

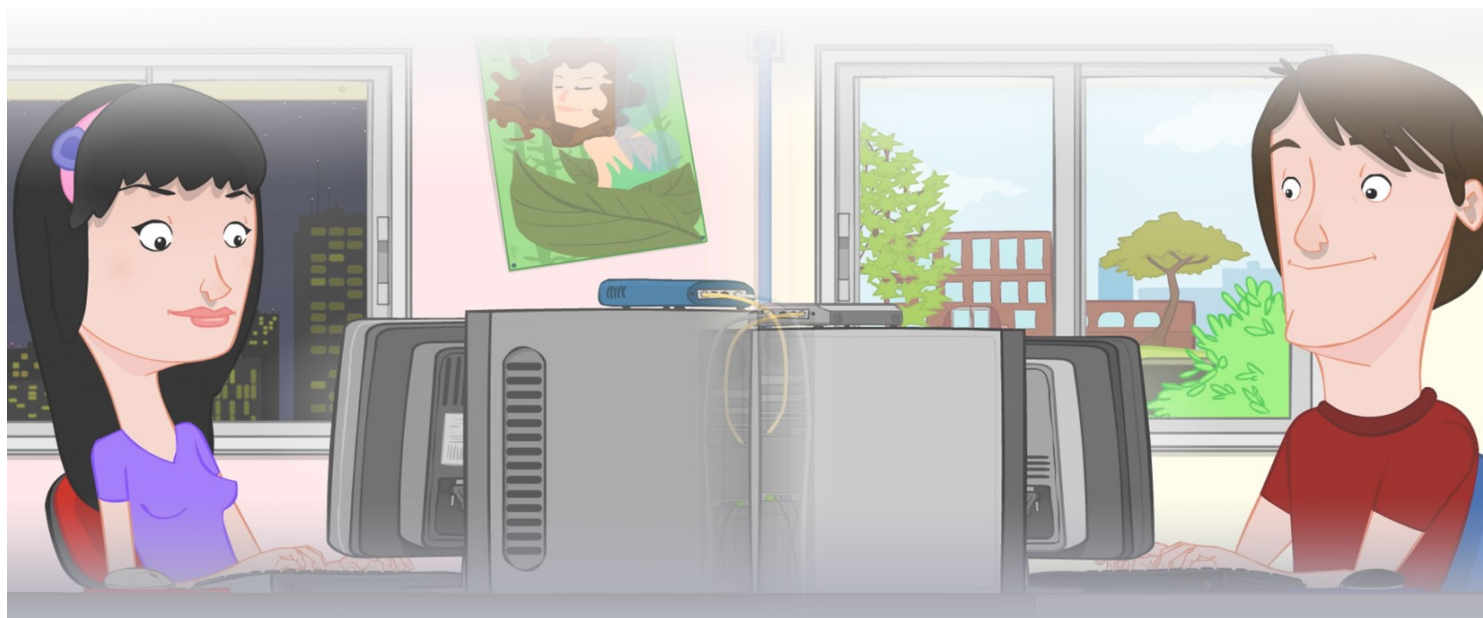
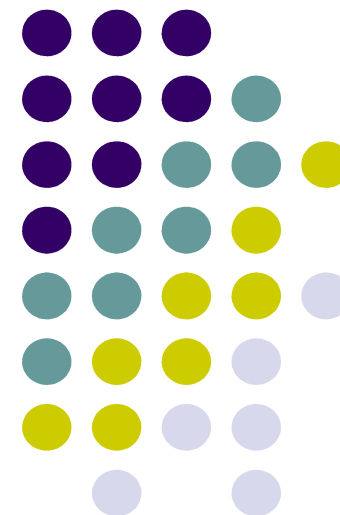
8.- DATUEN MAILA JAVAEEEn



Universidad
del País Vasco

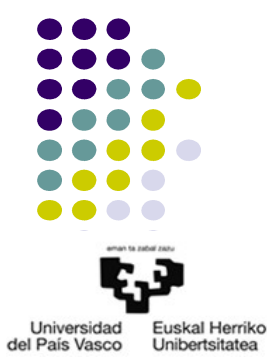
Euskal Herriko
Unibertsitatea

Maidier Huarte Arrayago, Gorka Prieto Agujeta, Jasone Astorga Burgo, Nerea Toledo Gandarias
Komunikazioen Ingeniaritza Saila
Bilboko Ingeniaritza Goi Eskola Teknikoa
UPV/EHU



TELEK:O
UPV/EHU Bilbao

ZERBITZU TELEMATIKO AURRERATUAK: 8.- DATUEN MAILA JAVAEEEn



Copyright © 2015 Mainer Huarte Arrayago, Gorka Prieto Agujeta, Jasone Astorga Burgo, Nerea Toledo Gandarias

This is a derivative work from:

Mainer Huarte Arrayago, Gorka Prieto Agujeta

“Servicios Telemáticos Avanzados: 8.- Capa de Datos en Java EE”

OCW UPV/EHU 2014 (ISSN 2255-2316), 2014

ZERBITZU TELEMATIKO AURRERATUAK: 8.- DATUEN MAILA JAVAEEEn lana, Mainer Huartek, Gorka Prietok, Jasone Astorga Burgok eta Nerea Toledo Gandariasek egina, Creative Commons-en Attribution-Share Alike 3.0 Unported License baimenaren menpe dago. Baimen horren kopia bat ikusteko, <http://creativecommons.org/licenses/by-sa/3.0/> webgunea bisitatu edo gutun bat bidali ondoko helbidera: Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

Lan hau beste honen eratorria da: Mainer Huarte Arrayago, Gorka Prieto Agujeta, “Servicios Telemáticos Avanzados: 8.- Capa de Datos en Java EE”, OCW UPV/EHU 2014 (ISSN 2255-2316), 2014

ZERBITZU TELEMATIKO AURRERATUAK: 8.- DATUEN MAILA JAVAEEEn by Mainer Huarte, Gorka Prieto, Jasone Astorga Burgo and Nerea Toledo Gandarias is licensed under a Creative Commons Attribution-Share Alike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or, send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

This is a derivative work from: Mainer Huarte Arrayago, Gorka Prieto Agujeta, “Servicios Telemáticos Avanzados: 8.- Capa de Datos en Java EE” OCW UPV/EHU 2014 (ISSN 2255-2316), 2014

DATUEN MAILA JAVAEEEn

INDIZEA

1.- Sarrera

2.- Multitier arkitekturak

3.- Java EE

3.1.- Sarrera

3.2.- Negozio Tier

3.2.1.- Osagaiak

3.2.2.- Negozio maila (BL): EJBak

3.2.3.- Datuak atzitzeko maila (DL): datuen iraunkortasuna

4.- ERREFERENTZIAK

1.- Sarrera

- Zer da “Web Aplikazio” bat?
 - Definizio orokorra
 - Java EEren definizioa
 - Aurkezpenerako web aplikazioak
 - Zerbitzurako web aplikazioak
- Exekuzio eremua
 - Web zerbitzaria+containers
 - Aplikazio zerbitzaria

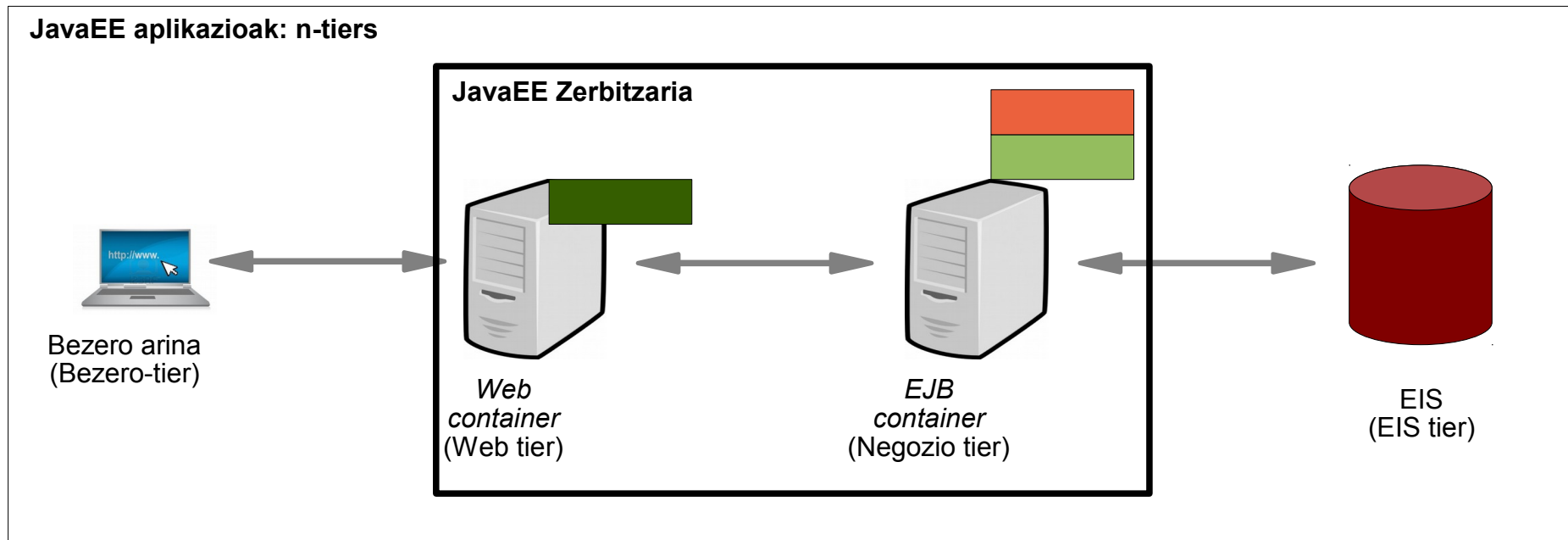
2.- *Multitier* arkitekturak

- Enpresetako aplikazioak, web aplikazioak
- 3 mailetako (*layers*) sw modeloa
 - *Datuen Atzipena*
 - *Negozioaren Logika*
 - *Aurkezpena*
- Sw banatua: *multitier* arkitektura
 - *Tiers vs layers*

3.- Java EE

3.1.- Sarrera

- Definizioa
- JavaEE aplikazioetarako eredua



3.- Java EE

3.2.- Negozio tier

- Negozio *tiereko* Java osagaiak
 - Negozio maila: EJBak
 - Java klase eta interfazeak
 - Berrerabilgarriak
 - EJB bezeroak
 - Managed Beans
 - EJB
 - Web aplikazioen bezeroak
 - Datuak atzitzeko maila: datuen iraunkortasunerako entitateak
 - Java klaseak
 - Datuak modu iraunkorrean atzitzea

