

## Informatika Fakultatea

### Informatika Ingeniaritzako Gradua

- ProWF: Workflow Editorea ▪
- Eskuliburua ▪

Instalazio eta erabiltzailearen dokumentazioa

Julen Rojo Raño

2020/06/24



## Edukien taula

<b>1. Eskuliburuaren helburua .....</b>	<b>3</b>
<b>2. Instalazioa .....</b>	<b>3</b>
2.1 Softwareak .....	3
2.2 Instalazio-prozesua.....	3
<b>3. Aurrebaldintzak.....</b>	<b>4</b>
<b>4. Erabiltzailearen dokumentazioa .....</b>	<b>4</b>
4.1 Workflow-ereduak sortzen .....	4
4.2 Workflow-ereduen eraldaketa-prozesua .....	5
4.3 Workflow-ereduen informazioa datu-base erlazionalean gorde.....	11
4.4 Web-aplikaziorako prestakuntza.....	12
4.5 Artefaktuen txantiloak.....	12



## 1. Eskuliburuaren helburua

---

Dokumentu honen helburua **ProWFren Workflow Editor** azpisistemaren instalazio eta erabilpenerako gida izatea da. Erabiltzaileak edozein momentutan izan ahal dituen arazo eta kezka bertan konpontzeko gai izan beharko litzateke, bestela nire kontaktua hemen jartzen dut:

[julenrojo12@gmail.com](mailto:julenrojo12@gmail.com)

## 2. Instalazioa

---

Atal honetan **ProWFren Workflow Editor** azpisistemaren osagaiak instalatzeko gida agertuko da, instalaziorako beharrezkoak diren softwareen zerrenda, ingurunea prestatzeko konfigurazioa eta instalazio-prozesua azalduz.

### 2.1 Softwareak

Jarraian erabiltzailearen makinan instalatu beharreko softwareen zerrenda agertuko da:

- *Graphviz 2.38*
- *Gephi v 0.9.2*
- *Protégé v 3.5*
- *EHSIS v 1.08 (EHSIS\_RT produkzio bertsioa)*

### 2.2 Instalazio-prozesua

Softwareak deskargatu hurrengo webguneetatik:

- *Graphviz 2.38:*  
<https://graphviz.org/download/>
- *Gephi v 0.9.2:*  
<https://gephi.org/users/download/>
- *Protégé v 3.5:*  
[https://protege.stanford.edu/download/protege/3.5/installanywhere/Web\\_Installers/](https://protege.stanford.edu/download/protege/3.5/installanywhere/Web_Installers/)
- *EHSIS\_RT:*

ProWF sistemarekin dator, proiektuaren webgunean atzigarri:

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

Bakoitzaren instalazio-prozesua jarraitu, pausuz-pausu.



### 3. Aurrebaldintzak

*Workflow Editor* azpisistema erabiltzeko **ezinbestekoa da “Workflow-lengoiaren Eskuliburua” dokumentua irakurri izatea**. Lengoaia horren bitartez definituko baita azpisistema honetatik aterako diren *workflow*-ereduak.

### 4. Erabiltzailearen dokumentazioa

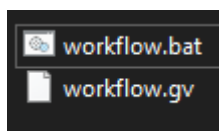
#### 4.1 Workflow-ereduak sortzen

Hasteko, workflow-eredu bat sortzeko .gv fitxategi bat sortu behar da, ondoren fitxategi hori testu-editore batekin editatu eta workflow-lengoaia baliatuz eredu bat sortu.

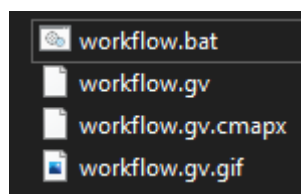
.gv fitxategiaren maila berdinean BATCH fitxategi bat sortu, hurrengo komandoarekin:

**dot** -Gcharset=latin1 -Tgif -Tcmapx workflow.gv -O workflow.gif

Beraz, hurrengo egitura daukagu direktorio nagusian:



BATCH fitxategia exekutatu eta DOT zein workflow-lengoiaren arauak ondo jarraitu badira, hurrengo fitxategiak sortuko dira:



GIF fitxategia workflow-ereduaren irudia da eta cmapx fitxategia elementu nabigagarrien (azpi-workflowak) kokapenak adierazteko HTML MAP taula bat da. Beraz, orain HTML fitxategi bat sortu behar da, izen berdinarekin, eta gero, cmapx fitxategiaren HTML MAP taula kopia eta pegatu.

