

## Sarrera

Laborategi honetan, existitzen den apustu sistema baten implementazioa aurkeztuko da. Proiektu hau oinarri bezala hartuko duzue, apustu sistema konplexuago bat egiteko. Hau da Sinking Soft enpresak eskatu dizuedana

## Helburuak

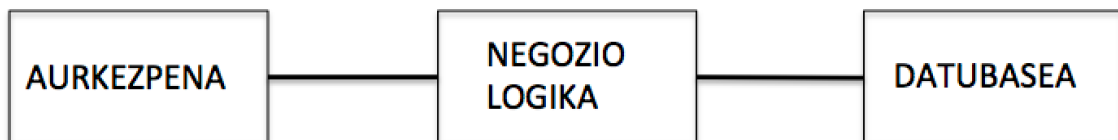
Laborategi honen helburuak hurrengoak dira:

- Uneko implementazioa nolakoa den ezagutu: Zein erabilpen kasu implementatu diren orain arte.
- Aplikazioaren arkitektura ezagutu: Aplikazioa osatzen duten geruza logikoak, eta nola desplegatu daiteke geruza bakoitza nodo fisiko desberdinetan.

Eta gainera: Software Garapen Prozesua Bateratua-ren oinarriak erabiltzen ari garela proiektua garatzeko ulertu, hau da: 1) **erabilpen kasuen**-gatik gidatuta dago, **arkitekturan** zentratua eta bizi-ziklo **iteratiboa eta inkrementala** jarraitzen du.

## Jarraitzeko pausoak

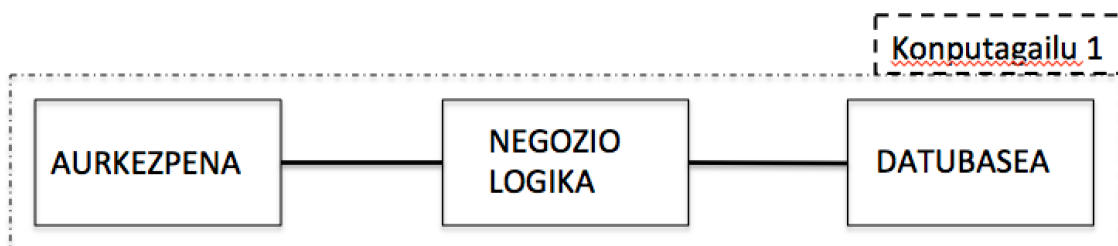
Garatu den aplikazio hiru maila logikoetan banatuta dago, hau da, Aurkezpena, Negozio Logika eta DatuBasea.



Hurrengo adibidetan ikusiko dugun bezala, hiru maila hauek, konputagailu desberdinetan desplegatu daitezke.

### 1. Aplikazioa konputagailu batean exekutatu.

Lehendabiziko adibidean hiru maila logikoak, ordenagailu batean exekutatu ditugu, hau da:



**Bets.zip** fitxategia deskargatu egela-tik, **M1 karpeta** sortu idazmahian eta bere barruan beste **Bets-M1** karpetan **Bets.zip** deskonprimitu.



**Bets-M1** karpetak Bets.jar fitxategi bat dauka, proiektua inplementatzen dituen klase guztiekin, internalizazio fitxategiak (.properties) eta beste konfigurazio fitxategi bat(config.xml).

Aplikazioa exekutatu bi aldiz klikatuz Bets.jar fitxategian.

Aplikazioak bi funtzionalitate desberdin eskaintzen dituela ikusiko duzue (**erabilpen kasuak** gure terminologia erabiliz). Frogatu zer egin daiteke eta zer ez. Baita ere ikusi dezakezue bets.temp fitxategi bat sortu dela objectDB datu basearekin. Beste aldetik hizkuntza ere aldatu daiteke.

Galdera: Zer gertatzen da aplikazioa itxi eta berriz irekitzen baduzue?

