



## Sarrera

Proiektu honen helburua, hiru mailako software arkitektura batean diseinatutako **produktuak saltzeko (eskaintzak)** aplikazio baten garapena izango da.

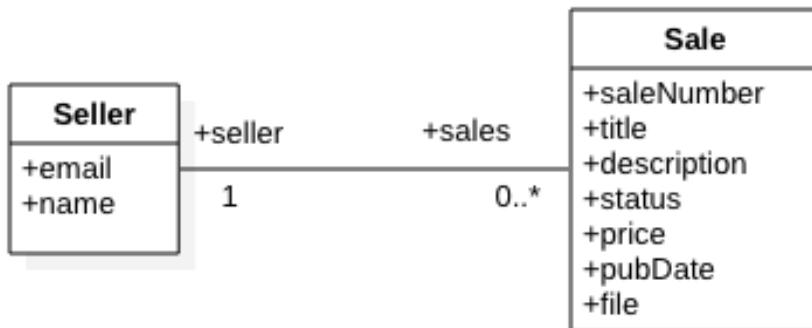
Dokumentu honetan orain arte garatutako informazioa ematen da (0 Iterazioa):

1. Aplikazioaren deskribapena.
2. Domeinuaren eredu.
3. Erabilpen kasuen eredu eta gertaera fluxuak.
4. "Create Sale" eta "Query Sales" erabilpen kasuen sekuentzia diagramak.
5. Iturburu kodea eta eclipse instalazioa.
6. Aplikazioaren egitura.
7. Aplikazioa martxan jartzeko argibideak.
8. Iturburu kodea.
9. Iterazio bakoitzean aurkeztu behar den gutxieneko dokumentazioa.

### 1. Aplikazioaren deskribapena

Orain arte garatutako aplikazioak, produktuen **eskaitzak sortzea** eta **eskaitzak konsultatzeko** aukera ematen du soilik. Hurrengo ataletan, 2 erabilpen-kasu hauen analisi/diseinua eta implementazioa deskribatzen dira.

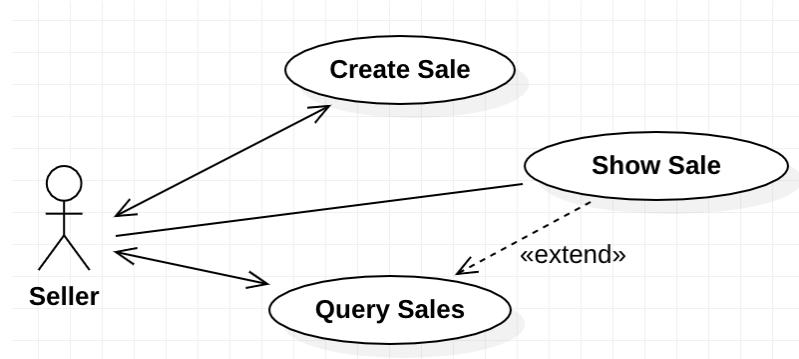
### 2. Domeinuaren eredu



Domeinuen ereduan **2 entitate** daude. **Seller** edo saltzailea eta **Sale** edo eskaitza, bakoitzak bere atributuekin.

**Saltzaile (Seller)** batek hainbat eskaitza (**Sale**) ditu (`0..*, sales`) eta eskaitza (**Sale**) bakoitza saltzaile (**Seller**) batena da (`1, seller`).

## Erabilpen Kasuen eredu



Oraingoz Seller aktorea daukagu soilik. Seller-ek eskaerak sortu (*Create Sale*) eta kontsultatu (*Query Sales*) *eta kontsultatu ondoren, nahi badu*, eskaera baten xehetasunak ikusi dezake (*Show Sale*).

## Gertaera fluxuak

### Create Sale Flow of events

#### Basic Flow

1. Seller inserts the **title**, **description**, **status** and **price**.
2. Seller selects **publication date**.
3. Seller chooses a **file**.
4. System creates a **Sale** with inserted data and assigns the **Sale** to the **Seller**.