3.- Java EE

3.2.- Negozio tier

- Datuak atzitzeko maila: datuen iraunkortasuna
 - Kontzeptuak
 - JPA
 - Entitateak
 - Entitateen datu-baseekiko erlazioa
 - Iraunkortasuna
 - Entitaterako klaseen beharrizanak
 - JavaBean
 - final ez
 - JPA oharrak

3.- Java EE

3.2.- Negozio tier

- Datuak atzitzeko maila: datuen iraunkortasuna
 - JPA oharrak
 - Derrigorrezkoak
 - @Entity
 - @Id
 - Atributuak
 - @NotNull
 - @Pattern
 - Kotsultak
 - @NamedQuery
 - @NamedQueries
 - Taulen arteko erlazioak
 - @OneToOne
 - @OneToMany
 - @ManyToOne
 - @ManyToMany

3.- Java EE

3.2.- Negozio tier

- Datuak atzitzeko maila: datuen iraunkortasuna
 - Iraunkortasunaren kudeaketa
 - Iraunkortasunaren Testuingurua (persistence context)
 - Iraunkortutako entitateak
 - Sinkronizazio gardena
 - Kudeatzailea:
 - EntityManager
 - Aplikazioak kudeatuta
 - Container-ak kudeatuta
 - Java Transaction API (JTA)
 - Erabilpena adierazteko

3.- Java EE

3.2.- Negozio tier

- Datuak atzitzeko maila: datuen iraunkortasuna
 - Iraunkortasunaren kudeaketa
 - Kudeatzailearen eragiketak (metodoak)
 - find: SELECT FROM x WHERE x.pk
 - createNamedQuery: kontsulta estatikoak
 - createQuery: kontsulta dinamikoak
 - persist: INSERT/UPDATE
 - remove: DELETE
 - detach
 - flush
 - clear

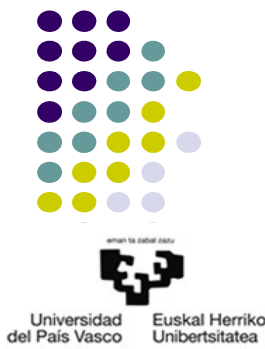
3.- Java EE

3.2.- Negozio tier

- Datuak atzitzeko maila: datuen iraunkortasuna
 - Datu-baseekiko kontsultak
 - Aukerak
 - Java Persistence Query Language (JPQL)
 - SQL lengoaiaren antzekoa
 - Eraitzen casting beharrezkoa
 - Irakurri/idazteko erraza
 - Criteria API
 - Eraginkorragoa
 - Ez da beharrezkoa beste lengoaia bat ikastea: JPQL
 - Ez da beharrezkoa casting egitea
 - Kodea luzeagoa: bitarteko objektu eta eragiketak

3.- Java EE

3.2.- Negozio tier



- Datuak atzitzeko maila: datuen iraunkortasuna
 - Datu-baseekiko kontsultak
 - Kontsulta dinamikoak

```
...
@PersistenceContext
public EntityManager em;
...

public List findWithName(String name) {
    return em.createQuery("SELECT c FROM Customer c WHERE c.name LIKE :custName")
        .setParameter("custName", name)
        .setMaxResults(10)
        .getResultList();
}
```

- Kontsulta estatikoak

```
@NamedQuery(
    name="findAllCustomersWithName",
    query="SELECT c FROM Customer c WHERE c.name LIKE :custName"
)
@Entity
public class Entitatea {
    ...
}
```

```
public class Bean {

    @PersistenceContext
    public EntityManager em;

    ...
    customers = em
        .createNamedQuery("findAllCustomersWithName")
        .setParameter("custName", "Smith")
        .getResultList();
}
```

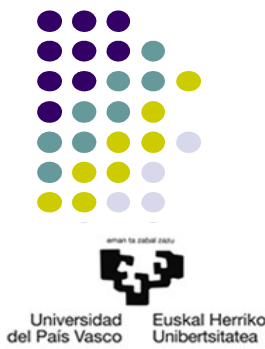
3.- Java EE

3.2.- Negozio tier

- Datuak atzitzeko maila: datuen iraunkortasuna
 - DBrako konexioaren konfigurazioa
 - JBoss: **standalone.xml**
 - JavaEE: **persistence.xml**

3.- Java EE

3.2.- Negozio tier



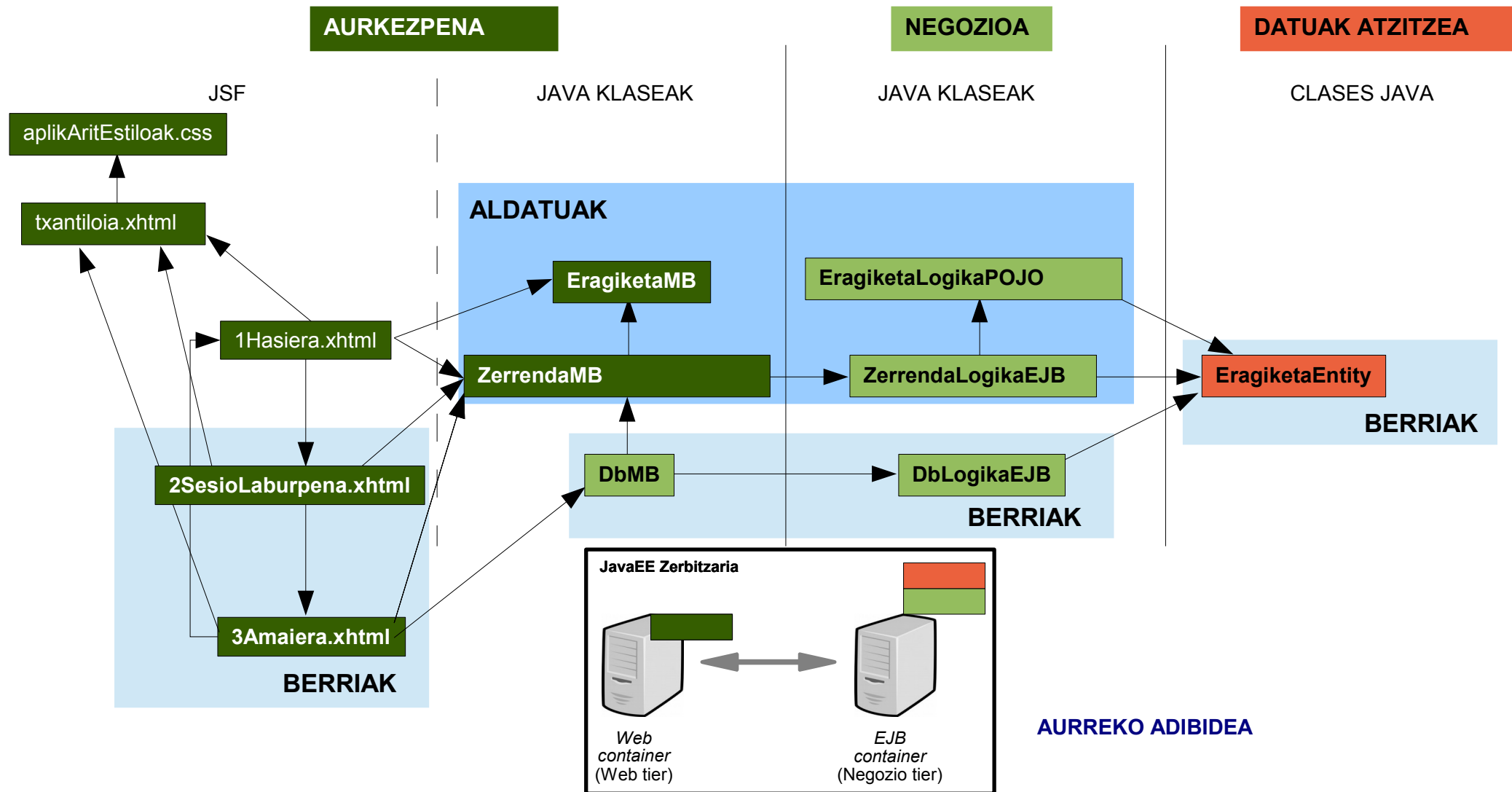
- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="T6-6AD_JSF-EJB-JPA_1" transaction-type="JTA">
    <jta-data-source>java:jboss/datasources/ZTA_T6_AD6</jta-data-source>
    <class>dl.EragiketaEntity</class>
    <properties>
      <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
      <property name="hibernate.hbm2ddl.auto" value="update" />
    </properties>
  </persistence-unit>
</persistence>
```

3.- Java EE

3.2.- Negozio tier

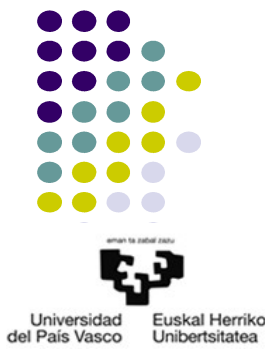
- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin



AURREKO ADIBIDEA

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

```
package dl;

import java.io.Serializable;
import javax.persistence.*;

@Entity
@Table(name="Eragiketa")
@NamedQueries
({
    @NamedQuery(name="EragiketaEntity.findAll", query="SELECT e FROM EragiketaEntity e"),
    @NamedQuery(name="EragiketaEntity.findAllSesioa", query="SELECT e FROM EragiketaEntity e WHERE e.sesioa = :sesioa ORDER BY e.ordena")
})
public class EragiketaEntity implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int idEragiketa;
    @Column(name="Em")
    private float em;
    @Column(name="Er")
    private char er;
    //TAULAKO BESTE ZUTABEEI DAGOZKIEN LERROAK

    public EragiketaEntity() {
    }

    public int getIdEragiketa() {
        return this.idEragiketa;
    }

    public void setIdEragiketa(int idEragiketa) {
        this.idEragiketa = idEragiketa;
    }

    //BESTE ATRIBUTU GUZTIEI DAGOZKIEN get eta set METODOAK
}
```

Entity: dl.EragiketaEntity.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

POJO: bl.EragiketaLogikaPOJO.java

```
package bl;

import dl.EragiketaEntity;

public class EragiketaLogikaPOJO {

    public static void kalkulatuEragiketa(EragiketaEntity e)
    {
//AURREKO BERTSIOKO KODE BERDINA
    }
}
```

EJB: bl.ZerrendaLogikaEJB.java

```
package bl;

