

Sarrera

Proiektu honen helburua, hiru mailako software arkitektura batean diseinatutako **bidaiak konpartitzeko** aplikazio baten garapena izango da.

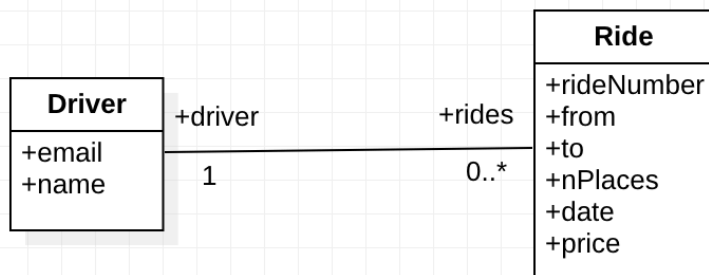
Dokumentu honetan orain arte garatutako informazioa ematen da (0 Iterazioa):

1. Aplikazioaren deskribapena.
2. Domeinuaren eredua.
3. Erabilpen kasuen eredua.
4. "Create Ride" eta "Query Rides" 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, bidaiak sortzeko eta dauden bidaiak kontsultatzeko aukera ematen du soilik. Hurrengo ataletan, 2 erabilpen-kasu hauen analisi/diseinua eta inplementazioa deskribatzen dira.

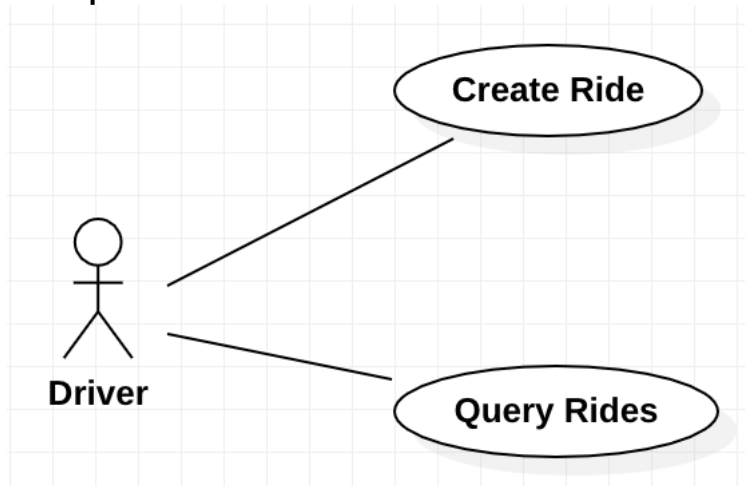
2. Domeinuaren eredua



Domeinuen ereduan **2 entitate** daude. *Driver* edo gidariak eta *Ride* edo bidaiak, bakoitzak bere atributuekin.

Driver batek hainbat (`0..*`) *Ride* eskaintzen ditu (*rides*) eta *Ride* bat driver batena (`1`) da.

Erabilpen Kasuen eredua



Oraingoz *Driver* aktorea daukagu soilik, eta *Create Ride* eta *Query Rides* erabilpen kasuak exekutatu dezake.

Gertaera fluxuak

Query Rides Flow of events

Basic Flow

1. System displays all **cities** where **rides** depart from
2. *Driver* selects a departing **city**
3. System displays all destination **cities** for **rides** that depart from a **selected city**.
4. *Driver* selects an arrival **city**
5. System highlights in a Calendar the days where **rides** exist from the **depart** to destination **cities** in a selected month
6. *Driver* selects a **date** in a Calendar
7. System displays the **rides** from the selected **departin city** to the selected **arrival city** on that **date**.