#### Alternative flow (Sale is not created)

1. Some fields are empty. System informs the **Seller**.
2. **price** is not a number. System informs the **Seller**.
3. **publication date** is before today. System informs the **Seller**.
4. the **Sale** already exist for this **Seller** (same **title**). System informs the **Seller**.

### Query Sales Flow of events

#### Basic Flow

1. Seller inserts a **text**.
2. System displays the active **Sales**(**text**, **description**, **publicationDate**) that contains a **text** in a title.
3. If the Seller\*\* wants to see the details a of a Sale\* extends Show Sale.

#### Alternative flow

1. There are no active **Sales** with a **text** in a title, the message describing no sales situation is displayed.

### Show Sale Flow of events

#### Basic Flow

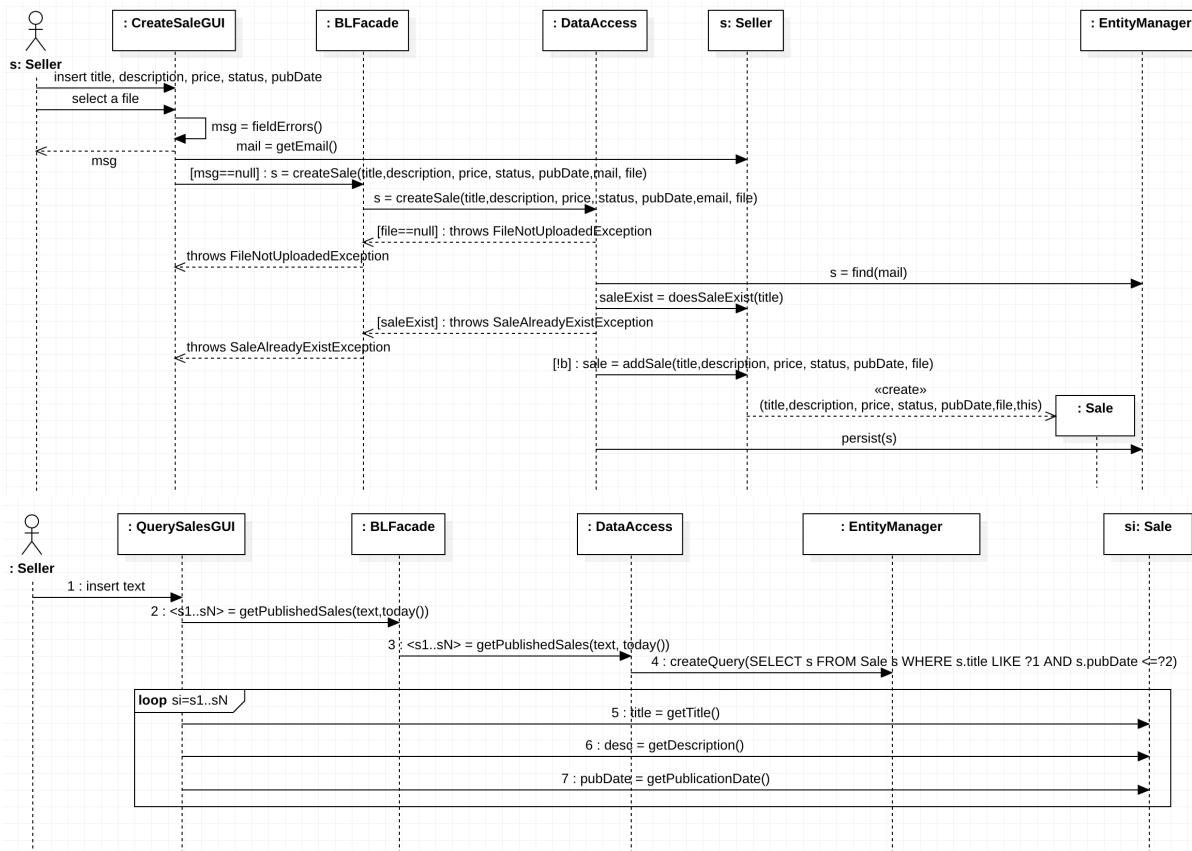
1. Seller inserts a **Sale key**.
2. System displays the information associated to a **Sale** with this **saleNumber** key (**title**, **description**, **status**, **price**, **publicationDate** and **file**).