//BEHARREZKO import LERROAK

import dl.EragiketaEntity;

@Stateful
@LocalBean
public class ZerrendaLogikaEJB {

    private List<EragiketaEntity> zerrenda;

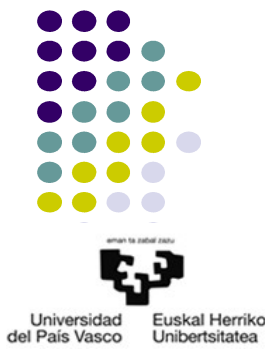
    public ZerrendaLogikaEJB() {
        zerrenda=new ArrayList<EragiketaEntity>();
    }

    public List<EragiketaEntity> getzerrenda() {
        return zerrenda;
    }

    public void addEragiketa(EragiketaEntity e) {
//AURREKO BERTSIOKO KODE BERDINA
    }
//clearZerrenda(): AURREKO BERTSIOKO KODE BERDINA
}
```

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

```
package bl;

//BEHARREZKO import LERROAK

@Stateless
@LocalBean
public class DbLogikaEJB {

    @PersistenceContext
    private EntityManager em;

    public void addZerrendaDB(List<EragiketaEntity> zerrenda)
    {
        EragiketaEntity ee;

        for(int i=0;i<zerrenda.size();i++)
        {
            ee = zerrenda.get(i);
            ee.setOrdena(i+1);

            em.persist(ee);
        }

        return;
    }

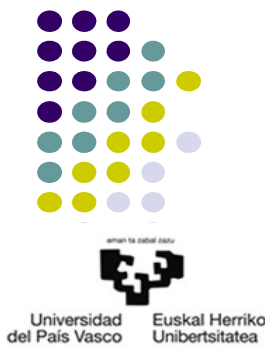
    public List<EragiketaEntity> getZerrendaDB(String sesioIzena) {
        @SuppressWarnings("unchecked")
        List<EragiketaEntity> zerrendaDB = (List<EragiketaEntity>)em.createNamedQuery("EragiketaEntity.findAllSesioa")
            .setParameter("sesioa",sesioIzena).getResultList();

        return (List<EragiketaEntity>)zerrendaDB;
    }
}
```

EJB: bl.DbLogikaEJB.java

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

```
package pl;

//BEHARREZKO import LERROAK

import dl.EragiketaEntity;

@ManagedBean
@RequestScoped
public class EragiketaMB extends EragiketaEntity {
    private static final long serialVersionUID = 1L;
    //BESTE KODE GUZTIA AURREKO BERTSIOKO BERDINA
}
```

ManagedBean: pl.EragiketaMB.java

```
package pl;

//BEHARREZKO import LERROAK

import dl.EragiketaEntity;

@ManagedBean
@SessionScoped
public class ZerrendaMB {

    @EJB
    private ZerrendaLogikaEJB zerrenda;

    public ZerrendaMB() {
        zerrenda=new ZerrendaLogikaEJB();
    }

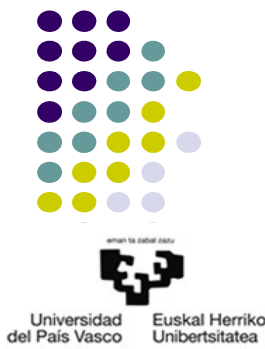
    public List<EragiketaEntity> getzerrenda() {
        return zerrenda.getzerrenda();
    }

    //HEMENGO KODE GUZTIA HURRENGO ORRIALDEAN DAGO
}
```

ManagedBean: pl.ZerrendaMB.java

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

```
package pl;

//BEHARREZKO import LERROAK

import dl.EragiketaEntity;

@ManagedBean
@SessionScoped
public class ZerrendaMB {

//HEMENGO KODE GUZTIA AURREKO ORRIALDEAN DAGO

    public void addEragiketa(EragiketaMB e) {
        EragiketaEntity ee=new EragiketaEntity();

        ee.setSesioa(getSesioIzena());
        ee.setEr1(e.getEr1());
        ee.setEr(e.getEr());
        ee.setEr2(e.getEr2());

        zerrenda.addEragiketa(ee);
        e.setEm(ee.getEm());
        e.setamaitua();

        return;
    }

    public String getSesioIzena() {
        String user=((HttpServletRequest)FacesContext.getCurrentInstance()
            .getExternalContext().getRequest()).getRemoteAddr();
        HttpSession sesioa=(HttpSession)FacesContext.getCurrentInstance().getExternalContext().getSession(true);
        String sesioIzena="IP-BEZEROA:"+user+" TIME:"+new Date(sesioa.getCreationTime());

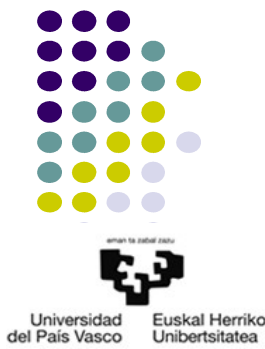
        return sesioIzena;
    }

//endSesioa(): AURREKO BERTSIOKO KODE BERDINA
}
```

ManagedBean: pl.ZerrendaMB.java

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

```
package pl;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.ejb.EJB;

import bl.DbLogikaEJB;

import java.util.List;

import dl.EragiketaEntity;

@ManagedBean
@RequestScoped
public class DbMB {

    @EJB
    private DbLogikaEJB db;

    public DbMB()
    {
        db=new DbLogikaEJB();
    }

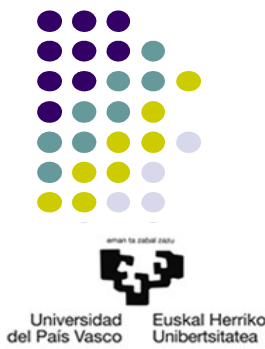
    public List<EragiketaEntity> getZerrendaDB(String sesioIzena) {
        return db.getZerrendaDB(sesioIzena);
    }

    public void addZerrendaDB(List<EragiketaEntity> zerrenda) {
        db.addZerrendaDB(zerrenda);
        return;
    }
}
```

ManagedBean: pl.DbMB.java

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

```
<!DOCTYPE html ...>
<!-- HEMENGO KODEA AURREKO BERTSIOKO 1Hasiera.xhtml ORRIALDEKO BERDINA DA
      <h:commandButton action="2SesioLaburpena" id="submit" value="ERAGIKETAK IKUSI" />
    </h:form>
  </p>
</ui:define>
</ui:composition>
</html>
```

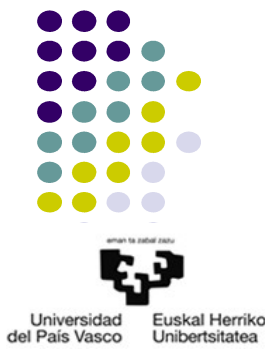
JSF: 1Hasiera.xhtml

```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <ui:composition template="/txantiloia.xhtml">
    <ui:define name="DokumentuIzenburua">2SesioLaburpena: Aplikazio aritmetikoaren sesio-laburpenaren orrialdea</ui:define>
    <ui:define name="OrrialdeIzenburua">APLIKAZIO ARITMETIKOA JSF, EJB eta JPArekin: SESIOAREN LABURPENEA</ui:define>
    <ui:define name="Edukia">
      <h1>SESIOAREN LABURPENEA</h1>
      Eragiketa kopurua: #{zerrendaMB.zerrenda.size()}
      <h:dataTable value="#{zerrendaMB.zerrenda}" var="eragiketa" border="2">
        <h:column>
          <f:facet name="header">ERAGIKETAK</f:facet>
          #{eragiketa.er1}#{eragiketa.er}#{eragiketa.er2}
        </h:column>
        <h:column>
          <f:facet name="header">EMAITZAK</f:facet>
          #{eragiketa.em}
        </h:column>
      </h:dataTable>
      <p>
        <h:form>
          <h:commandButton action="3Amaiera" actionListener="#{dbMB.addZerrendaDB(zerrendaMB.zerrenda)}" id="submit" value="SESIOA DBan GRABATU" />
        </h:form>
      </p>
    </ui:define>
  </ui:composition>
</html>
```

JSF: 2SesioLaburpena.xhtml

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 1. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta JPEntities-ekin

JSF: 3Amaiera.xhtml

```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/txantiloia.xhtml">
<ui:define name="DokumentuIzenburua">3Amaiera: Aplikazio aritmetikoaren amaierako orrialdea</ui:define>
<ui:define name="OrrialdeIzenburua">APLIKAZIO ARITMETIKOA JSF, EJB eta JPAREKIN: AMAIERA</ui:define>
<ui:define name="Edukia">
  <h1>DBan GRABATUTAKO SESIOAREN LABURPENA</h1>
  Sesioa: #{zerrendaMB.sesioIzena}
  <h:dataTable value="#{dbMB.getZerrendaDB(zerrendaMB.sesioIzena)}" var="eragiketa" border="2">
    <h:column>
      <f:facet name="header">ERAGIKETAK</f:facet>
      #{eragiketa.er1}#{eragiketa.er}#{eragiketa.er2}
    </h:column>
    <h:column>
      <f:facet name="header">EMAITZAK</f:facet>
      #{eragiketa.em}
    </h:column>
  </h:dataTable>
  <p>
    <h:form>
      <h:commandButton action="1Hasi" actionListener="#{zerrendaMB.endSesio()}" id="submit" value="SESIO BERRIA HASI" />
    </h:form>
  </p>
</ui:define>
</ui:composition>
</html>
```


3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP Aplikazioak
 - Programatu beharreko blokeak
 - Datu-basea
 - JavaEE aplikazioa JSFekin
 - Maila bakoitzeko Java klaseak:
 - DL
 - BL
 - PL
 - .xhtml fitxategiak
 - Konfigurazio-fitxategiak
 - persistence.xml
 - web.xml
 - ...

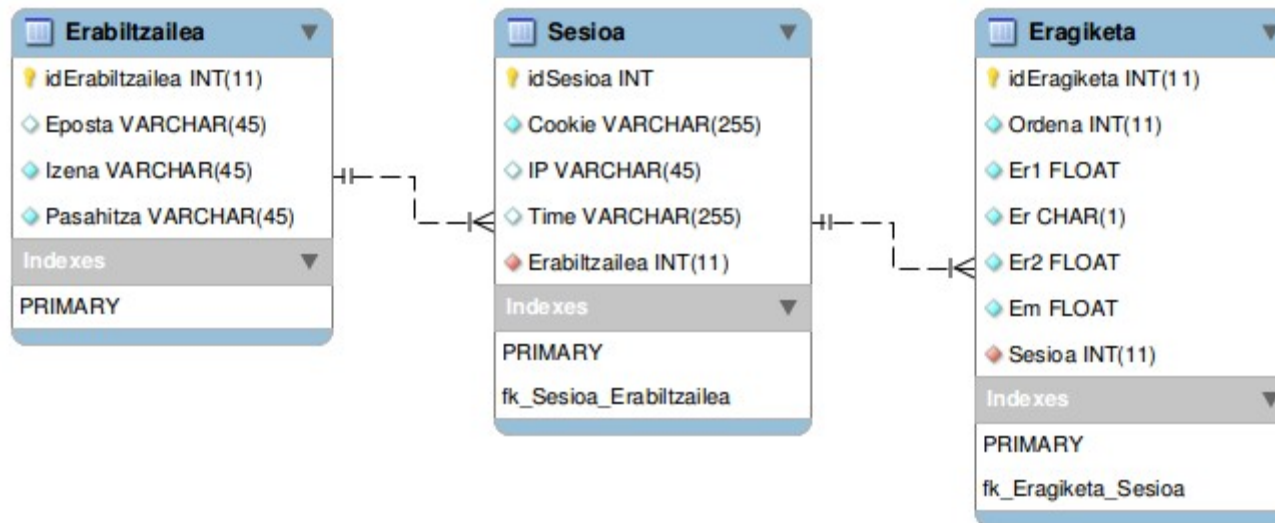
3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin

– Planteamendu posiblea

- Datu-basea: Datuen eta erlazioen eredua



3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - Datu-basea: taulen sorrera