HTML fitxategiaren adibide bat:

```

<HTML>
<HEAD>
  <TITLE>Workflow</TITLE>
</HEAD>
<BODY>
  <IMG src="workflow.gv.gif" border="0" usemap="#workflow.gv.cmapx">
  <map id="CNAS" name="workflow.gv.cmapx">
    <area shape="rect" id="node1" href="/Azpi-workflow/azpi-workflow.html" title="Azpi-workflow" alt="" coords="150,137,234,185"/>
  </map>
</BODY>

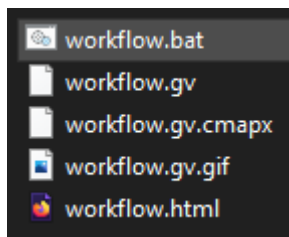
```

<map> etiketaren barruan dagoena cmapx fitxategitik kopia da.

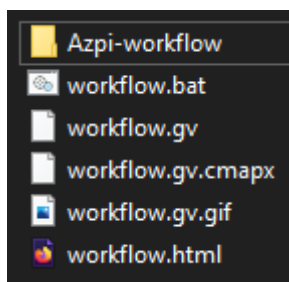
HTML fitxategia nabigatzaile baten zabalduz, workflow-ereduaren GIF irudia zabalduko da eta nabigagarria izango da. Baina, lehenengo, azpi-workflowaren eredua sortu behar da.



Beraz, orain honelako egitura daukagu:



Azpi-workflowa sortzeko, azpi-workflowaren nodoko URL atributuan jarri den kokapena kontua hartu behar da. Gomendatzen da azpi-workflowaren direktorio bere maila goreneko direktorioan kokatzea, adibidez:



Azpi-workflow horren barruan .gv bat sortu eta aurreko prozesu berdina errepikatu. Horrela, ereduen arteko nabigagarritasuna lortzen da, eta direktorioak ondo antolatuta gelditzen dira.

Horrela, workflow-ereduaren irudi nabigagarria lortu dugu, HTML lengoaian, nabigatzailean zabaltzeko.

## 4.2 Workflow-ereduen eraldaketa-prozesua

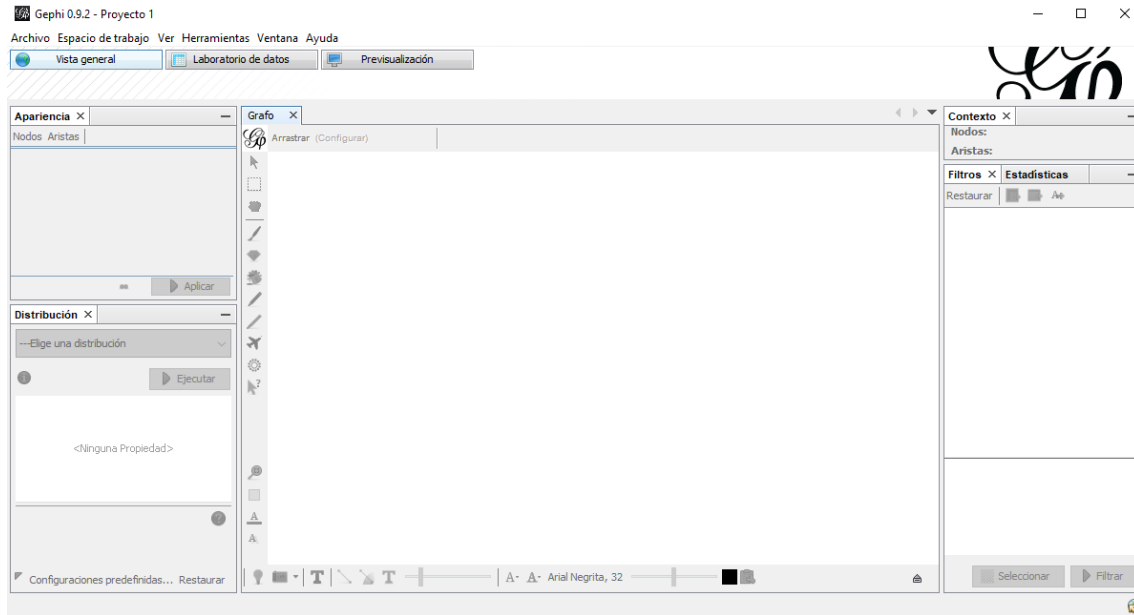
Behin, workflow-eredua eta bere irudia sortuta, workflow motorrak DOT lengoaian idatzitakoa interpretatu ahal izateko eta workflowaren informazioa datu-base erlazionalean sartzeko eraldaketa-prozesu bat egin behar da.