#### Alternative flow

1. If there is any attribute with no value, the field will be empty.

#### 4. "Query Sales" eta "Create Sale" erabilpen kasuen sekuentzia diagramak

Aplikazioa hiru mailako arkitektura batean diseinatuta dago. AWT/SWING lengoia erabiltzaileren interfaze grafikoak definitzeko erabili da, negozio logikaren maila Java-n garatu da (Web Zerbitzuen bitartez ere garatu da), eta persistentzia, objektuetan oinarritutako objectdb datu basea erabili da. Negozio Logikaren eragiketa guztiak biltzeko, eta GRASP patroiaren gomendioak jarraituz, BLFacade interfaze bat definitu da eragiketa guziekin.

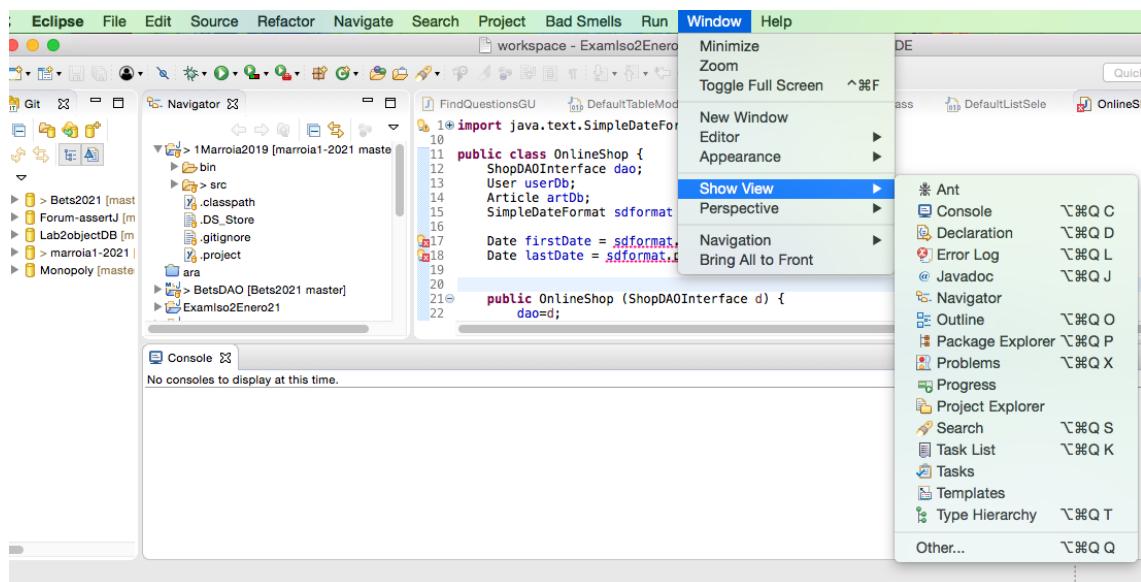




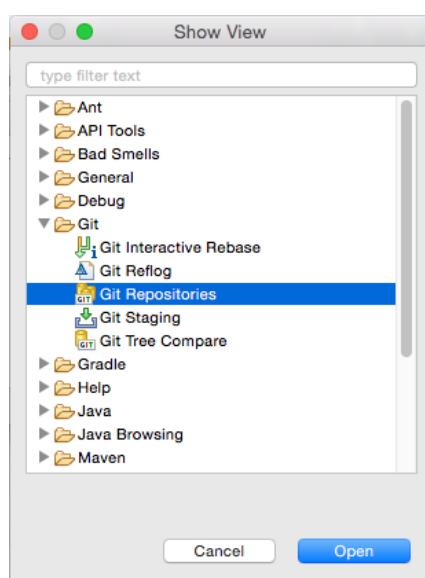
## 5. Iturburu kodea eta eclipse instalazioa.