```
CREATE TABLE IF NOT EXISTS `Erabiltzailea` (  
  `idErabiltzailea` INT(11) NOT NULL AUTO_INCREMENT ,  
  `Eposta` VARCHAR(45) NULL ,  
  `Izena` VARCHAR(45) NOT NULL ,  
  `Pasahitza` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`idErabiltzailea`) )  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

Erabiltzailea taula sortzeko sententziak

```
CREATE TABLE IF NOT EXISTS `Sesioa` (  
  `idSesioa` INT NOT NULL AUTO_INCREMENT ,  
  `Cookie` VARCHAR(255) NOT NULL ,  
  `IP` VARCHAR(45) NULL ,  
  `Time` VARCHAR(255) NULL ,  
  `Erabiltzailea` INT(11) NOT NULL ,  
  PRIMARY KEY (`idSesioa`) ,  
  INDEX `fk_Sesioa_Erabiltzailea` (`Erabiltzailea` ASC) ,  
  CONSTRAINT `fk_Sesioa_Erabiltzailea1`  
  FOREIGN KEY (`Erabiltzailea` )  
  REFERENCES `Erabiltzailea` (`idErabiltzailea` )  
  ON DELETE CASCADE  
  ON UPDATE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

Sesioa taula sortzeko sententziak

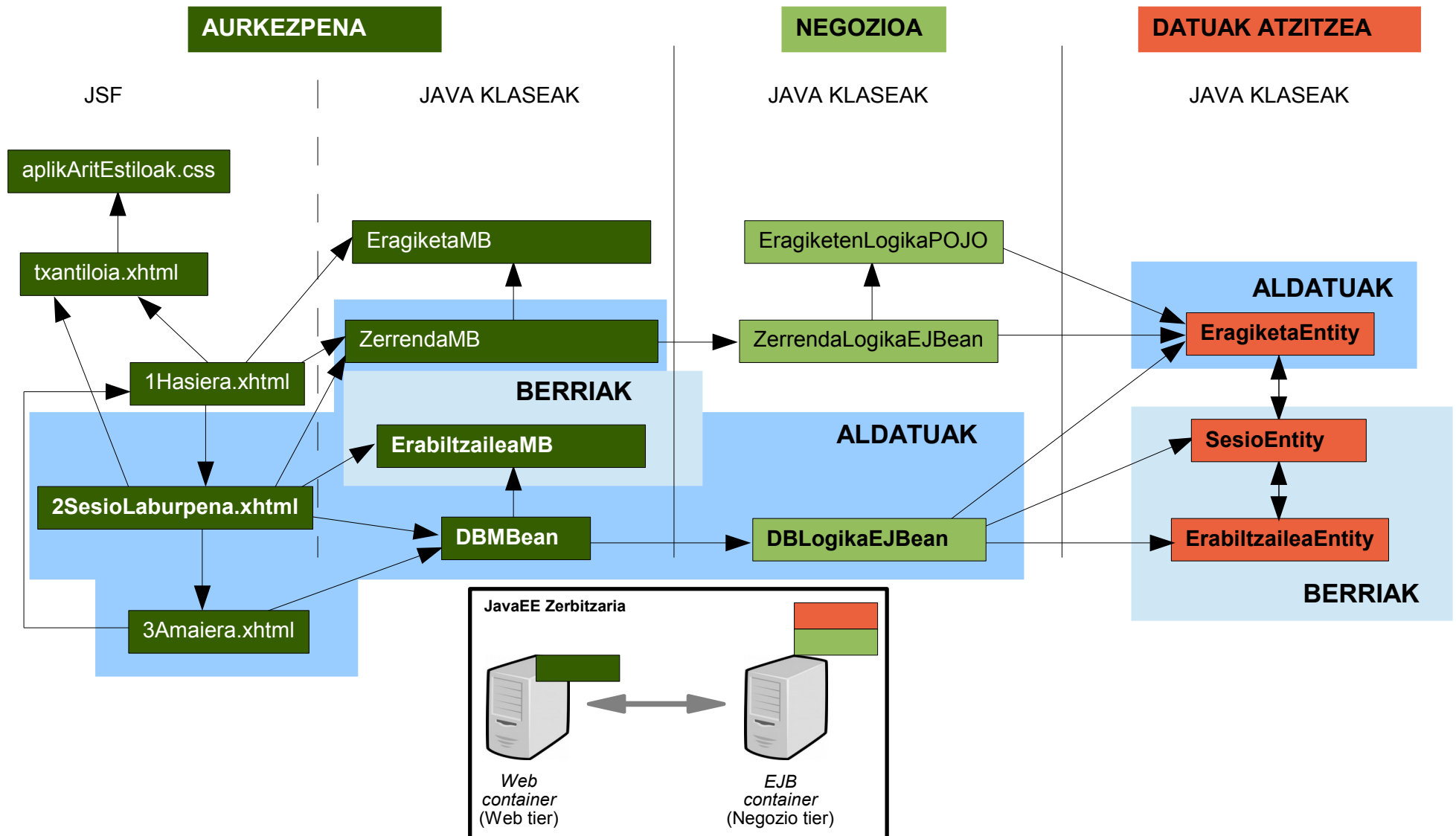
```
CREATE TABLE IF NOT EXISTS `Eragiketa` (  
  `idEragiketa` INT(11) NOT NULL AUTO_INCREMENT ,  
  `Ordena` INT(11) NOT NULL ,  
  `Er1` FLOAT NOT NULL ,  
  `Er` CHAR(1) NOT NULL ,  
  `Er2` FLOAT NOT NULL ,  
  `Em` FLOAT NOT NULL ,  
  `Sesioa` INT(11) NOT NULL ,  
  PRIMARY KEY (`idEragiketa`) ,  
  INDEX `fk_Eragiketa_Sesioa` (`Sesioa` ASC) ,  
  CONSTRAINT `fk_Eragiketa_Sesioa`  
  FOREIGN KEY (`Sesioa` )  
  REFERENCES `Sesioa` (`idSesioa` )  
  ON DELETE CASCADE  
  ON UPDATE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

Eragiketa taula sortzeko sententziak

3.- Java EE

3.2.- Negozio tier

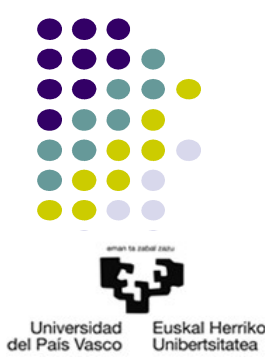
- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin



3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
package dl;

//AURREKO BERTSIOKO KODE BERDINA
public class EragiketaEntity implements Serializable {

//AURREKO BERTSIOKO ATRIBUTU BERDINAK

    //bi-directional many-to-one association to SesioaEntity
    @ManyToOne
    @JoinColumn(name="Sesioa")
    private SesioaEntity sesioaBean;

    public EragiketaEntity() {
    }

    public EragiketaEntity(float em, char er, float er1, float er2, int ordena, SesioaEntity sesioaBean) {
        this.em=em;
        this.er=er;
        this.er1=er1;
        this.er2=er2;
        this.ordena=ordena;
        this.sesioaBean=sesioaBean;
    }

//AURREKO BERTSIOKO get eta set METODOAK

    public SesioaEntity getSesioaBean() {
        return this.sesioaBean;
    }

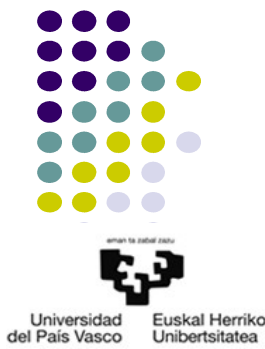
    public void setSesioaBean(SesioaEntity sesioaBean) {
        this.sesioaBean = sesioaBean;
    }
}
```

Entity: dl.EragiketaEntity.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
package dl;

//BEHARREZKO import LERROAK

@Entity
@Table(name="Sesioa")
@NamedQuery(name="SesioaEntity.findAll", query="SELECT s FROM SesioaEntity s")
public class SesioaEntity implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int idSesioa;

    @Column(name="Cookie")
    private String cookie;

    @Column(name="IP")
    private String ip;

    @Column(name="Time")
    private String time;

    //bi-directional many-to-one association to EragiketaEntity
    @OneToMany(mappedBy="sesioaBean",cascade=CascadeType.ALL) //cascade idatzi behar da
    private List<EragiketaEntity> eragiketas;

    //bi-directional many-to-one association to ErabiltzaileaEntity
    @ManyToOne
    @JoinColumn(name="Erabiltzailea")
    private ErabiltzaileaEntity erabiltzaileaBean;

    //HEMENGO KODE GUZTIA HURRENGO ORRIALDEAN DAGO
}

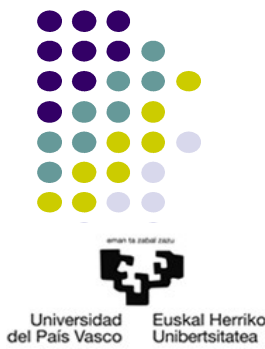
```