Eraldaketa-prozesu honen helburua DOT lengoaian idatzitako ereduak COOL lengoaiaren objektu bihurtzea da, EHSIS inferentzia motorrak erabiliko dituen klase eta instantziak izango dira.

Hona hemen eraldaketa-prozesuaren urratsak:

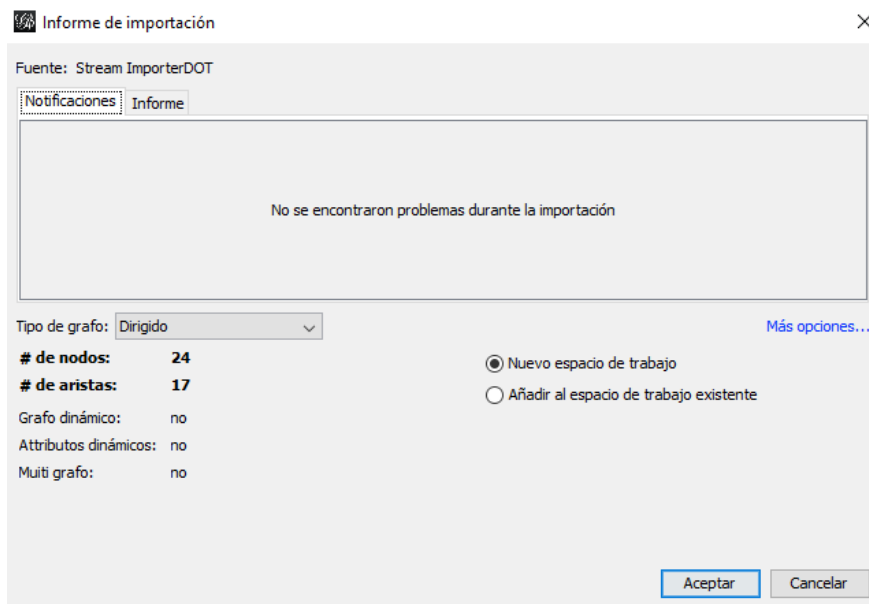
1. ***DOT* lengoian idatzitako ereduak *graphml* formatura eraldatzea, *Gephi* softwarea erabiliz. XML lengoian oinarrituta, *Graphml* lengoaiak grafiko edo diagramen datuak partekatzeko balio du.**

Horretarako, Gephi softwarea ireki. Bere hasierako orria zabalduko da:



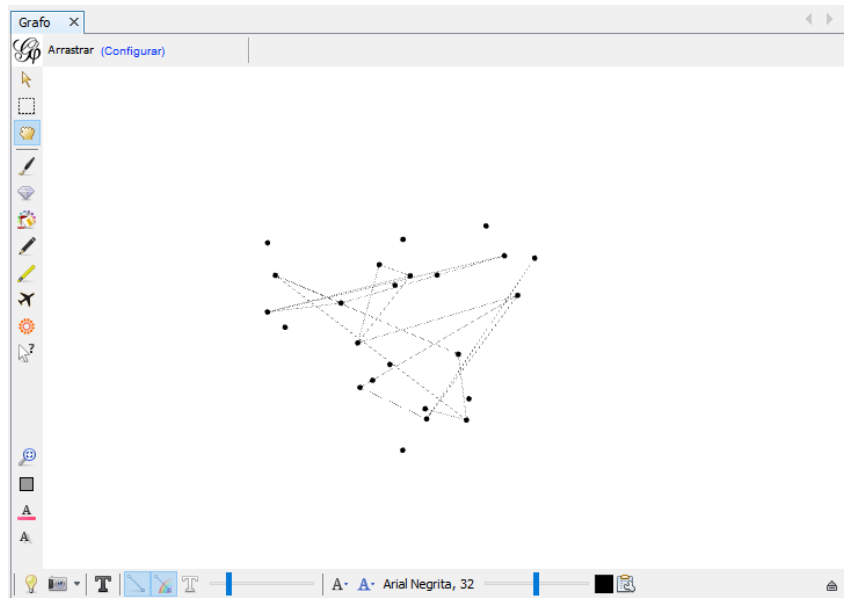
Ondoren, **Archivo** → **Abrir** eman eta hor aukeratu workflow-ereduaren **.gv** fitxategia.