Alternative flow

1. There are no **rides** on the selected **date**. **Rides** display is empty.

Create Ride Flow of events

Basic Flow

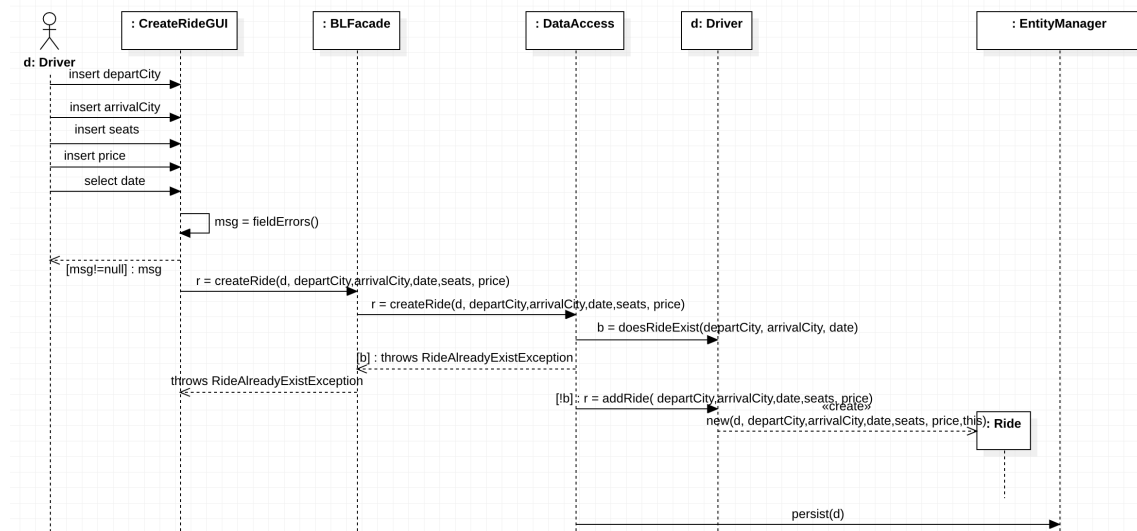
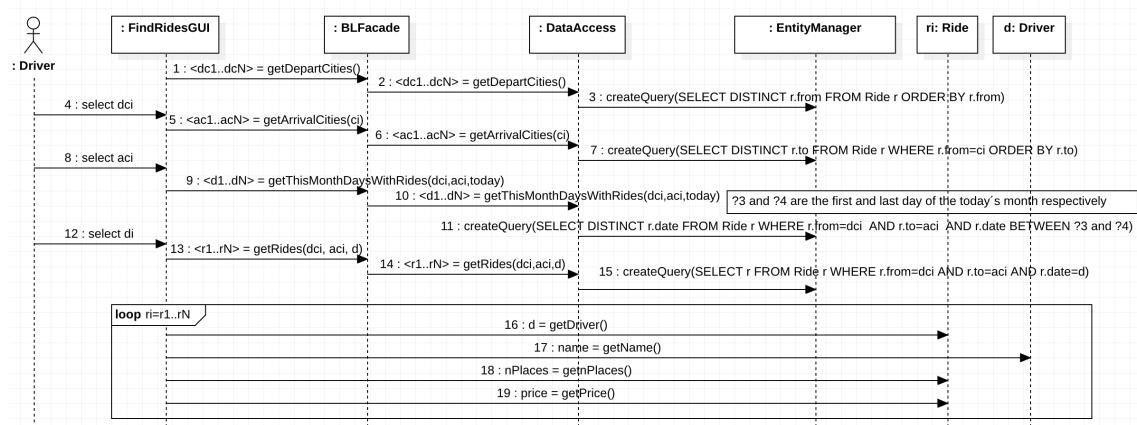
1. *Driver* insert the **departing city**, **arrival city**, **number of seats** and **price**
2. *Driver* selects a **ride date**
3. System creates a **ride** with inserted data and assigns to the **Driver**.

Alternative flow


1. Some of fields are empty. Final.
2. **number of seat** or **price** is not a number. System informs the user.
3. **Ride date** is before today. System informs the user.
4. the **ride** already exist for this **driver**. Ride is not created. System informs the user.