Entity: dl.SesioaEntity.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
//HEMENGO KODE GUZTIA AURREKO ORRIALDEAN DAGO
```

Entity: dl.SesioaEntity.java

```
public SesioaEntity() {
}

public SesioaEntity(String cookie, String ip, String time, ErabiltzaileaEntity erabiltzaileaBean) {
    this.cookie=cookie;
    this.ip=ip;
    this.time=time;
    this.erabiltzaileaBean=erabiltzaileaBean;
}

//FK EZ DIREN ATRIBUTUEN get eta set METODOAK

public List<EragiketaEntity> getEragiketas() {
    return this.eragiketas;
}

public void setEragiketas(List<EragiketaEntity> eragiketas) {
    this.eragiketas = eragiketas;
}

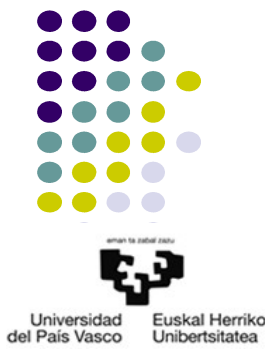
public ErabiltzaileaEntity getErabiltzaileaBean() {
    return this.erabiltzaileaBean;
}

public void setErabiltzaileaBean(ErabiltzaileaEntity erabiltzaileaBean) {
    this.erabiltzaileaBean = erabiltzaileaBean;
}
}
```

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
package dl;

//BEHARREZKO import LERROAK

@Entity
@Table(name="Erabiltzailea")
@NamedQueries({
    @NamedQuery(name="ErabiltzaileaEntity.findAll", query="SELECT e FROM ErabiltzaileaEntity e"),
    @NamedQuery(name="ErabiltzaileaEntity.findErabiltzailea", query="SELECT e FROM ErabiltzaileaEntity e WHERE e.izena = :izena")
})
public class ErabiltzaileaEntity implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int idErabiltzailea;

    @Column(name="Eposta")
    private String eposta;

    @Column(name="Izena")
    private String izena;

    @Column(name="Pasahitza")
    private String pasahitza;

    //bi-directional many-to-one association to SesioaEntity
    @OneToMany(mappedBy="erabiltzaileaBean", cascade=CascadeType.ALL)
    private List<SesioaEntity> sesioas;

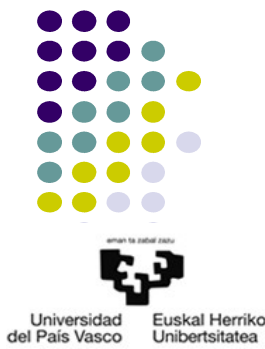
    //HEMENGO KODE GUZTIA HURRENGO ORRIALDEAN DAGO
}
```

Entity: dl.Erabiltzailea.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
//HEMENGO KODE GUZTIA AURREKO ORRIALDEAN DAGO
```

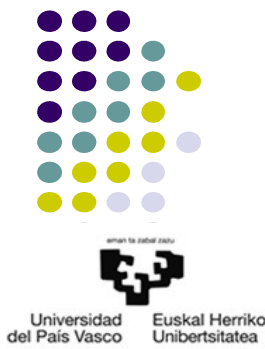
Entity: dl.Erabiltzailea.java

```
public ErabiltzaileaEntity() {  
}  
  
public ErabiltzaileaEntity(String eposta, String izena, String pasahitza) {  
    this.eposta=eposta;  
    this.izena=izena;  
    this.pasahitza=pasahitza;  
}  
  
//FK EZ DIREN ATRIBUTUEN get eta set METODOAK  
  
public List<SesioaEntity> getSesioas() {  
    return this.sesioas;  
}  
  
public void setSesioas(List<SesioaEntity> sesioas) {  
    this.sesioas = sesioas;  
}  
}
```

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
package bl;
//BEHARREZKO import LERROAK
@Stateful
@LocalBean
public class DbLogikaEJB {
    @PersistenceContext
    private EntityManager em;
    private ErabiltzaileaEntity erabiltzaileaDB;
    private SesioaEntity sesioaDB;

    public void addZerrendaDB(List<EragiketaEntity> zerrenda, ErabiltzaileaEntity erabiltzailea, SesioaEntity sesioa) {
        addErabiltzaileaDB(erabiltzailea);

        if(erabiltzaileaDB!=null) //erabiltzailea ezin izan daba DBan sartu, ez da beste ezer egin behar
        {
            addSesioaDB(sesioa);

            EragiketaEntity ee;
            for(int i=0;i<zerrenda.size();i++) {
                ee=zerrenda.get(i);
                ee.setOrdena(i+1);
                ee.setSesioaBean(sesioaDB);

                em.persist(ee);
            }
        }
        return;
    }
}
//HEMENGO KODE GUZTIA HURRENGO ORRIALDEAN DAGO
}
```

EJB: bl.DbLogikaEJB.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa

```
//HEMENGO KODE GUZTIA AURREKO ORRIALDEAN DAGO EJB: bl.DbLogikaEJB.java

private void addErabiltzaileaDB(ErabiltzaileaEntity e) {
    try {

erabiltzaileaDB=(ErabiltzaileaEntity)em.createNamedQuery("ErabiltzaileaEntity.findErabiltzailea").setParameter("izena",e.ge
tIzena()).getSingleResult();

        if(!((erabiltzaileaDB.getPasahitza()).equals(e.getPasahitza()))
            erabiltzaileaDB=null;
        }
        catch(Exception ex)
        {
            erabiltzaileaDB=new ErabiltzaileaEntity(e.getEposta(),e.getIzena(),e.getPasahitza());
            em.persist(erabiltzaileaDB); //erabiltzaile berria bada, DBan sartzen da
        }

        return;
    }

private void addSesioaDB(SesioaEntity sesioa)
{
    sesioaDB=sesioa;
    sesioaDB.setErabiltzaileaBean(erabiltzaileaDB);

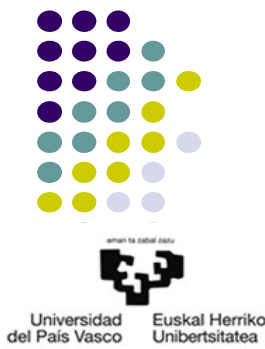
    em.persist(sesioaDB);
    return;
}

//HEMENGO KODE GUZTIA HURRENGO ORRIALDEAN DAGO
}
```

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
//HEMENGO KODE GUZTIA AURREKO ORRIALDEAN DAGO
```

EJB: bl.DbLogikaEJB.java

```
public ErabiltzaileaEntity getErabiltzaileaDB() {
    try
    {
        erabiltzaileaDB=em.find(ErabiltzaileaEntity.class, erabiltzaileaDB.getIdErabiltzailea());
    }
    catch(Exception ex)
    {
        }//erabiltzaileaDB null bada, salbuespena gertatzen da
    }
    return erabiltzaileaDB;
}

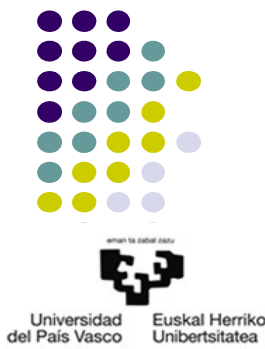
public SesioaEntity getSesioaDB() {
    try
    {
        sesioaDB=(SesioaEntity)em.find(SesioaEntity.class, sesioaDB.getIdSesioa());
    }
    catch(Exception ex)
    {
        }//sesioaDB null bada, salbuespena gertatzen da
    }
    return sesioaDB;
}

public List<EragiketaEntity> getZerrendaDB() {
    List<EragiketaEntity> zerrendaDB =
(List<EragiketaEntity>)em.createNamedQuery("EragiketaEntity.findAllSesioa").setParameter("sesioa",sesioaDB).getResultList();

    return (List<EragiketaEntity>)zerrendaDB;
}
}
```

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa

```
package pl;

//BEHARREZKO import LERROAK

@ManagedBean
@SessionScoped
public class ZerrendaMB {

    @EJB
    private ZerrendaLogikaEJB zerrenda;

    // ZerrendaMB(): AURREKO BERTSIOKO BERDINA
    // List<EragiketaEntity> getzerrenda(): AURREKO BERTSIOKO BERDINA

    public void addEragiketa(EragiketaMB e) {
        EragiketaEntity ee = new EragiketaEntity(0, e.getEr(), e.getEr1(), e.getEr2(), 0, null);
        zerrenda.addEragiketa(ee);
        e.setEm(ee.getEm());
        e.setamaitua();

        return;
    }

    // String getSesioIzena(): AURREKO BERTSIOKO BERDINA
    // void endSesioa(): AURREKO BERTSIOKO BERDINA
}
```

MB: pl.ZerrendaMB.java

```
package pl;

//BEHARREZKO import LERROAK

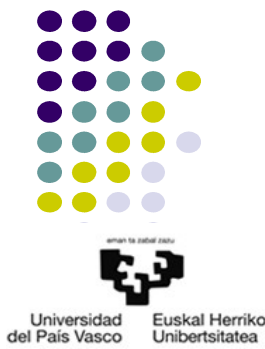
@ManagedBean
@RequestScoped
public class ErabiltzaileaMB extends ErabiltzaileaEntity {
    private static final long serialVersionUID = 1L;
}
```

MB: pl.ErabiltzaileaMB.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
package pl;

//BEHARREZKO import LERROAK

@ManagedBean
@RequestScoped
public class DbMB {
    @EJB
    private DbLogikaEJB db;

    public DbMB() {
        db=new DbLogikaEJB();
    }

    public void addZerrendaDB(List<EragiketaEntity> zerrenda, ErabiltzaileaMB eMB, SesioaEntity se) {
        ErabiltzaileaEntity ee=new ErabiltzaileaEntity(eMB.getEposta(),eMB.getIzena(),eMB.getPasahitza());

        db.addZerrendaDB(zerrenda, ee, se);
        return;
    }

    public List<EragiketaEntity> getZerrendaDB() {
        return db.getZerrendaDB();
    }

    public ErabiltzaileaEntity getErabiltzaileaDB() {
        return db.getErabiltzaileaDB();
    }

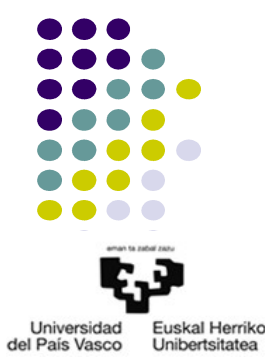
    public SesioaEntity getSesioaDB() {
        return db.getSesioaDB();
    }
}
```