Zabalduko den grafoaren informazioa agertuko da, bere nodoak eta erpinak adibidez.



Grafoa zabaltzeko aukera ezberdinak daude, baina ez dugu ezer aldatuko. **Aceptar** botoia sakatu.

Grafoaren irudia agertuko da, nodo eta erpinen arteko erlazioak irudikatuz:

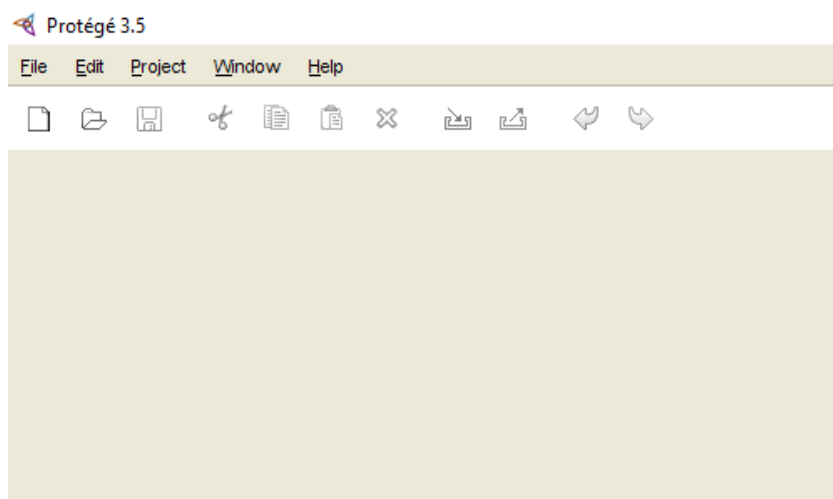


Azkenengo pausua, grafoa *graphml* formatura esportatzea da. Horretarako eman **Archivo → Exportar → Archivo de grafo...** eta ondoren, **aukeratu *graphml* formatua** eta kokalekua.

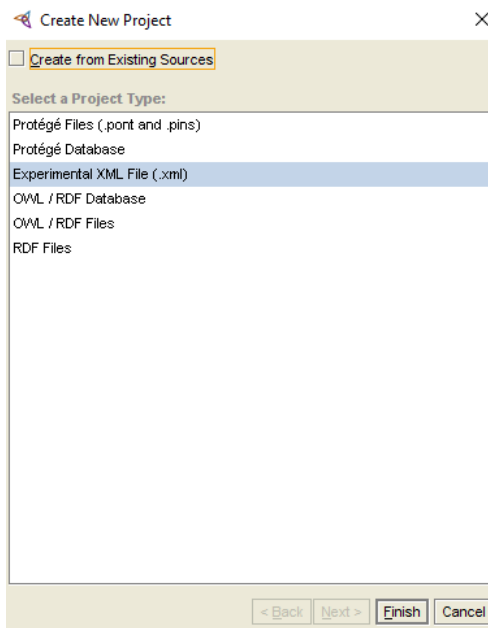
*Gephi* softwarearekin amaitu dugu.

2. **Graphml** formatuan eraldatutako eredu hori *Protégé* softwarearen bitartez klase eta instantzietan bihurtzea. Azken finean, *workflowan* agertzen den informazioa COOL lengoaiaren objektu bihurtzea du helburu.