4. "Query Rides" eta "Create Ride" erabilpen kasuen sekuentzia diagramak

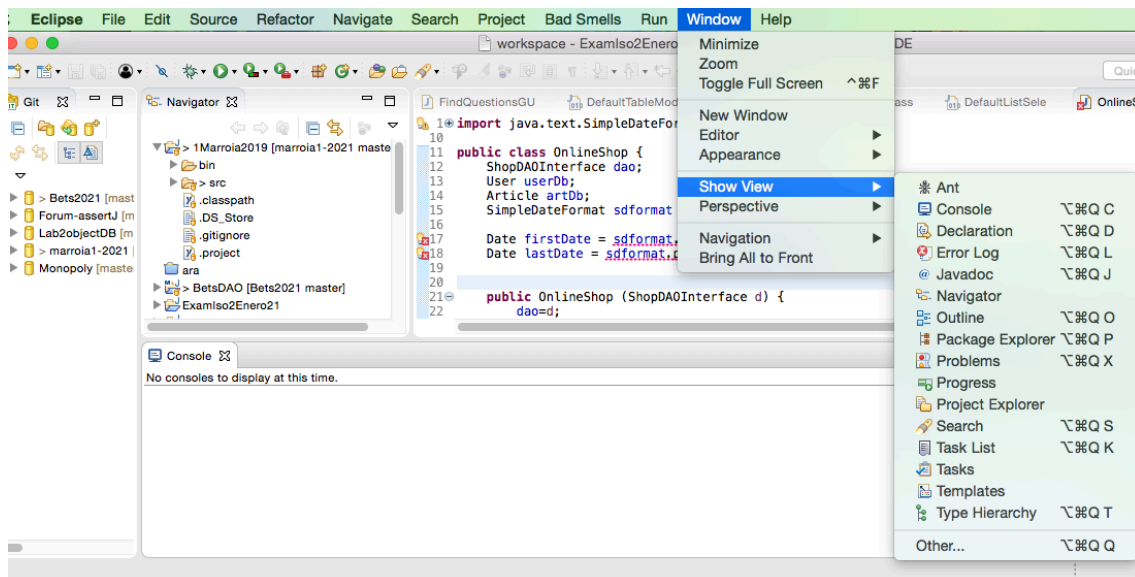
Aplikazioa hiru mailako arkitektura batean diseinatuta dago. AWT/SWING lengoia erabiltzailearen interfaze grafikoak definitzeko erabili da, negozio logikaren atzipena Web Zerbitzuren bitartez garatu da, eta persistentzia mailarako, objektuetan oinarritutako objectdb datu basea erabili da. Negozio Logikaren eragiketa guztiak biltzeko, eta GRASP patroia gomendioak jarraituz, Facade klase bat definitu da eragiketa guztiekin.



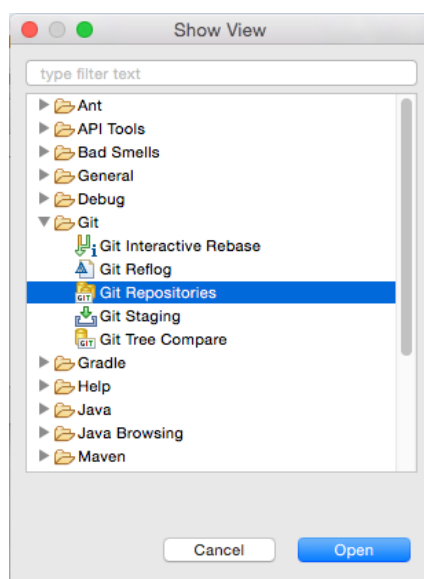
5. Iturburu kodea eta eclipse instalazioa.


Hasierako proiektuaren iturburu kodea <https://github.com/jononekin/Rides24> dago. Lehendabizi helbide honetara joan, eta goiko partean dagoen  botoia sakatzuz eskatzen diren pausoak jarraitu proiektua zure kontuan kopiatzeko. Honen ondorioz, proiektuaren kopia bat izango duzue zuen Github kontuan.

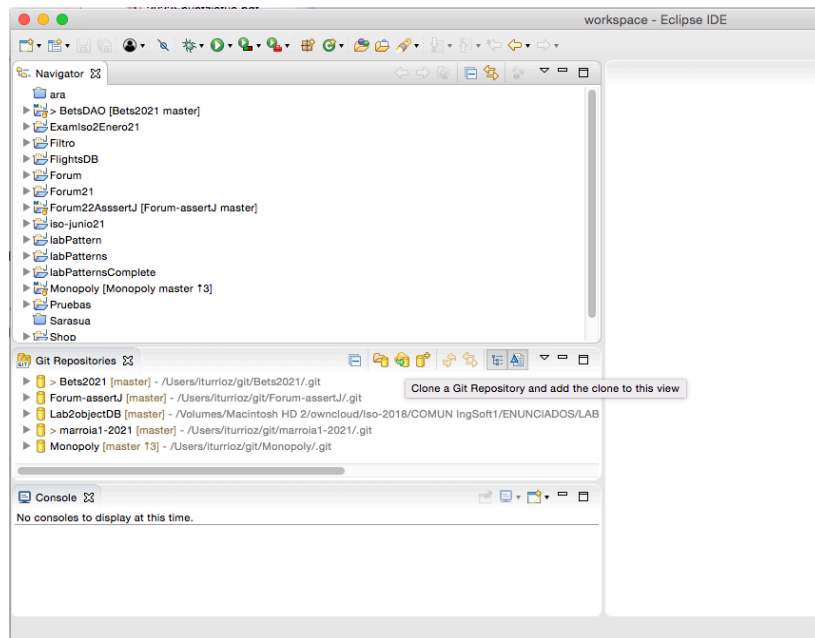
Kopiatutako proiektua Eclipsen instalatzeko, lehendabizi *Window-> Show View -> Other* aukeratu:



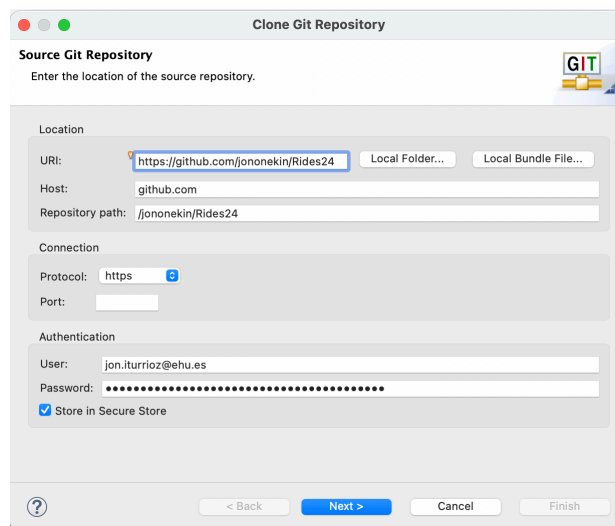
eta *Git->Git Repositories* bista ireki hurrengo irudian agertzen den bezala:



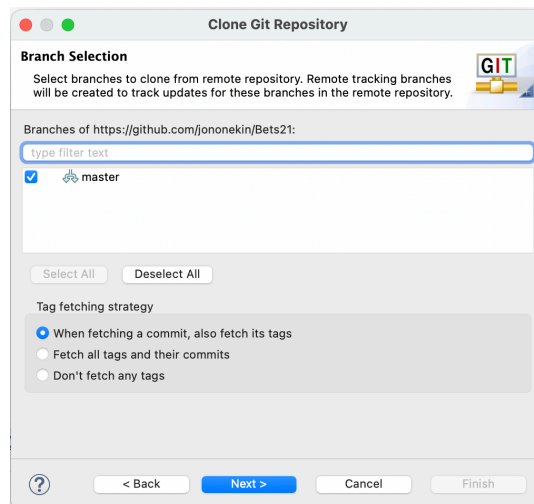
Jarraian, Eclipse-ko **Git Repositories** leihoan, **Clone a Git repository and add the clone to this view**  botoia aukeratzuz.



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

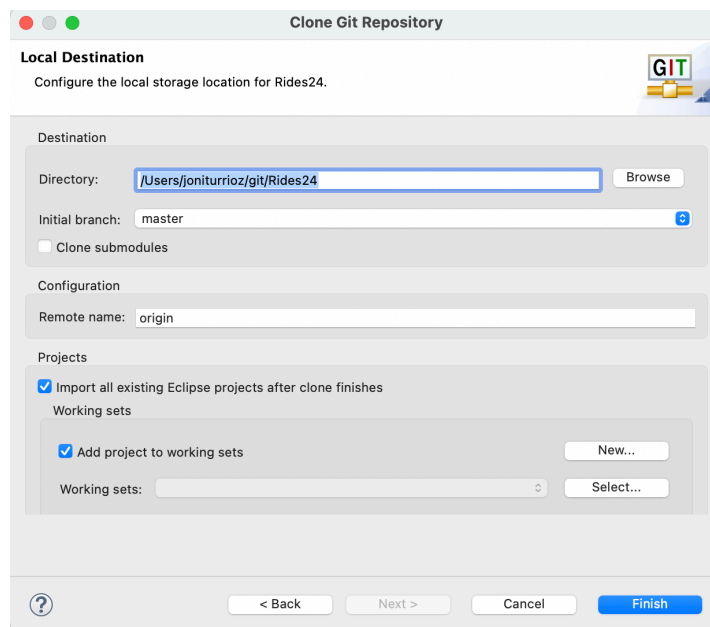


eta **Next** sakatu. Hurrengo **Branch Selection window** irudian **Next** egin.