MB: pl.DbMB.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa

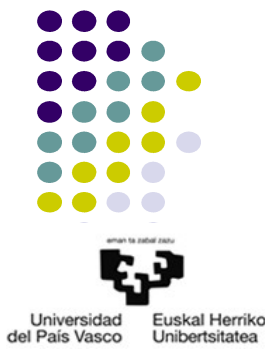


JSF: 2SesioLaburpena.xhtml

```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml" ...>
<ui:composition template="/txantiloia.xhtml">
<ui:define name="DokumentuIzenburua">2SesioLaburpena: Aplikazio aritmetikoaren sesio-laburpenaren orrialdea</ui:define>
<ui:define name="OrrialdeIzenburua">APLIKAZIO ARITMETIKOA JSF, EJB eta JPArekin: SESIOAREN LABURPENEA</ui:define>
<ui:define name="Edukia">
  <h1>SESIOAREN LABURPENEA</h1>
  <h:dataTable value="#{zerrendaMB.zerrenda}" var="eragiketa" border="2">
    <h:column>
      <f:facet name="header">ERAGIKETAK</f:facet>
      #{eragiketa.er1}#{eragiketa.er}#{eragiketa.er2}
    </h:column>
    <h:column>
      <f:facet name="header">EMAITZAK</f:facet>
      #{eragiketa.em}
    </h:column>
  </h:dataTable>
  <p>
  <h:form>
    <h1>SAR ITZAZU ERABILTZAILE-DATUAK</h1>
    <label>Izena: </label>
    <h:inputText id="izena" value="#{erabiltzaileaMB.izena}" required="true" requiredMessage="ERROREA: izena sartzea derrigorrezkoa da" />
    <h:message for="izena" style="color:blue" /><br />
    <label>Pasahitza: </label>
    <h:inputText id="pasahitza" value="#{erabiltzaileaMB.pasahitza}" required="true" requiredMessage="ERROREA: pasahitza sartzea derrigorrezkoa da" />
    <h:message for="pasahitza" style="color:blue" /><br />
    <label>E-posta: </label>
    <h:inputText id="eposta" value="#{erabiltzaileaMB.eposta}" /><br />
    <h:commandButton action="3Amaiera" actionListener="#{dbMB.addZerrendaDB(zerrendaMB.zerrenda,erabiltzaileaMB,zerrendaMB.sesioa)}" id="submitAdd" value="SESIOA DBan GRABATU" />
  </h:form>
  </p>
</ui:define>
</ui:composition>
</html>
```

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 2. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin
 - Planteamendu posiblea
 - JavaEE aplikazioa

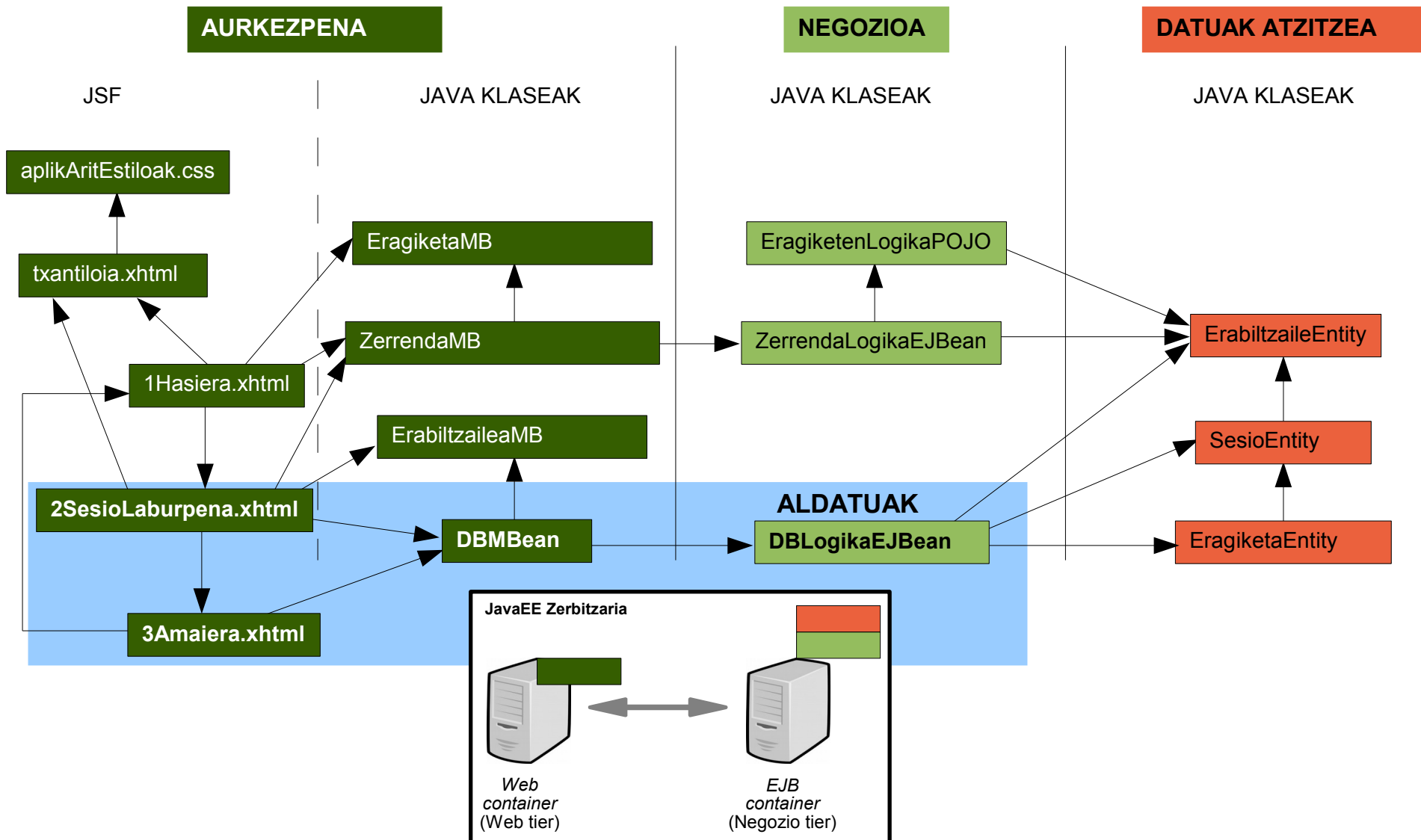
JSF: 3Amaiera.xhtml

```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:composition template="/txantiloia.xhtml">
<ui:define name="DokumentuIzenburua">3Amaiera: Aplikazio aritmetikoaren amaierako orrialdea</ui:define>
<ui:define name="OrrialdeIzenburua">APLIKAZIO ARITMETIKOA JSF, EJB eta JPAREKIN: AMAIERA</ui:define>
<ui:define name="Edukia">
  <h1>DBan GRABATUTAKO SESIOAREN LABURPENA</h1>
  Erabiltzailea: #{dbMB.erabiltzaileaDB.izena}<br />
  Sesioa: #{dbMB.sesioaDB.cookie}<br />
  <h:dataTable value="#{dbMB.zerrendaDB}" var="eragiketa" border="2">
    <h:column>
      <f:facet name="header">ERAGIKETAK</f:facet>
      #{eragiketa.er1}#{eragiketa.er}#{eragiketa.er2}
    </h:column>
    <h:column>
      <f:facet name="header">EMAITZAK</f:facet>
      #{eragiketa.em}
    </h:column>
  </h:dataTable>
  <p>
    <h:form>
      <h:commandButton action="1Hasiera" actionListener="#{zerrendaMB.endSesio()}" id="submit" value="SESIO BERRIA HASI" />
    </h:form>
  </p>
</ui:define>
</ui:composition>
</html>
```


3.- Java EE

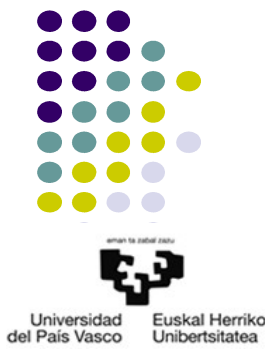
3.2.- Negozio tier

- JSF+EJB+JP 3. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu



3.- Java EE

3.2.- Negozio tier

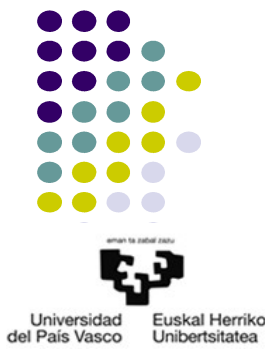


- JSF+EJB+JP 3. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu
 - Planteamendu posiblea
 - JavaEE aplikazioa

```
//AURREKO BERTSIOKO KODE BERDINA+removeErabiltzaileaDB(ErabiltzaileaEntity e) METODOA EJB: bl.DbLogikaEJB.java  
  
public void removeErabiltzaileaDB(ErabiltzaileaEntity e) {  
    ErabiltzaileaEntity erabiltzaileaBean=null;  
    try  
    {  
  
erabiltzaileaBean=(ErabiltzaileaEntity)em.createNamedQuery("ErabiltzaileaEntity.findErabiltzailea").setParameter("izena",e.getIzena()).getSingleResult(); //Emaitarik ez badago, salbuespena botatzen du getSingleResult-ek  
        if((erabiltzaileaBean.getPasahitza()).equals(e.getPasahitza()))  
            em.remove(erabiltzaileaBean);  
    }  
    catch(Exception ex)  
    {  
    }  
  
    return;  
    }  
}
```

3.- Java EE

3.2.- Negozio tier



- JSF+EJB+JP 3. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu
 - Planteamendu posiblea
 - JavaEE aplikazioa

```
package pl;
//BEHARREZKO import LERROAK

@ManagedBean
@RequestScoped
public class DbMB {
    @EJB
    private DbLogikaEJB db;
    private boolean addDB;
    private boolean removeDB;

    //ERAIKITZAILEA AURREKO BERTSIOKO BERDINA

    public void addZerrendaDB(List<EragiketaEntity> zerrenda, ErabiltzaileaEntity erabiltzailea, SesioaEntity se) {
        ErabiltzaileaEntity ee=new ErabiltzaileaEntity(eMB.getEposta(),eMB.getIzena(),eMB.getPasahitza());
        db.addZerrendaDB(zerrenda, ee, se);
        addDB=true;
        return;
    }

    public boolean getAddDB() {
        return addDB;
    }

    //getZerrendaDB(), getErabiltzaileaDB(), getSesioaDB() AURREKO BERTSIOAN BEZALA

    public void removeErabiltzaileaDB(ErabiltzaileaMB eMB) {
        ErabiltzaileaEntity ee=new ErabiltzaileaEntity(eMB.getEposta(),eMB.getIzena(),eMB.getPasahitza());
        db.removeErabiltzaileaDB(ee);
        removeDB=true;
        return;
    }

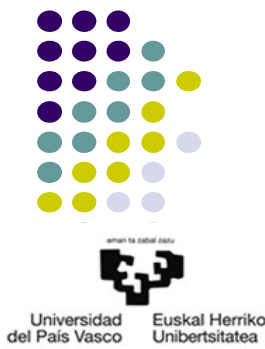
    public boolean getRemoveDB() {
        return removeDB;
    }
}
```