Protégé softwarea ireki. Hurrengo da bere hasierako orria:

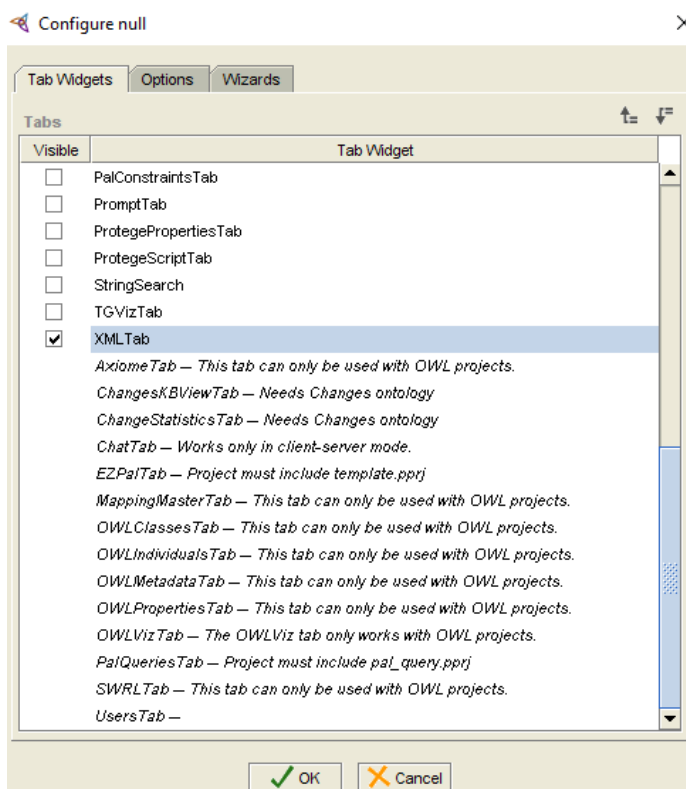


Proiektu bat sortu behar da, horretarako **File → New Project** eman, eta **Experimental XML File (.xml)** aukeratu:



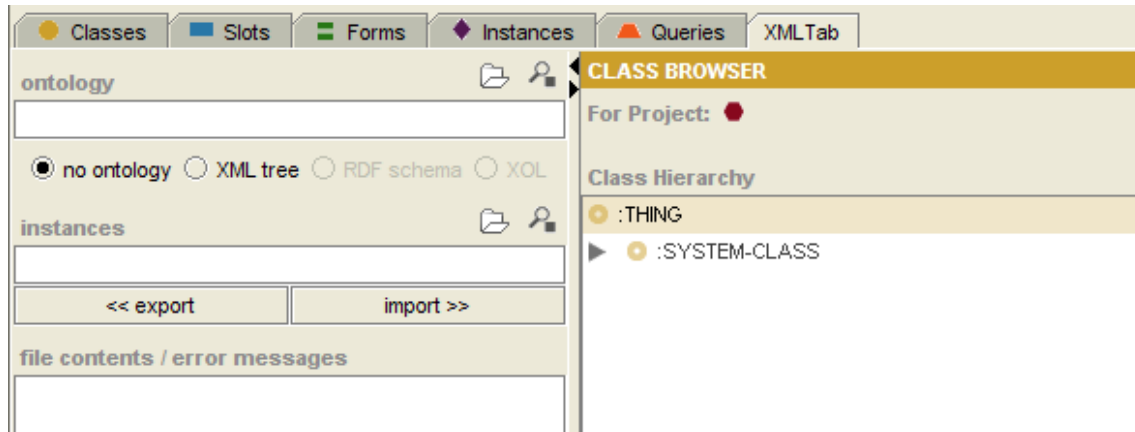
**Finish** botoia sakatu, proiektua sortzeko.


Ondoren, XML Tab modulua aktibatu behar dugu, horretarako **Project** → **Configure...** eman eta **XML Tab modulua** aukeratu.