- ⇒ Aurreko datuak galdu direla (sortu dituzuen galderak) datubasea hasieratu delako hasierako datu batzuekin.
- ⇒ config.xml fitxategiaren konfigurazioa aldatu eta ikusi zer gertatzen den. Mantentzen dira datuak?

⇒

```
<dataBaseOpenMode>initialize</dataBaseOpenMode> ordeztu  
<dataBaseOpenMode>open</dataBaseOpenMode> ipini
```

Galdera: Sortutako galderak, beste ikasleek ikusi dezakete?

- ⇒ Ez, konputagailu bakoitzak modu lokalean exekutatzen ari delako bere datubasearekin, eta ez dago konpartituta besteekin.

#### **Idea garrantzitsua:**

BALIABIDE KONPARTITU (datuak, funtzionalitateak, aplikazioak, etab..) bat atzitu nahi badugu aplikazio/konputagailu desberdinetik, baliabidea gorde eta kudeatzen duen ZERBITZARI bat definitu behar da.

#### **SGPB-ari buruz gogoratu behar duzuenak:**

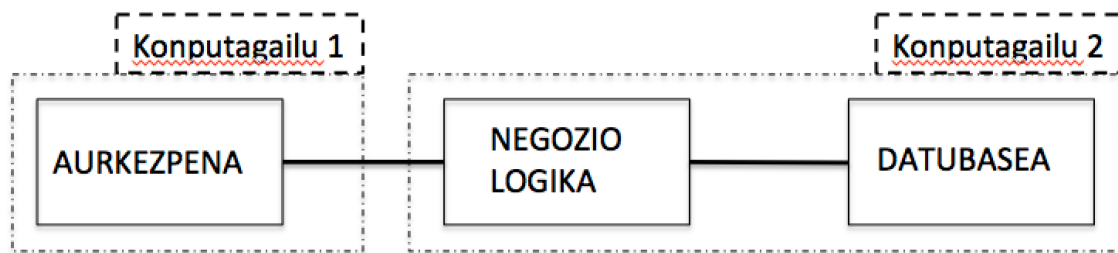
Proiektu hau ERABILPEN KASUEZ gidatuta dago. Orain arte 2 erabilpen kasuen eskakizunak, diseinua eta inplementazioa garatu da.

Proiektu honek bizi ziklo ITERATIBO eta INKREMENTALA jarraitzen du. Orain arte iterazio bat egin da eta produktu bat eskuragarri dago. Zuen lana beste 3 iterazioa garatzea izango da, iterazio bakoitzean aplikazioa funtzionalitate gehiagorekin aberastuz.

## **PUNTU HONETARA AILEGATU BAZARA, ITXARON MESEDEZ...**

### **2. Aplikazioa 2 mailako arkitektura batean exekutatu (auzkepena y NL+DB)**

Hurrengo aukeran, hiru maila logikoak bi konputagailuetan desplegatuko dira: Lehendabiziko konputagailuan aurkezpena desplegatuko dugu, eta beste konputagailuan negozio logika eta datuak.



Horretarako, M2 karpeta bat sortu idazmahian, eta bere barruan **Bets-M1** direktorioa bi aldiz kopiatu adibidez **Bets-Aurk-M2** y **Bets-NL+DB-M2** izenarekin.

Bi direktorioetan config.xml fitxategia aldatu: `<businessLogic local="true">` ipintzen duen lekuan `<businessLogic local="true">` ipini. Hemen esaten dugu, negozio logika ez dela lokalean exekutatu, urruneko zerbitzari batean baizik.