MB: pl.DbMB.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 3. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



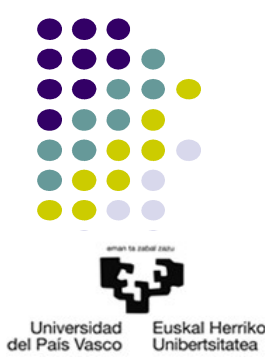
```
<!-- AURREKO BERTSIOKO KODE BERDINA -->
<h:form>
  <h1>SAR ITZAZU ERABILTZAILE-DATUAK</h1>
  <label>Izena: </label>
  <h:inputText id="izena" value="#{erabiltzaileaMB.izena}" required="true" requiredMessage="ERROREA: izena sartzea derrigorrezkoa da"
/>
  <h:message for="izena" style="color:blue" /><br />
  <label>Pasahitza: </label>
  <h:inputText id="pasahitza" value="#{erabiltzaileaMB.pasahitza}" required="true" requiredMessage="ERROREA: pasahitza sartzea
derrigorrezkoa da" />
  <h:message for="pasahitza" style="color:blue" /><br />
  <label>E-posta: </label>
<h:inputText id="eposta" value="#{erabiltzaileaMB.eposta}" /><br />
  <h:commandButton action="3Amaiera" actionListener="#{dbMB.addZerrendaDB(zerrendaMB.zerrenda,erabiltzaileaMB,zerrendaMB.sesioa)}"
id="submitAdd" value="SESIOA DBan GRABATU" />
  <h:commandButton action="3Amaiera" actionListener="#{dbMB.removeErabiltzaileaDB(erabiltzaileaMB)}" id="submitRemove"
value="ERABILTZAILEAREN DATU GUZTIAK EZABATU" />
</h:form>
</p>
</ui:define>
</ui:composition>
</html>
```

JSF: 2SesioLaburpena.xhtml

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 3. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



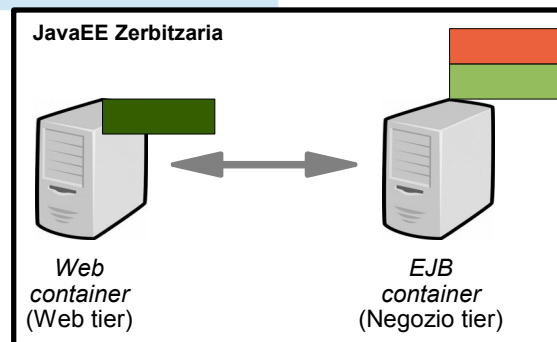
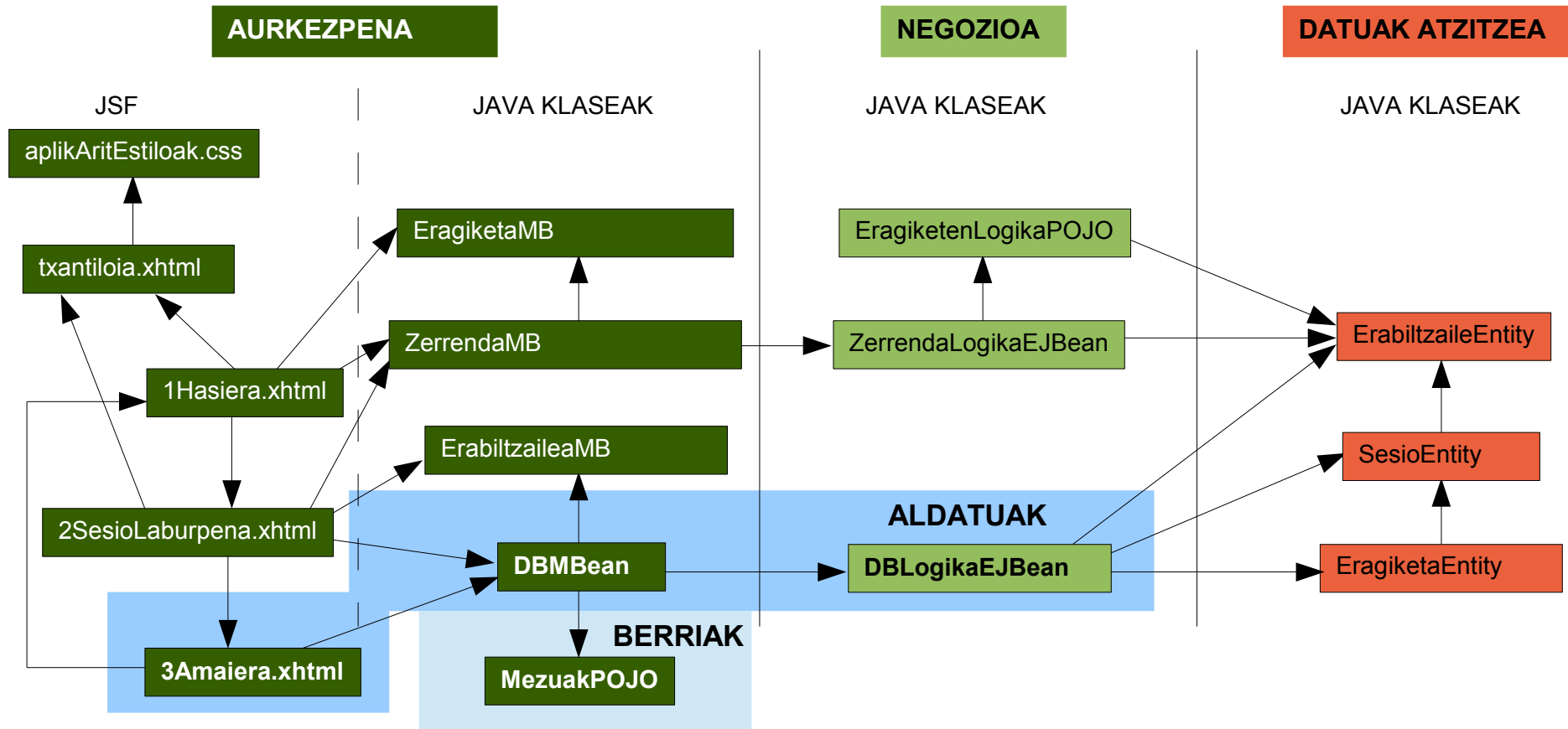
JSF: 3Amaiera.xhtml

```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml" ...>
<ui:composition template="/txantiloia.xhtml">
<ui:define name="DokumentuIzenburua">3Amaiera: Aplikazio aritmetikoaren amaierako orrialdea</ui:define>
<ui:define name="OrrialdeIzenburua">APLIKAZIO ARITMETIKOA JSF, EJB eta JParekin: AMAIERA</ui:define>
<ui:define name="Edukia">
  <h:panelGroup rendered="#{dbMB.addDB}">
    <h1>DBan GORDETA KO DATUAK HAUEK IZAN DIRA</h1>
    Erabiltzailea: #{dbMB.erabiltzaileaDB.izena}<br />
    Sesioa: #{dbMB.sesioaDB.cookie}<br />
    <h:dataTable value="#{dbMB.zerrendaDB}" var="eragiketa" border="2">
      <h:column>
        <f:facet name="header">ERAGIKETAK</f:facet>
        #{eragiketa.er1}#{eragiketa.er}#{eragiketa.er2}
      </h:column>
      <h:column>
        <f:facet name="header">EMAITZAK</f:facet>
        #{eragiketa.em}
      </h:column>
    </h:dataTable>
  </h:panelGroup>
  <h:panelGroup rendered="#{dbMB.removeDB}">
    <h1>DBtik EZABATUTZEN SAIATUTAKO ERABILTZAILEA</h1>
    Erabiltzailea: #{erabiltzaileaMB.izena}<br />
  </h:panelGroup>
  <p>
    <h:form>
      <h:commandButton action="1Hasiera" actionListener="#{zerrendaMB.endSesioa()}" id="submit" value="SESIO BERRIA HASI" />
    </h:form>
  </p>
</ui:define>
</ui:composition>
</html>
```

3.- Java EE

3.2.- Negozio tier

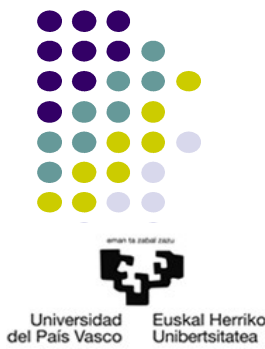
- JSF+EJB+JP 4. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu; erroreak kontrolatu



3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 4. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu; erroreak kontrolatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
package bl;
//BEHARREZKO import LERROAK

@Stateful
@LocalBean
public class DbLogikaEJB {

    @PersistenceContext
    private EntityManager em;
    private ErabiltzaileaEntity erabiltzaileaDB;
    private SesioaEntity sesioaDB;
    private int dbKodea;

    public DbLogikaEJB() {
    }

    public void addZerrendaDB(List<EragiketaEntity> zerrenda, ErabiltzaileaEntity erabiltzailea, SesioaEntity sesioa) {
        addErabiltzaileaDB(erabiltzailea);

        if(erabiltzaileaDB!=null) {
            addSesioaDB(sesioa);

            EragiketaEntity e;
            for(int i=0;i<zerrenda.size();i++) {
                e=zerrenda.get(i);
                e.setOrdena(i+1);
                e.setSesioaBean(sesioaDB);
                em.persist(e);
            }
            dbKodea=0; //Guzti hau ondo egin bada, kodea 0 izan behar da
        }
        //DBan erabiltzailea ez badago, errore-kodea beste metodoan sartu da