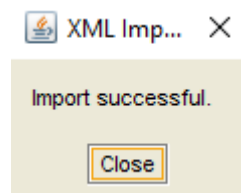
**OK** botoia sakatu eta XML Tab leihoa ikusi ahal izango da. Leiho honetan honako hau ikusiko da:



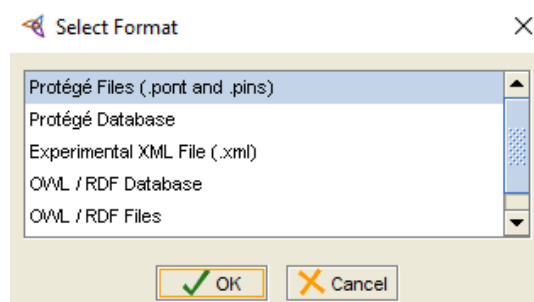


**Ontology** atalean  ikurraren gainean sakatu eta aurreko urratsean sortutako graphml fitxategia aukeratu.

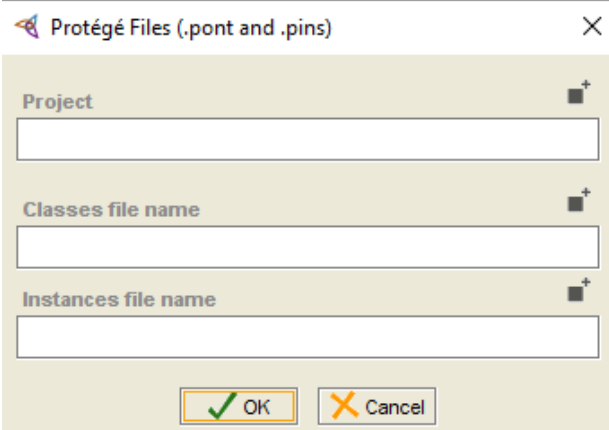
Berdina egin **instances** atalarekin. Ondoren, **import >>** botoia sakatu. Ondo inportatu bada, hurrengo mezua agertuko da:



**Close** botoia sakatu eta orain **File → Convert Project to Format...** eman eta aukeratu **Protégé Files (.pont and .pins)**.

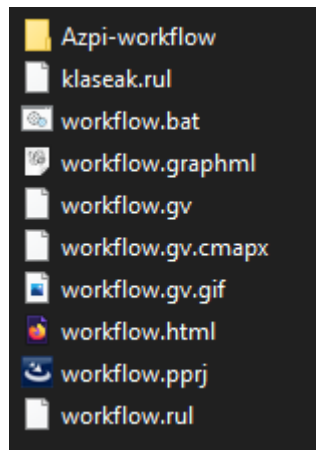


**OK** botoia sakatu eta hurrengo agertuko da:

Proiektuari izena eman, klaseen fitxategiari `_graphml_klaseak.rul` izena jarri eta instantzien fitxategiei `.rul` jarri amaieran. Horrekin, klase eta instantziak sortuko dira.

Urrats honetara helduta, guztira, fitxategi guzti honek sortuta egon behar dira:



*Protégé* softwarearekin amaitu dugu.

### 3. Sortutako klase eta instantziak EHSIS inferentzia motorrarekin exekutagarriak izateko egokitu.

Testu-editore baten bitartez, `_graphml_klaseak.rul` fitxategia editatu. Fitxategi horretan agertzen diren klaseen artean (**role concrete**) idatzita dutenetan (**pattern-match reactive**) gehitu behar da. Adibidez:

```

(defclass data
  (is-a USER)
  (role concrete) (pattern-match reactive)
  (single-slot Text
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write))
  (single-slot _key
    (type STRING)
;+    (cardinality 0 1)
    (create-accessor read-write)))
  
```



Ondoren, instantziak egokitu behar dira ere. Horretarako, kontuan izan behar instantzien fitxategi bat egongo dela workflow edo azpi-workflow bakoitzeko, beraz izenak ezin dira errepikatu. Hori dela eta, hurrengo testua gehitu behar da lehenengo lerroan:

#### **(definstances workflowaren\_izena**

Eta azkenengo lerroan “)” ikurra gehitu behar da.

Gainera, datuen izenak ez errepikatzeko izen berdina gehitu behar zaie hasieran. Adibidez:

[data\_0] → [workflowaren\_izena\_data\_0]

Hori egiteko “Buscar y reemplazar” bezalako erramintak erabiltzea gomendatzen da prozesua arintzeko.

Horrekin eraldaketa-prozesua amaitu da, klase fitxategi bat (workflow guztietan berdina) eta workflow zein azpi-workflow bakoitzeko instantzia fitxategi bat sorturik. Klase eta instantziak karpeta baten gordetzea gomendatzen da, hurrengo urratsak errazteko.