Hasierako proiektuaren iturburu kodea <https://github.com/jononekin/Market26> helbidean dago. Lehendabizi helbide honetara joan, eta goiko partean dagoen botoia sakatuz eskatzen diren pausoak jarraitu proiektua zure kontuan kopiatzeko. Honen ondorioz, proiektuaren kopia bat izango duzu zure Github-eko kontuan.

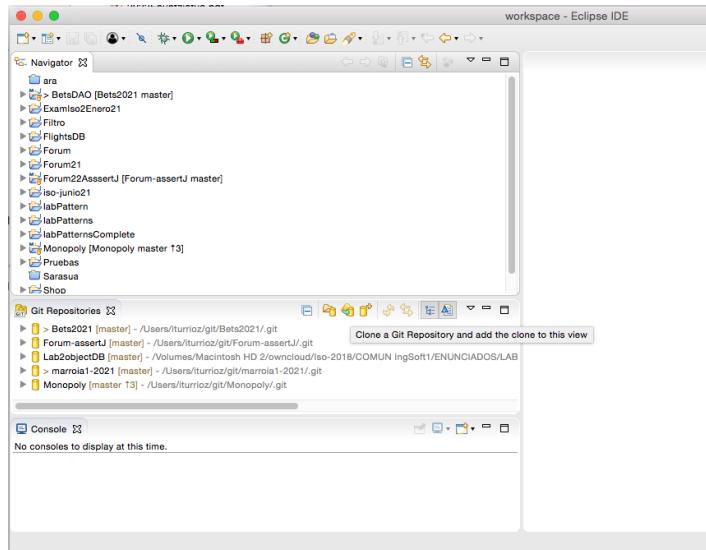
Kopiatutako proiektua Eclipsen instalatzeko, lehendabizi *Window-> Show View -> Other* aukeratu:



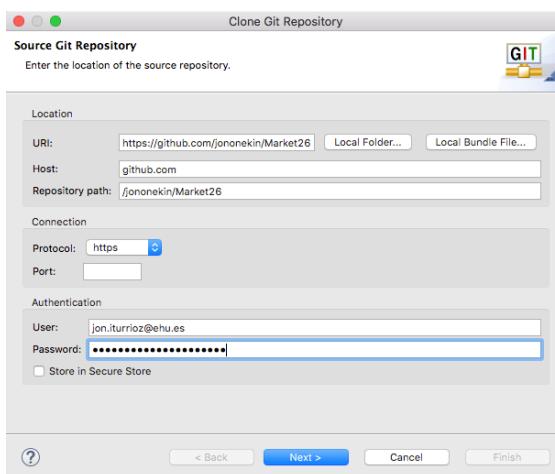
eta Git->Git Repositories bista ireki hurrengo irudian agertzen den bezala:



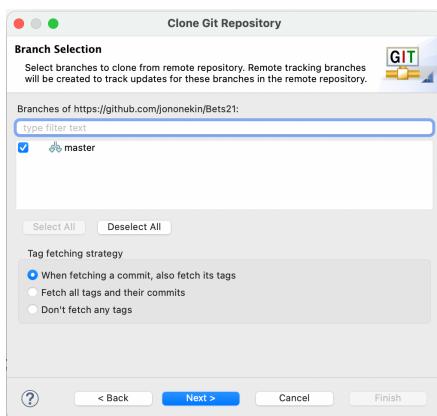
Jarraian, Eclipse-ko **Git Repositories** leihoa, **Clone a Git repository and add the clone to this view** botoia aukeratuz.