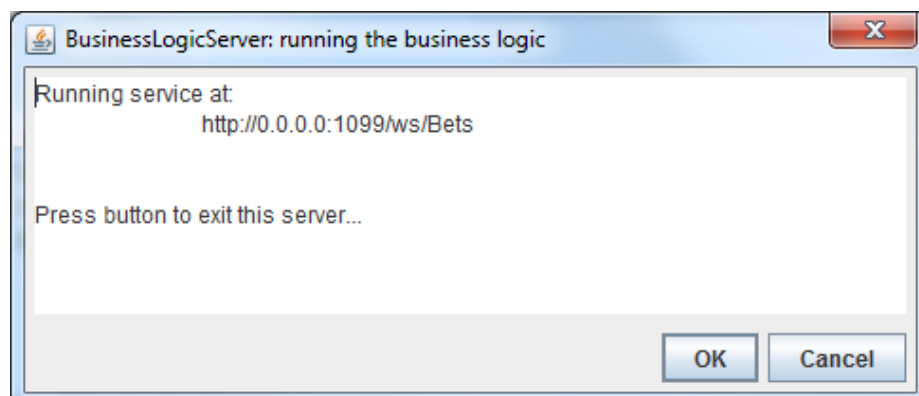
**Bets-NL+DB-M2** karpetan Bets.jar bi aldiz klikatzerakoan exekutatzen den gui.ApplicationLauncher funtzionalitatea businessLogic.BusinessLogicServer funtzionalitarenegatik aldatu behar da. Horretarako Bets.jar fitxategia deskonprimitu eta MANIFEST.MF fitxategian Main-Class eredu aldatu behar duzue:

```
Manifest-Version: 1.0
Class-Path: .
Main-Class: businessLogic.BusinessLogicServer
```

**Ohar:** .jar fitxategi bat .7z, winzip, winrar aplikazioekin deskonprimitu daiteke. META-INF karpetan dagoen MANIFEST.MF atera idazmahaira, fitxategia aldatu eta berriz Bets.jar fitxategira mugitzen duzue.

Orain aplikazioaren bi mailak exekutatu dezakezu, hau da, lehendabizi, **Bets-NL+DB-M2**-ren Bets.jar eta jarraian **Bets-Aurk-M2**-ren Bets.jar.

**Ohar:** "Press button to exit this server" mezua agertzen bazaizu, negozio logikaren zerbitzaria ondo exekutatu dela jakingo duzu



Galdera: Zer gertatzen da **Bets-Aurk-M2**-ren Bets.jar fitxategia exekutatzen badugu **Bets-NL+DB-M2**-ren Bets.jar baino lehenago?

Galdera: Aurreko datu konpartitze arazoa konpondu dugu?



- ⇒ Ez, konputagailu batean **Bets-NL+DB-M2**-ren Bets.jar exekutatu behar dugu, eta beste konputagailu guztietan **Bets-Aurk-M2**-ren Bets.jar fitxategia exekutatu dugu soilik bere negozio logikaren zerbitzaria modu egokian konfiguratuta.

Bezero guztietan (hau da aurkezpenetan) `<businessLogicNode>0.0.0.0</businessLogicNode>` aldatu behar da eta **0.0.0.0** ipini ordez, **Bets-NL+DB-M2**-ren Bets.jar fitxategia exekutatu den konputagailuaren IP-a ipini behar da.

Ohar: Konputagailu baten IP-a jakiteko: Inicio => exekutatu => cmd => ipconfig

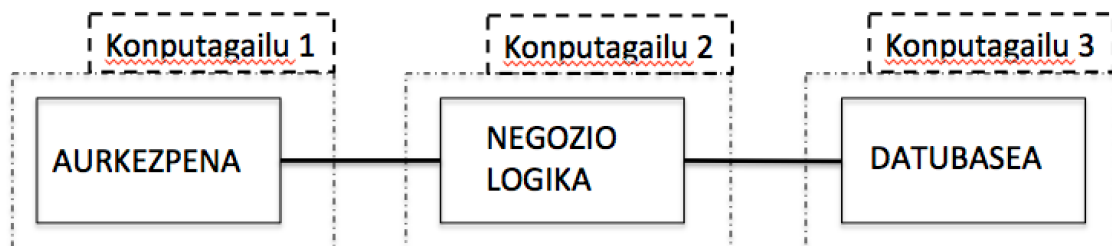
### **SGPB-ari buruz gogoratzeko kontzeptua:**

Proiektu hau ARKITEKTURAN ZENTRATUA garatu da, despliege desberdinak egiteko aukera eskaintzen dituelako.