### 4.3 Workflow-ereduen informazioa datu-base erlazionalean gorde

Workflowaren informazioa datu-base erlazionalean gordetzea ezinbestekoa da workflowetan oinarritutako web-aplikazioak lan egiteko.

**Aurrebaldintza:** datu-base erlazionala (PROCESS\_DB) sortuta egon behar da XAMPP-eko datu-basean. Gainera, XAMPPen Apache eta MySQL moduluak abiarazita egon behar dira.

Lehenengo urratsa aurreko atalean sortutako **\_graphml\_klaseak.rul** fitxategia eta instantzia guztiak **WorkflowMotorra** direktoriaira eramatea izango da.

Ondoren, hurrengo fitxategiak editatu behar dira:

- **\_DFS\_Hurrengo\_Artefaktuak\_Sakoneran.bch**
- **\_DFS\_SakonerakoBilaketaGrafoan\_DBko-egitatetan.bch**
- **\_DFS\_Hurrengo\_Artefaktuak\_Sakoneran\_bin-sortu.bch**
- **\_DFS\_SakonerakoBilaketaGrafoan\_DBa-sortu.bch**

Hurrengo gehitu behar da:

```
;;-----  
KLASEAK  
  
  (load "_graphml_klaseak.rul")  
  
;;-----  
INSTANTZIAK  
  
(load "instantzia_baten_izena.rul") → Lerro bakoitzean instantzia bat
```

Amaitzeko, **WorkflowMotorra** direktorioan dauden fitxategi hauek eskuz exekutatu:

- **\_DFS\_Hurrengo\_Artefaktuak\_Sakoneran\_bin-sortu.bat**
- **\_DFS\_SakonerakoBilaketaGrafoan\_bin-sortu.bat**



**\_DFS\_SakonerakoBilaketaGrafoan.bin** eta **\_DFS\_Hurrengo\_Artefaktuak\_Sakoneran.bin** fitxategiak sortuko dira. Instantzia guztiekin sare bat eratu eta prozesua azkarrago exekutatzeko konpilatze binarioak dira.

Orain, prest gaude **\_DFS\_SakonerakoBilaketaGrafoan\_bin-exe.bat** fitxategia exekutatzeko. Hau egitean, instantzietan bananduta dagoen workflowaren informazioa datu-base erlazionalean automatikoki gordeko da.

#### 4.4 Web-aplikaziorako prestakuntza

Bizi-zikloaren workflow-ereduak IO-System azpisistemarekin lan egiteko, prestakuntza batzuk egin behar dira.

Hasteko, drupal karpeta barruan **...\drupal\sites\default\files** kokapenean workflow-ereduaren direktorio nagusia jarri edo kopiatu, HTML fitxategiekin. Horri ezker gero web-aplikaziotik bizi-zikloaren workflowaren irudi nabigagarria ikusi ahal izango da.

Ondoren, aurreko atalean komentatutako **WorkflowMotorra** direktorioa **...\drupal\sites\default\files** kokapenera eraman eta bere izena aldatu, **\_ProzesuaKorritu** jarritz. Bertan, workflowaren instantzia guztiak egongo dira. Hau eginez, gero prozesua exekutatu ahal izango da web-aplikaziotik eta bere sarrera/irteerak maneiatuko dira.

#### 4.5 Artefaktuen txantiloiak

Artefaktuen sekzio bakoitzaren txantiloia edo laguntza-testua web-aplikazioan agertzeko hurrengo prozedura jarraitu behar da:

- 1) Workflow-ereduan **SectionWT** mota duten nodoen izenetan **PATH** elementua lortu. Adibidez:

"Artifact SectionWT 1 1 2 3 5 7 4 Project milestones and objectives |  
**Templates/ProjectPlan/ProjectMilestonesAndObjectives.txt**"

- 2) Path edo kokaleku hori kontuan hartuz drupalen, direktorio berria sortu **...\drupal\sites\default\files** kokapenean. Direktorioa path bidean agertutako izen berdina izan behar du, kasu honetan, **Templates**.
- 3) Direktorioaren barruan sortu behar diren azpi-karpetak eta txantiloien .txt fitxategiak bertan pegatu.