moduan erabili ahal izateko. Nahiko zaharkitua dago, Java 8 32 biteko bertsioa eskatzen du.

#### 10.10.2 Rational Method Composer

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

#### 10.10.3 Editore grafikoa

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

#### 10.10.4 Testu editorea

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

### 10.11 Prozesua bistaratu

#### 10.11.1 DOT

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

#### 10.11.2 XPDL

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

#### 10.11.3 Webgunea

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

## 10.12 Proiektuaren kudeaketa

10.12.1 ClickUp

10.12.2 Quire

10.12.3 Trello

## 10.13 Denboraren kontrola

10.13.1 Clockify

10.13.2 Toggl

10.13.3 Wakatime

## 10.14 Gantt diagrama

10.14.1 Teamgantt

10.14.2 Elegantt

10.14.3 Ganttproject

## 11 Proposatutako Sistemaren Deskribapena

### 11.1 Arkitektura

### 11.2 Analisia

### 11.3 Diseinua

### 11.4 Garapena

### 11.5 Proba

### 11.6 Hedapena

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

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

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

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

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

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

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

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

Proiektua amaitutakoan, lana [GAUR](#)en matrikulatu behar da eta zuzendariak oniritzia eman behar du. Ondoren, ikasleak lana [ADDI](#) plataformara igoko du. Horrez gain, ikasleak bere lanaren posterra bidali behar du [dif.gral@ehu.eus](mailto:dif.gral@ehu.eus) helbidera.

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

## 12 Arriskuen Analisia

## 13 Proiektuaren Antolamendua eta Kudeaketa

### 13.1 Proiektuaren Antolamendua

#### 13.1.1 Proiektuaren Aktoreak eta Erlazioak

#### 13.1.2 Barne Egitura

#### 13.1.3 Kanpo Interfazeak

#### 13.1.4 Rolak eta Ardurak

### 13.2 Proiektuaren Kudeaketa

#### 13.2.1 Integrazioaren Kudeaketa

#### 13.2.2 Irismenaren Kudeaketa

#### 13.2.3 Epeen Kudeaketa

#### 13.2.4 Produktuaren Kostuen Kudeaketa

#### 13.2.5 Kalitate Kudeaketa

#### 13.2.6 Giza Baliabideen Kudeaketa

#### 13.2.7 Komunikazioen Kudeaketa

#### 13.2.8 Arriskuen Kudeaketa

#### 13.2.9 Erosketen Kudeaketa

#### 13.2.10 Interesatuen Kudeaketa

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



Izena	Deskribapena	Erantzukizunak
Proiektuko Prozesu Ingeniaria	Software proiektuaren eskakizunak eta betebeharrak aztertuz, prozesuak sortzen ditu.	Prozesua sortu/editatu, metaeredu eta ereduetan aldaketak eginez.  Prozesuaren gida automatikoak jarraitu
Proiektuaren beste taldekideak	Software proiektuaren arkitektoak, garatzaileak, testerrak...	Prozesuaren gida automatikoak jarraitu

## 14 Denbora Planifikazioa

### 14.1 Proiektu Planaren Eboluazioa

### 14.2 Hornitzailearen Proiektu Planaren Eboluzioa

## 15 Aurrekontuaren Laburpena

## 16 Dokumentuen Lehentasun Ordena

## 17 Memoriaren Eranskinak

### 17.1 Sarrerako Dokumentazioa

### 17.2 Analisia eta Diseinua

#### 17.2.1 Arkitektura Kuadernoa

#### 17.2.2 Analisiaren Eredua

#### 17.2.3 Diseinuaren Eredua

### 17.3 Tamaina eta Esfortzu Estimazioa

### 17.4 Kudeaketa Plana

#### 17.4.1 Integrazioaren Kudeaketa

#### 17.4.2 Irismenaren Kudeaketa

#### 17.4.3 Epeen Kudeaketa

#### 17.4.4 Produktuaren Kostuen Kudeaketa

#### 17.4.5 Kalitate Kudeaketa

#### 17.4.6 Giza Baliabideen Kudeaketa

#### 17.4.7 Komunikazioen Kudeaketa

#### 17.4.8 Arriskuen Kudeaketa

#### 17.4.9 Erosketen Kudeaketa

#### 17.4.10 Interesatuen Kudeaketa

### 17.5 Segurtasun Plana

### 17.6 Beste Eranskinak

#### 17.6.1 Hedapena

#### 17.6.2 Garapena

#### 17.6.3 Ingurunea

#### 17.6.4 Proba

## 18 Sistemaren Espezifikazioa

### 18.1 Glosategia

### 18.2 Ikuspegia

### 18.3 Betebeharren Espezifikazioa

### 18.4 Erabilpen Kasuak

### 18.5 Erabilpen Kasuen Eredua

19 Aurrekontua

20 Ikerlanak