## **PUNTU HONETARA AILEGATU BAZARA, ITXARON MESEDEZ...**

### **3. Aplikazioa 3 mailako arkitektura batean exekutatu (auzkepena, NL eta DB)**

Azkeneko implementazioan, maila bakoitza konputagailu desberdin batean desplegatuko da.



M3 karpeta bat sortu idazmahian, eta bere barruan **Bets-M1** direktorioa hiru aldiz kopiatu adibidez **Bets-Aurk-M3**, **Bets-NL-M3** eta **Bets-DB-M3** izenarekin.

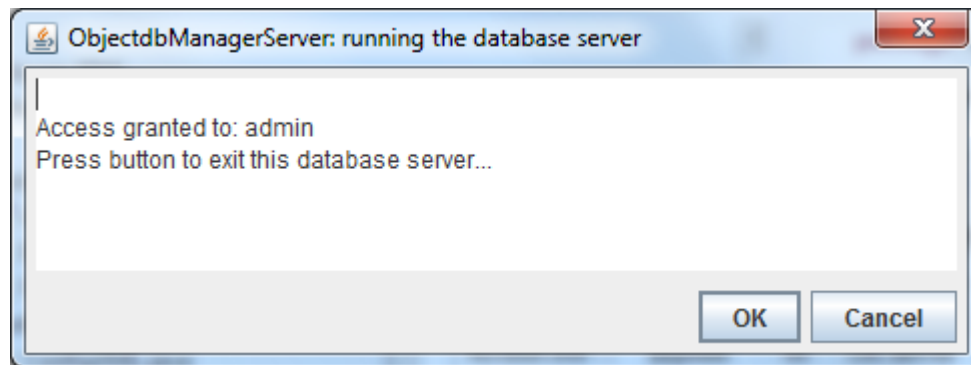
Guztien config.xml fitxategian hurrengoak ipini: `<businessLogic local="false">` y `<database local="false">`, negozio logika eta datubasea lokalak ez direla erazagutzeko.

**Bets-NL-M3**-ren Bets.jar MANIFEST.MF fitxategian ipini:  
Main-Class: businessLogic.BusinessLogicServer

**Bets-DB-M3**-ren Bets.jar MANIFEST.MF fitxategian ipini::  
Main-Class: dataAccess.ObjectdbManagerServer

Hiru despliegeak exekutatu ordenan, hau da, lehendabizi, **Bets-DB-M3**-ren Bets.jar, jarraian **Bets-NL-M3**-rena eta azkenik **Bets-Aurk-M3**-rena.

Ohar: “Press button to exit this database server” mezua agertzen bazaizu, datubasearen zerbitzaria ondo exekutatu dela jakingo duzu.



Galdera: Zer gertatzen da despliegeak beste orden batean exekutatzen badira?

Bukatzeko, ez ahaztu, aplikazioa hiru konputagailuetan desplegatzeko, negozio logika eta database zerbitzaria desplegatura dauden konputagailuen IP-ak konfiguratuz egiten dela.

Horretarako:

- 1) Bezero guztien `<businessLogicNode>0.0.0.0</businessLogicNode>` aldatu behar da eta **Bets-NL-M3** exekutatzen ari den Bets.jar konputagailuen IP-a ipini behar da.
- 2) Negozio Logika guztien `<databaseNode>0.0.0.0</databaseNode>` aldatu behar da eta **Bets-DB-M3** exekutatzen ari den Bets.jar konputagailuen IP-a ipini behar da.

Puntu honetan hurrengo egin daiteke: konputagailu batek datu zerbitzaria exekutatzen du, bi konputagailu negozio logika exekutatzen dute aurreko datu zerbitzarira konektatuta, geratzen diren konputagailuak, erdiak aurkezpena exekutatzen dute negozio logika zerbitzari batera konektuta, eta beste erdiak, beste zerbitzarira.

Proieku honen kodea hurrengo helbidean eskuragarri dago:

<https://github.com/jononekin/Bets21>