Kopiatu zuen urruneko errepositoriaren URL-a (zuen kontuan sortu duzuen kopien helbidea, hau da, [http://github.com/\\$zurelzena\\$/Rides24](http://github.com/$zurelzena$/Rides24))



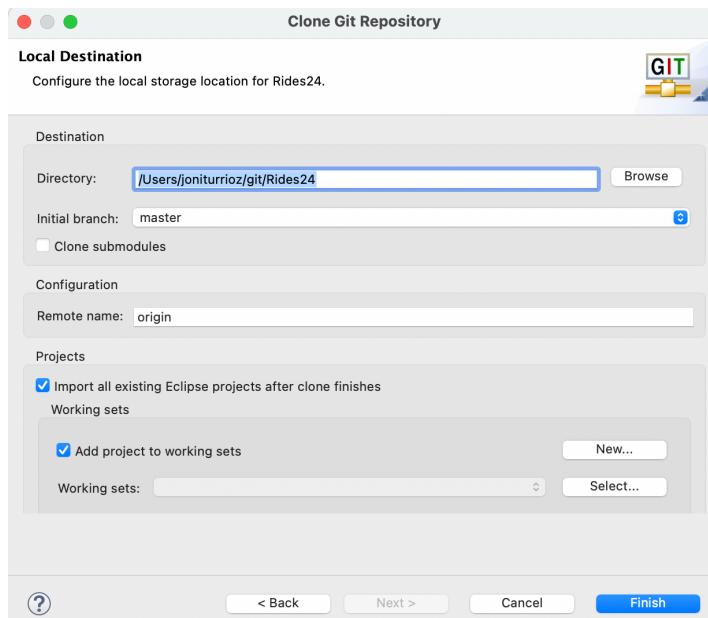
eta **Next** sakatu. Hurrengo **Branch Selection window** irudian **Next** egin



**Branch Selection** leihoa, urruneko errepositorioaren zein zati gure errepositorio lokalean kopiatu eta sinkronizatu nahi dugun adierazteko.



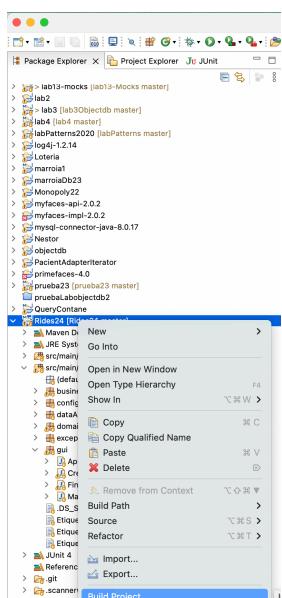
Eclipse-k zure lokal errepositorio non utzi jakin behar du (Urruneko errepositorioarekin sinkronizatzen den kopia), hau **Local Destination** leihoa agertzen da. **Oso garrantzitsua da** “Add project to working sets” aukeratzea Finish sakatu baino lehen.



**Local Destination** leihoa, klonatutako urruneko errepositorioaren helbidea adierazteko.

Honen ondorioz, proiekta zure eclipseko proietkuen artean agertuko da.

Bukatzeko, behar diren liburutegiak deskargatzeko, proiektauren erroan ipini, eskuineko botoia sakatu, eta “Build project” aukeratu.





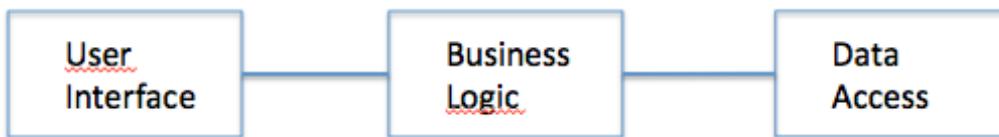
Github-eri buruz gehiago jakin nahi baduzu (adibidez, eclipsen egin dituzun aldaketak Github-era igo), egelan dagoen 8. laborategia kontsultatu dezakezu.

Proiektuan karpeta desberdinak egongo dira:

- (src/main/java) karpetan, proiektuaren Java iturburu kodea egongo da.
- (src/main/resources) karpetan proiektuak behar dituen baliabide osagarriak kokatuko dira, hau da, proiektuaren konfigurazio fitxategia (config.xml) eta datubasearen fitxategia (rides.temp)

## 6. Aplikazioaren egitura.

Aplikazioa hiru maila logikoetan banatuta dago: Interfaze grafikoa, Negozio Logika, eta Datu Atzipena.



