

CURSO DE JAVASERVER FACES

EJERCICIO

INTEGRACIÓN JSF Y JAVA EE



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

El objetivo de este ejercicio es integrar algunas de las tecnologías empresariales en nuestro proyecto de Bolsa de Trabajo, con el objetivo de obtener el catálogo de código postal de una Base de Datos utilizando Java EE e integrándolo con JSF. El resultado será similar a la siguiente figura:

The image shows a web form titled "Bolsa de Trabajo" with a purple header. The form contains the following fields and elements:

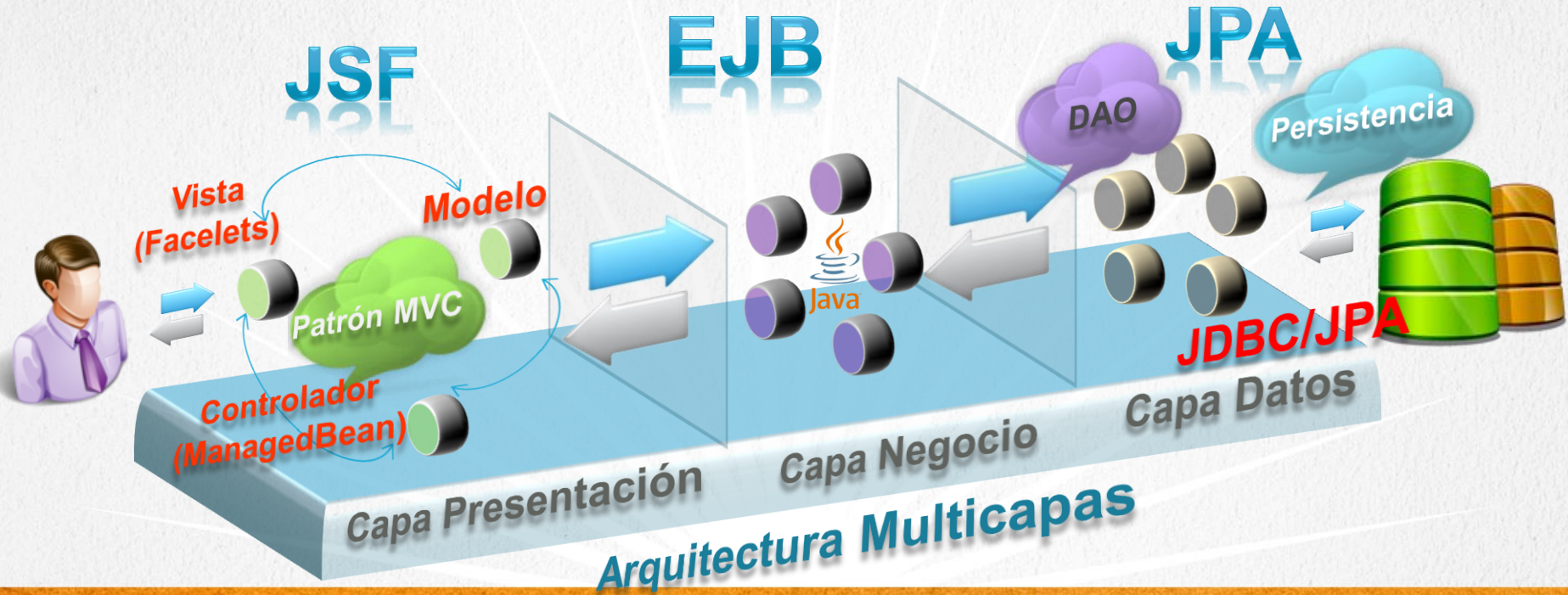
- Nombre:
- Apellido:
- Sueldo deseado:
- Fecha de Nacimiento:
- Código Postal: (highlighted with an orange oval)
- Colonia: (dropdown menu)
- Ciudad:
- Mostrar:

A black "Enter" button is positioned over the "Código Postal" field. Three green curved arrows point from the "Código Postal" field towards the "Colonia" and "Ciudad" fields, indicating a data flow or relationship.