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

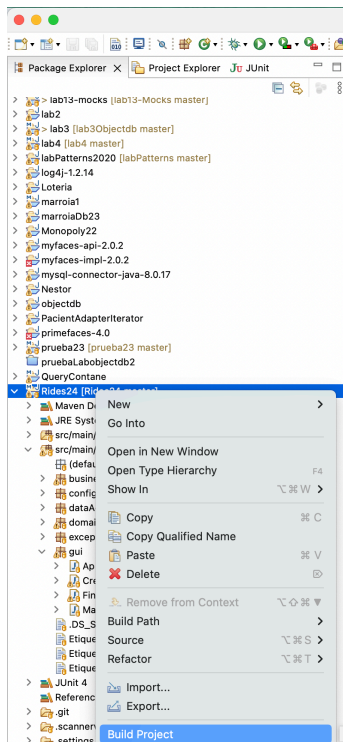
Eclipse-k zure lokal errepositorio non utzi jakin behar du (Urruneko errepositorioarekin sinkronizatzen den kopia), hau **Local Destination** leihoan agertzen da. **Oso garrantzitsua da** "Add project to working sets" aukeratzea Finish sakatu baino lehen.



Local Destination leihoa, klonatutako urruneko errepositorioaren helbidea adierazteko.

Honen ondorioz, proiektua zure eclipseko proiektuen artean agertuko da.

Bukatzeko, behar diren liburutegiak deskargatzeko, proiektuaren erroan ipini, eskuineko botoia sakatu, eta "Build project" aukeratu.



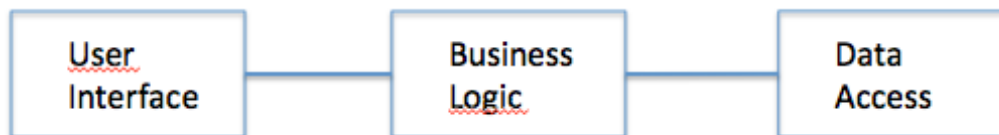
Github-eri buruz gehiago jakin nahi baduzu (adibidez, eclipsen egin dituzun aldaketak Github-era igo), egelan dagoen 8. laborategia kontsultatu dezakezu.

Proiektuan karpeta desberdinak egongo dira:

- (src/main/java) karpeta, proiektuaren Java iturburu kodea egongo da.
- (src/main/resources) karpeta 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).

businessLogic: Negozio logikako klaseak (Business Logic tier).

dataAccess: Datu baseko klaseak (Data Access tier).

domain: Domeinu ereduan definitu diren klaseak.

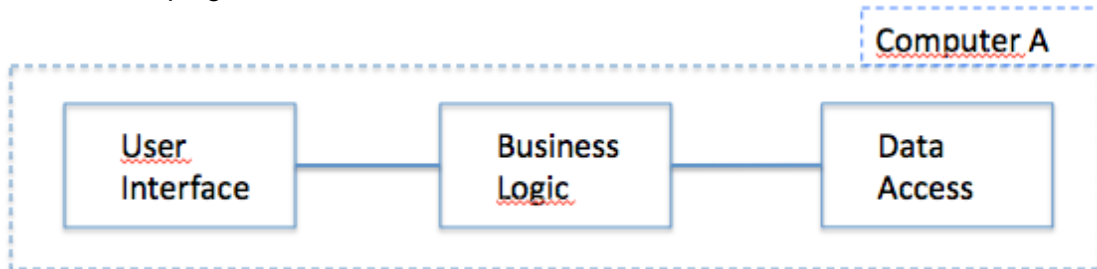
exception: Salbuespenak definitzen dituzten klaseak.

Maila bakoitza, 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>
  <businessLogic local="true">
    <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>

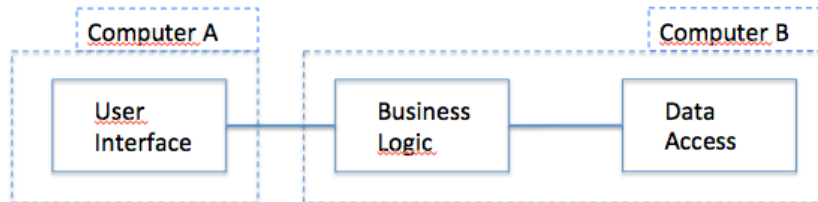
```

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* paketea 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:

```

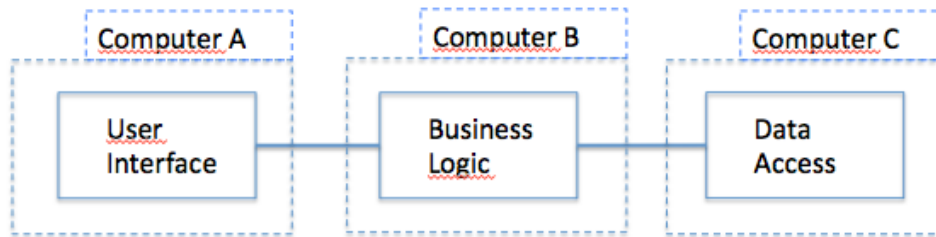
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <businessLogic local="false">
    <businessLogicNode>0.0.0.0</businessLogicNode>
    <businessLogicPort>1099</businessLogicPort>
    <businessLogicName>Bets</businessLogicName>
  </businessLogic>
  <database local="true" initialize="true">
    <databaseNode>0.0.0.0</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* paketearen dagoen **BusinessLogicServer** klasea exekutatu. Negozio Logikaren zerbitzaria martxan ipiniko da (zerbitzu bat bezala).
3. Aurkezpena exekutatu, hau da, aurreko *gui* paketearen **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.

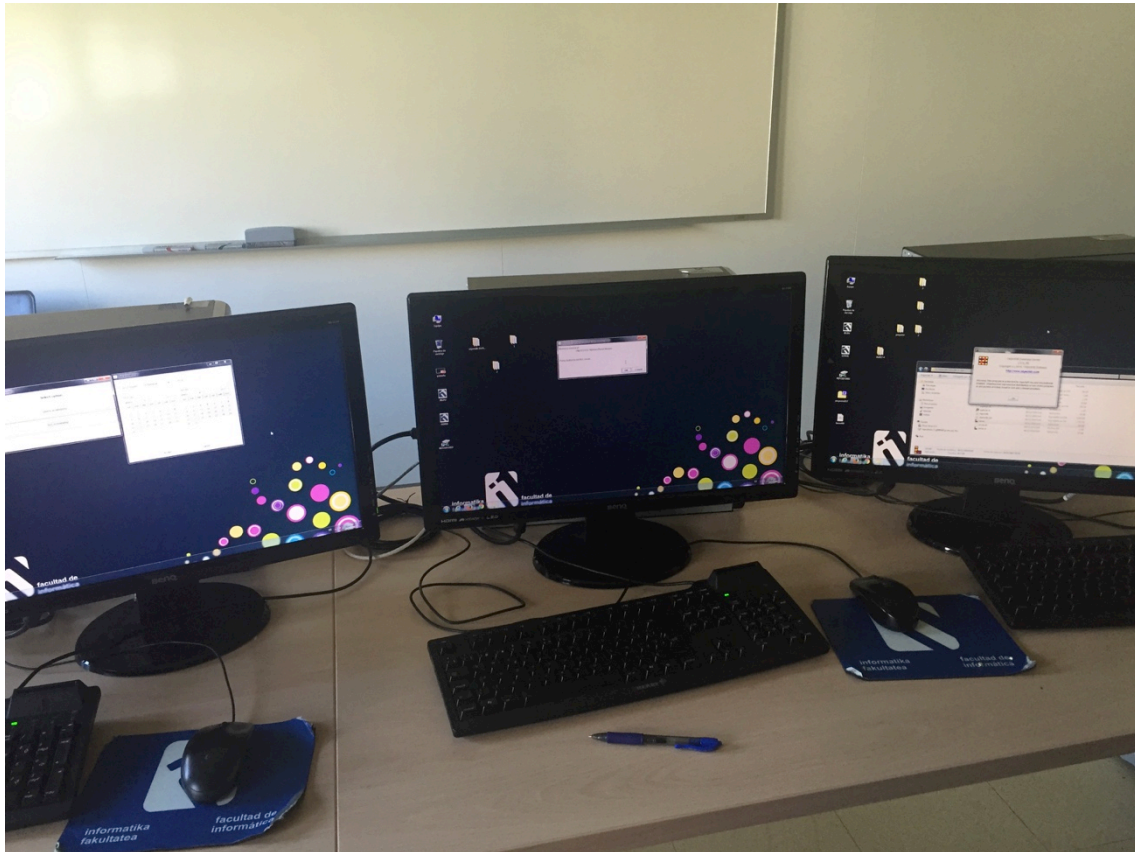
```

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

2. Database zerbitzaria exekutatu. Horretarako, *dataAccess* paketearen dagoen **ObjectdbManagerServer** exekutatu. Database zerbitzaria martxan ipiniko da (zerbitzu bat bezala).
3. Negozio logikaren zerbitzaria exekutatu, aurreko *businessLogic* paketearen **BusinessLogicServer** klasea exekutatu.
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 desberdinetan? 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.

Proiektua irakasgai zehar garatuko da 3 iterazioetan. Iterazio bakoitzeko aurkeztu behar den beharrezko dokumentazioa hurrengoa da:

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