Aplikazioa 5 java paketetan antolatuta dago:

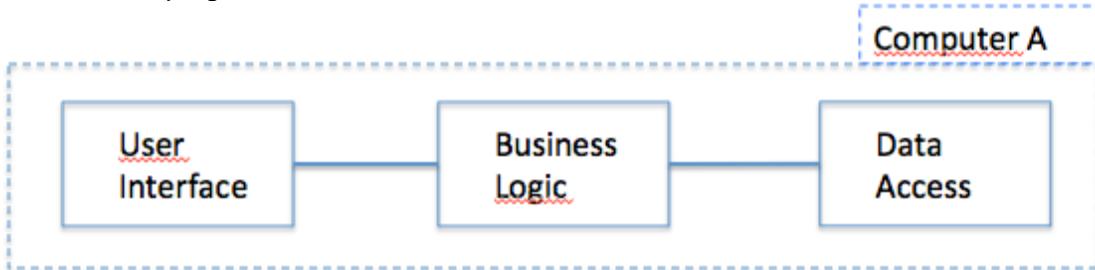
**gui**: Interfaze grafikoko klaseak (User Interface tier).  
**bussinessLogic**: Negozio logikako klaseak (Business Logic tier).  
**dataAccess**: Datu baseko klaseak (Data Access tier).  
**domain**: Domeinu ereduan definitu diren klaseak.  
**exception**: Salbuespenak definitzen dituzten klaseak.

Maila bakotza, makina berdinean edo beste ordenagailu desberdin batean exekutatu daiteke. Informazio hori config.xml fitxategian definitzen da. Ikus dezagun existitzen diren aukerak.

## 7. Aplikazioa martxan jartzeko argibideak.

### Aukera 1: Aplikazioa guztia maila fisiko batean.

Aukera honetan Aurkezpena, Negozio Logika eta Datu Atzipen mailak ordenagailu berdinean desplegatzen dira.



Horretarako, config.xml fitxategia honela konfiguratu behar da:

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <bussinessLogic local="true">
    <businessLogicNode>0.0.0.0</businessLogicNode>
    <businessLogicPort>1099</businessLogicPort>
```