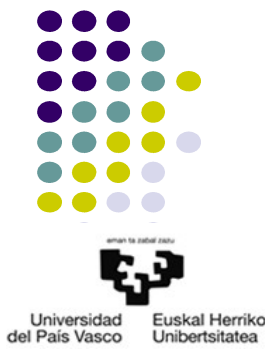
        return;
    }
    //HEMENGO KODE GUZTIA HURRENGO ORRIALDEAN DAGO
```

EJB: bl.DbLogikaEJB.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 4. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu; erroreak kontrolatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



EJB: bl.DbLogikaEJB.java

```
//HEMENGOKODEGUZTIAAURREKORRRIALDEANDAGO
```

```
privatevoidaddErabiltzaileaDB(ErabiltzaileaEntitye){
    try{
        erabiltzaileaDB=(ErabiltzaileaEntity)em.createNamedQuery("ErabiltzaileaEntity.findErabiltzailea").setParameter("izena",e.getIzena()).getSingleResult();

        if(!((erabiltzaileaDB.getPasahitza()).equals(e.getPasahitza()))){
            erabiltzaileaDB=null;
            dbKodea=1;
        }
    }
    catch(Exceptionex){
        erabiltzaileaDB=e;
        em.persist(erabiltzaileaDB);
    }
    return;
}

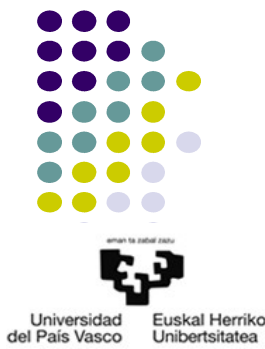
privatevoidaddSesioaDB(SesioaEntitysesioa){
    {
        sesioaDB=sesioa;
        sesioaDB.setErabiltzaileaBean(erabiltzaileaDB);
        em.persist(sesioaDB);
        return;
    }
}

//HEMENGOKODEGUZTIAHURRENGORRRIALDEANDAGO
```


3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 4. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu; erroreak kontrolatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
//HEMENGO KODE GUZTIA AURREKO ORRIALDEAN DAGO
```

EJB: bl.DbLogikaEJB.java

```
public ErabiltzaileaEntity getErabiltzaileaDB() {
    try {
        erabiltzaileaDB=em.find(ErabiltzaileaEntity.class, erabiltzaileaDB.getIdErabiltzailea());
    } catch(Exception ex) {
        dbKodea=2; //Erabiltzailea ez badago DBan edo ezin izan bada lortu DBtik, errore-kodea
    }

    return erabiltzaileaDB;
}

public SesioaEntity getSesioaDB() {
    try {
        sesioaDB=(SesioaEntity)em.find(SesioaEntity.class, sesioaDB.getIdSesioa());
    } catch(Exception ex) {
        dbKodea=2; //Sesioa ez badago DBan edo ezin izan bada lortu DBtik, errore-kodea
    }

    return sesioaDB;
}

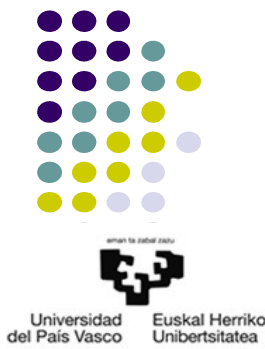
public List<EragiketaEntity> getZerrendaDB() {
    List<EragiketaEntity> zerrendaDB=null;
    try {
        zerrendaDB =
(List<EragiketaEntity>)em.createNamedQuery("EragiketaEntity.findAllSesioa").setParameter("sesioa",sesioaDB).getResultList();
    } catch(Exception ex) {
        dbKodea=2; //ERAGIKETA-ZERRENDA DBan EZ BADAGO EDO EZIN IZAN BADA DBtik LORTU: ERROREA
    }

    return (List<EragiketaEntity>)zerrendaDB;
}
//HEMENGO KODE GUZTIA HURRENGO ORRIALDEAN DAGO
```

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 4. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu; erroreak kontrolatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



EJB: bl.DbLogikaEJB.java

```
//HEMENGOKODEGUZTIAAURREKORRRIALDEANDAGO
publicvoidremoveErabiltzaileaDB(ErabiltzaileaEntitye){
    ErabiltzaileaEntityerabiltzaileaBean=null;
    dbKodea=3;

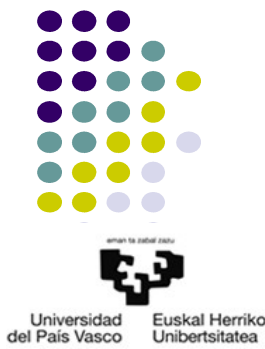
    try
    {
erabiltzaileaBean=(ErabiltzaileaEntity)em.createNamedQuery("ErabiltzaileaEntity.findErabiltzailea").setParameter("izena",e.getIzena()).getSingleResult();
        if((erabiltzaileaBean.getPasahitza()).equals(e.getPasahitza()))
            em.remove(erabiltzaileaBean);
        else
            dbKodea=4; //SARTUTAKOPASAHITZAGORDETAKOARENBERDINAEZBADA:ERROREA+EZDAEZEREZABATUKO
    }
    catch(Exceptionex)
    {
        dbKodea=5; //ERABILTZAILEADBAN EZBADAGO:ERROREA+EZDAEZEREZABATUKO
    }
    return;
}

publicintgetDbKodea(){
    returndbKodea;
}
}
```

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 4. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu; erroreak kontrolatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
package pl;

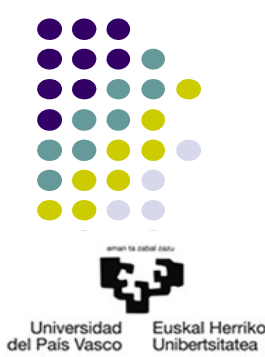
public abstract class MezuakPOJO {
    public final static String mezuak[]=
    {
        "GUZTIA ONDO GORDE DA DBan",//0
        "PASAHTZA OKERRA DA: ez da ezer gorde",//1
        "ZERBAIT EZ DA ONDO GORDE DBan",//2
        "ERABILTZAILEAREN DATU GUZTIAK ONDO EZABATU DIRA DBan",//3
        "PASAHTZA OKERRA DA: ez da ezer ezabatu",//4
        "ADIERAZITAKO ERABILTZAILEA EZ DAGO DBan: ez da ezer ezabatu"//5
    };
}
```

POJO: pl.MezuakPOJO.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 4. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu; erroreak kontrolatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
package pl;

//BEHARREZKO import LERROAK
@ManagedBean
@RequestScoped
public class DbMB {
    @EJB
    private DbLogikaEJB db;
//Aurreko bertsioko addDB eta removeDB jadanik ez dira behar
    private int dbKodea;

    public void addZerrendaDB(List<EragiketaEntity> zerrenda, ErabiltzaileaMB eMB, SesioaEntity se) {
        ErabiltzaileaEntity ee=new ErabiltzaileaEntity(eMB.getEposta(),eMB.getIzena(),eMB.getPasahitza());

        db.addZerrendaDB(zerrenda,ee,se);
//
        addDB=true;
        dbKodea=db.getDbKodea();
        return;
    }
//getAddDB(), getREmoveDB(): BERTSIOKO HONETAN EZ DIRA BEHAR
//getZerrendaDB(), getSesioaDB(), getErabiltzaileaDB(), getREmoveDB(): AURREKO BERTSIOKO KODE BERDINA

    public void removeErabiltzaileaDB(ErabiltzaileaMB eMB) {
        ErabiltzaileaEntity ee=new ErabiltzaileaEntity(eMB.getEposta(),eMB.getIzena(),eMB.getPasahitza());

        db.removeErabiltzaileaDB(ee);
//
        removeDB=true;
        dbKodea=db.getDbKodea();
        return;
    }

    public int getDbKodea() {
        return dbKodea;
    }

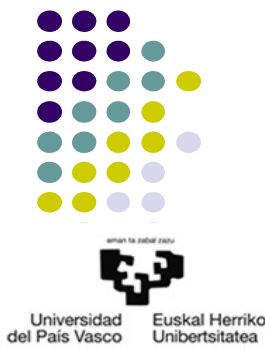
    public String getDbMezua() {
        return MezuakPOJO.mezuak[dbKodea];
    }
}
```

MB: pl.DbMB.java

3.- Java EE

3.2.- Negozio tier

- JSF+EJB+JP 4. ADIBIDEA: Aplikazio aritmetikoa JSF, EJB eta ERLAZIONATUTAKO JPEntities-ekin; DBtik datuak ezabatu; erroreak kontrolatu
 - Planteamendu posiblea
 - JavaEE aplikazioa



```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml" ...>
<ui:composition template="/txantiloia.xhtml">
<ui:define name="DokumentuIzenburua">3Amaiera: Aplikazio aritmetikoaren amaierako orrialdea</ui:define>
<ui:define name="OrrialdeIzenburua">APLIKAZIO ARITMETIKOA JSF, EJB eta JPArekin: AMAIERA</ui:define>
<ui:define name="Edukia">
  <h1>#{dbMB.dbMezua}</h1>
  <h:panelGroup rendered="#{dbMB.dbKodea==0}">
    Erabiltzailea: #{dbMB.erabiltzaileaDB.izena}<br />
    Sesioa: #{dbMB.sesioaDB.cookie}<br />
    <h:dataTable value="#{dbMB.zerrendaDB}" var="eragiketa" border="2">
      <h:column>
        <f:facet name="header">ERAGIKETAK</f:facet>
        #{eragiketa.er1}#{eragiketa.er}#{eragiketa.er2}
      </h:column>
      <h:column>
        <f:facet name="header">EMAITZAK</f:facet>
        #{eragiketa.em}
      </h:column>
    </h:dataTable>
  </h:panelGroup>
  <h:panelGroup rendered="#{dbMB.dbKodea==1}">
    Erabiltzailea: #{erabiltzaileaMB.izena}<br />
  </h:panelGroup>
  <h:panelGroup rendered="#{dbMB.dbKodea==3 or dbMB.dbKodea==4 or dbMB.dbKodea==5}">
    Erabiltzailea: #{erabiltzaileaMB.izena}<br />
  </h:panelGroup>
  <p>
    <h:form>
      <h:commandButton action="1Hasiera" actionListener="#{zerrendaMB.endSesio()}" id="submit" value="SESIO BERRIA HASI" />
    </h:form>
  </p>
</ui:define>
</ui:composition>
</html>
```

JSF: 3Amaiera.xhtml

4.- ERREFERENTZIAK

- Java Platform, Enterprise Edition (Java EE) 7
 - Persistence: <http://docs.oracle.com/javaee/7/tutorial/partpersist.htm#BNBPY>