```
<businessLogicName>Bets</businessLogicName >
</businessLogic>
<database local="true" initialize="true">
    <databaseNode>0.0.0</databaseNode>
    <dbFilename>bets.temp</dbFilename>
    .....
</database>
</config>
```

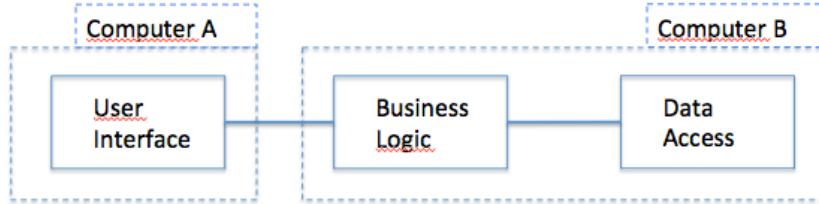
Egoera honetan, aurkezpenak negozio logika (**<bussinessLogic>**) modu lokalean atzitzen duela (**local="true"**) eta negozio logikak datu basea (**<database>**) modu lokalean ere atzitzen duela (**local="true"**) erazagutzen dugu.

**Ariketa:** config.xml fitxategia konfiguratu eta ondoren *gui* paketean dagoen **ApplicationLauncher** klasea exekutatu.



## Aukera 2: Aplikazioa bi maila fisikoetan.

Aukera honetan Aurkezpena maila makina batean, eta Negozio Logika eta Datu Atzipen mailak beste ordenagailu desberdinean desplegatzen dira.



### Aplikazioa martxan jartzeko.

1. Aplikazioaren konfigurazioa definitu config.xml fitxategia aldatuz:

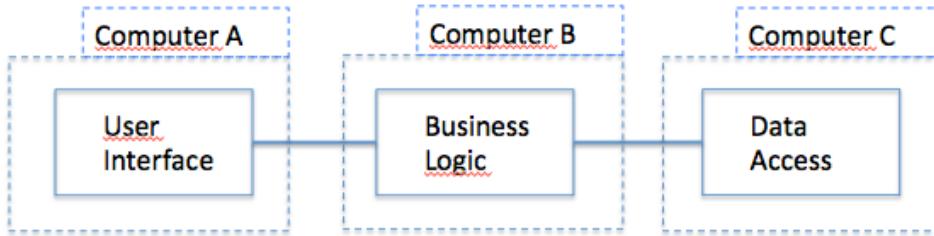
```
<?xml version="1.0" encoding="UTF-8"?>
<config>
    <businessLogic local="false">
        <businessLogicNode>0.0.0.0</businessLogicNode>
        <businessLogicPort>1099</businessLogicPort>
        <businessLogicName>Bets</businessLogicName >
    </businessLogic>
    <database local="true" initialize="true" >
        <databaseNode>0.0.0.0</databaseNode>
        <dbFilename>bets.temp</dbFilename>
    .....
    </database>
</config>
```

Konfigurazio honetan, bussinessLogic lokalean ez dagoela erazagutzen dugu (**businessLogic local="false"**). Lokalean ez dagoenez, negozio logika non kokatuta dagoen erazagutu behar da, hau da: a) Zein makinatan dagoen negozio logika (**businessLogicNode**), (0.0.0.0. bertan gauden konputagailuaren helbidea adierazten du), zein portutan (**businessLogicPort**) eta zein da zerbitzuaren izena (**businessLogicName**).

2. Negozio logika exekutatu. Horretarako *businessLogic* paketean dagoen **BusinessLogicServer** klasea exekutatu. Negozio Logikaren zerbitzaria martxan ipiniko da (zerbitzu bat bezala).
3. Aurkezpena exekutatu, hau da, aurreko *gui* paketearren **ApplicationLauncher** klasea exekutatu eta aurreko erabilpen kasuak frogatu.

**Ariketa:** Nola exekutatuko zenuke beste ordenagailu batean dagoen negozio logika? Bi ordenagailuen artean froga egin.

### Aukera 3: Aplikazioa hiru maila fisikoetan.



Aukera honetan Aurkezpena maila ordenagailu batean, Negozio Logika beste batean eta Datu Atzipen maila beste ordenagailu batean desplegatzen dira.

#### Aplikazioa martxan jartzeko.

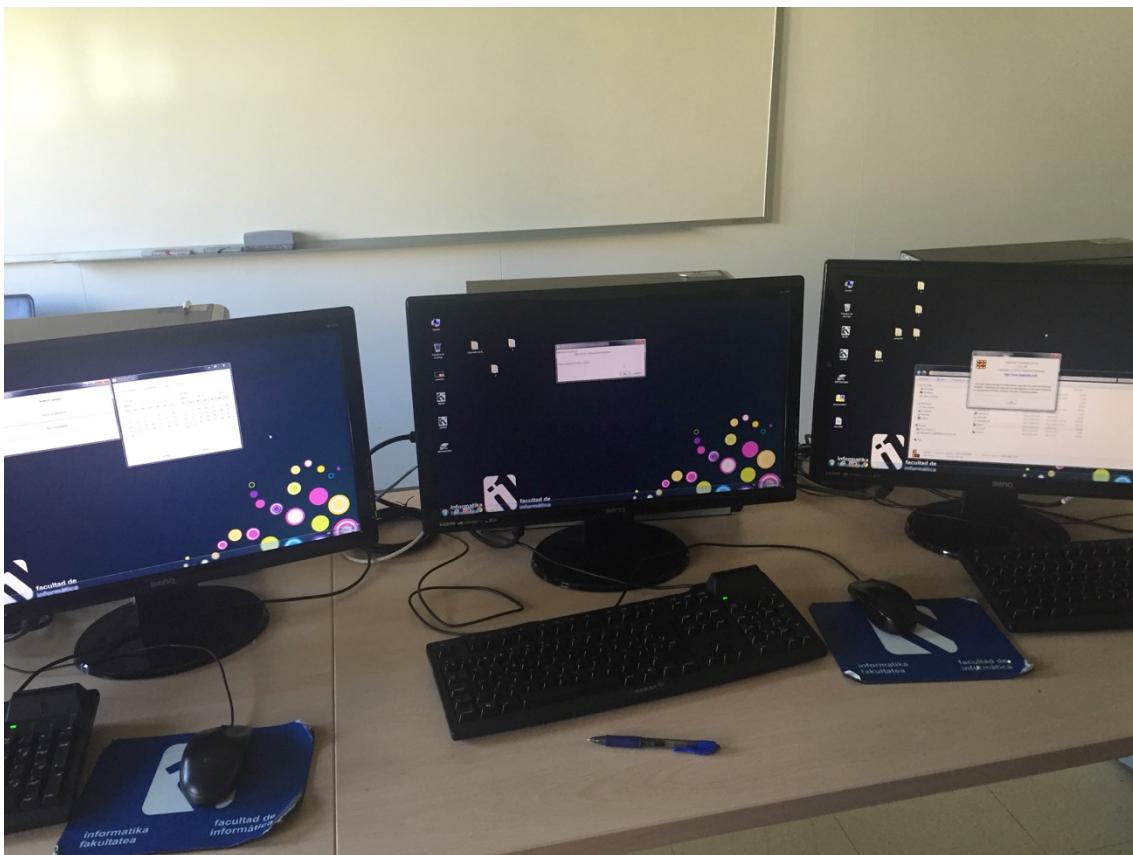
1. Aplikazioaren konfigurazioa definitu config.xml fitxategia aldatuz `<database local="true">` jartzen duen lekuan `<database local="false">` ipiniz.

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
    <businessLogic local="false">
        <businessLogicNode>0.0.0</businessLogicNode>
        <businessLogicPort>1099</businessLogicPort>
        <businessLogicName>Bets</businessLogicName >
    </businessLogic>
    <database local="false" initialize="true" >
        <databaseNode>0.0.0</databaseNode>
        <dbFilename>bets.temp</dbFilename>
        .....
    </database>
</config>
```

2. Datubase zerbitzaria exekutatu. Horretarako, *dataAccess* paketean dagoen **ObjectdbManagerServer** exekutatu. Datubase zerbitzaria martxan ipiniko da (zerbitzu bat bezala).
3. Negozio logikaren zerbitzaria exekutatu, aurreko *businessLogic* paketean **BusinessLogicServer** klasea exekutatzu.
4. Bukatzeko, aurkezpena exekutatu, hau da, *gui* paketearen **ApplicationLauncher** fitxategia.

#### Ariketa:

1. Nola desplegatuko zenuke Aurkezpena eta Negozio Logika ordenagailu batean eta Datu Basea beste makina desberdinean? Bi ordenagailuen artean froga egin.
2. Nola desplegatuko zenuke zure aplikazioa hiru ordenagailu desberdinietan? Azkenean, hurrengo argazkian agertzen den exekuzioa agertu beharko litzateke. Eskuineko ordenagailuan, datu zerbitzaria martxan. Erdiko ordenagailuan, negozio logikako zerbitzaria, eta ezkerrean interfaze grafikoa.





## 8. Iterazio bakoitzean aurkeztu behar den gutxienezko dokumentazioa.

Proiektoa irakasgaian zehar garatuko da 3 iterazioetan. Iterazio bakoitzeko aurkeztu beharko den beharrezko dokumentazioa hurrengoa da:

1. Aurkitutako arazoak. Zeintzuk dira eduki dituzuen arazo garrantzitsuenak eta nola bideratu dituzue.
2. Erabilpen kasu hedatuen eredu.
3. Domeinu eredu hedatua.
4. Diseinatutako erabilpen kasuen UML sekuentzia diagramak.
5. Diseinuzko klaseak
6. Implementazioaren iturburu-kodea formatu elektronikoan. Ez da inprimatu behar.