Derechos Reservados - GlobalMentoring.com.mx - UniversidadJava.com

ARQUITECTURA DE LA APLICACIÓN

La arquitectura de la aplicación utilizando Java EE es similar a la siguiente:

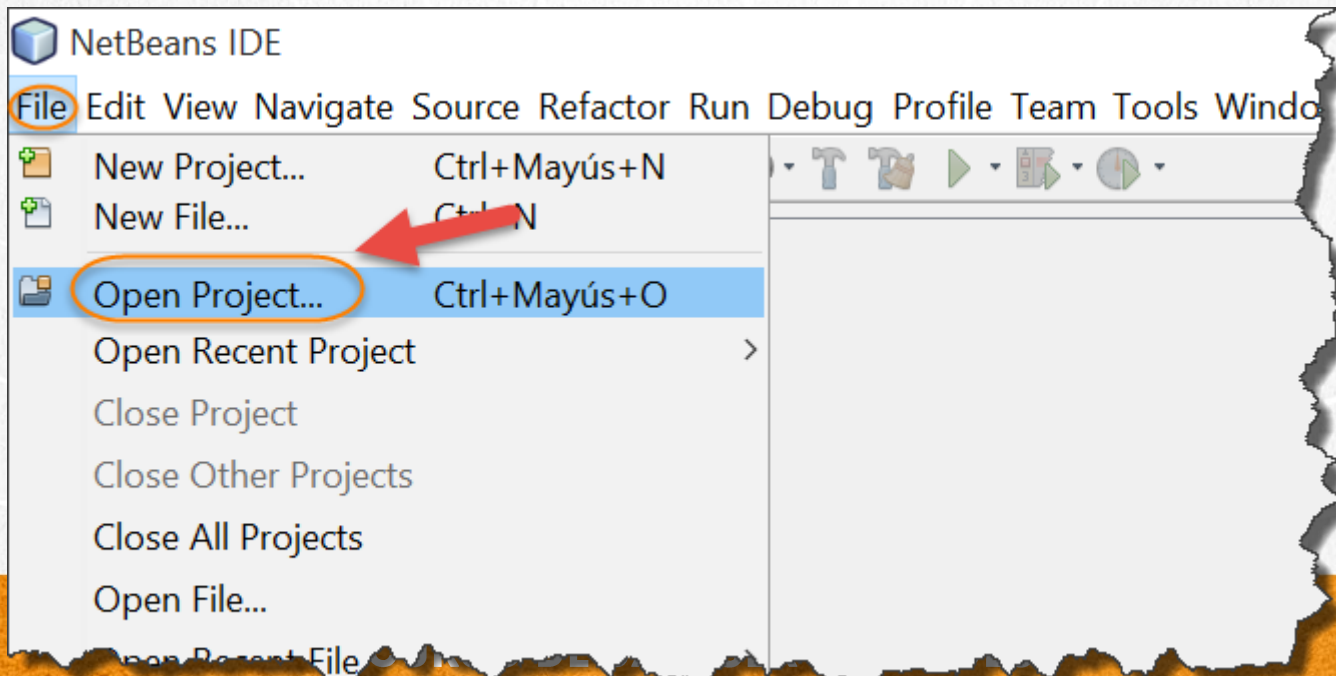


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

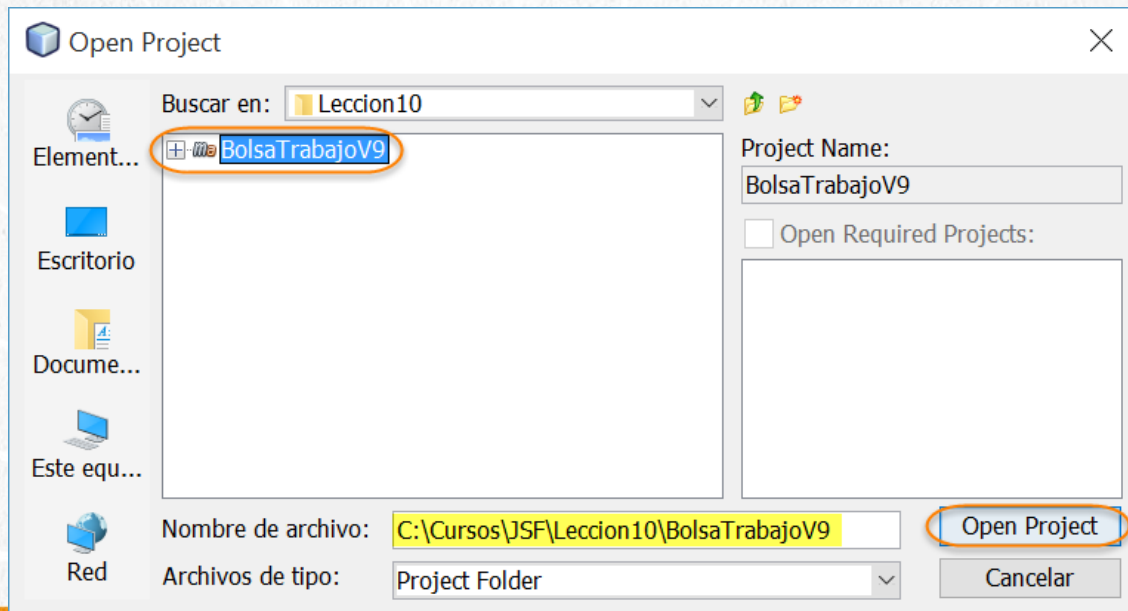
PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto BolsaTrabajoV9, sólo en caso de que no lo tengamos ya abierto:



PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto BolsaTrabajoV9, sólo en caso de que no lo tengamos ya abierto:

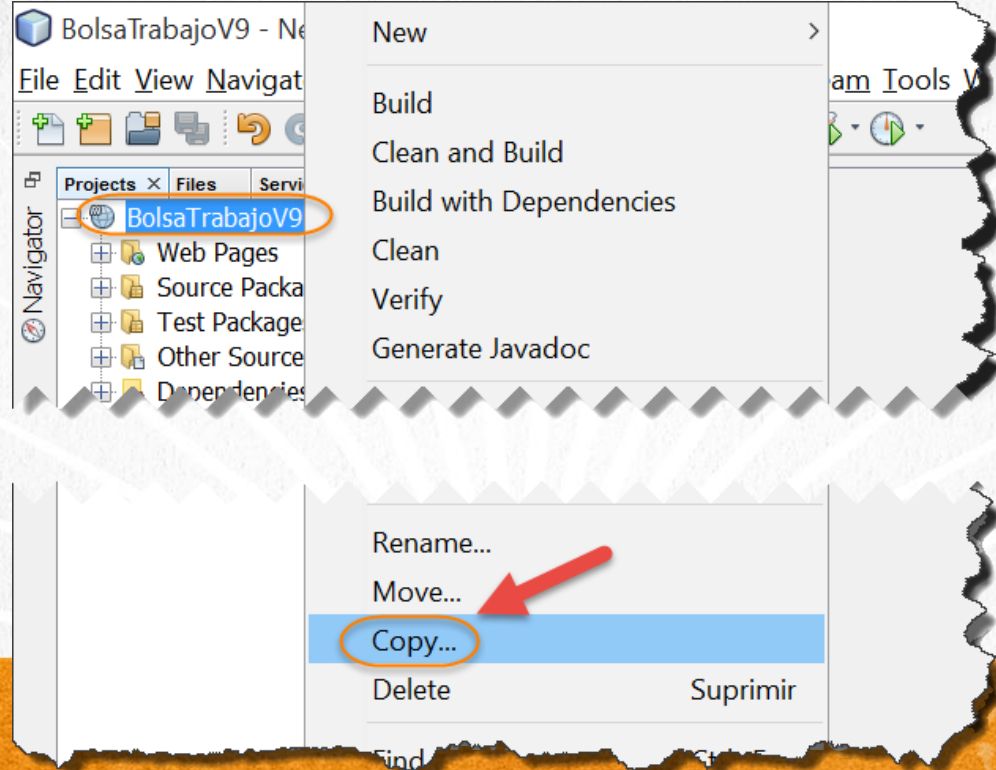


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

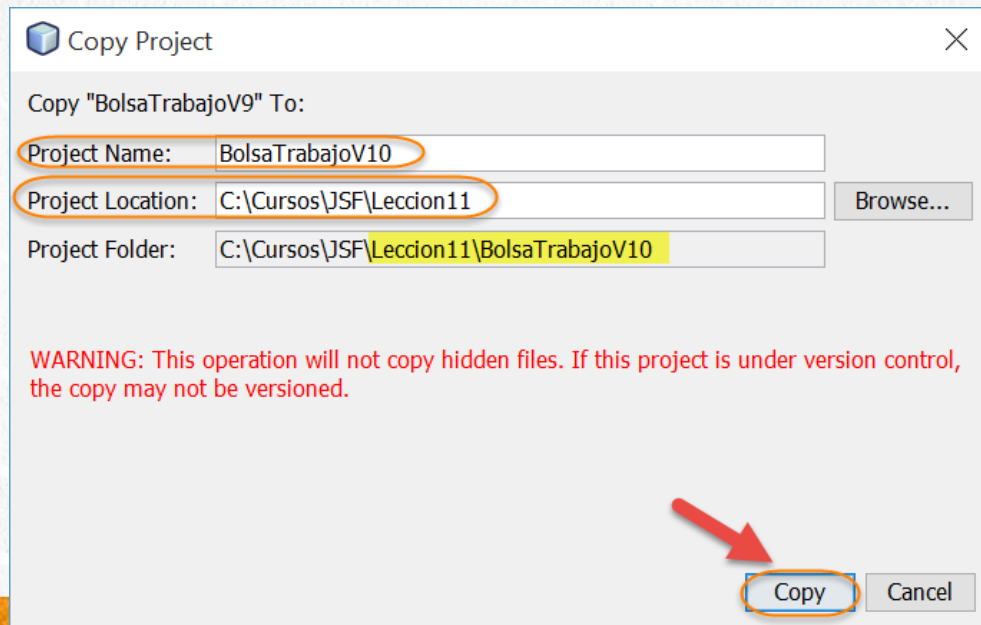
PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto BolsaTrabajoV9:



PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto BolsaTrabajoV9, lo renombramos a BolsaTrabajoV10 y lo depositamos en la nueva ruta mostrada:

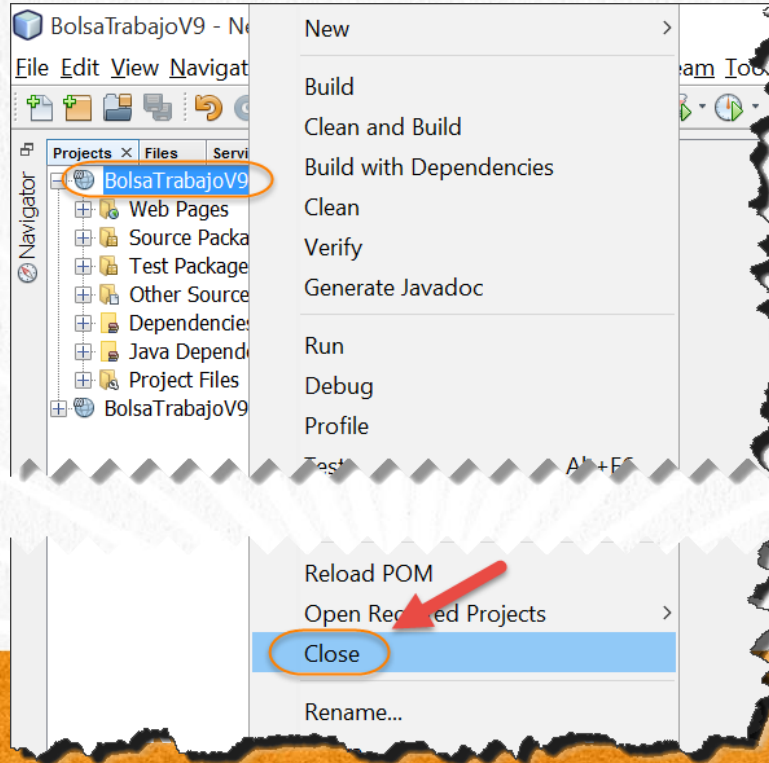


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

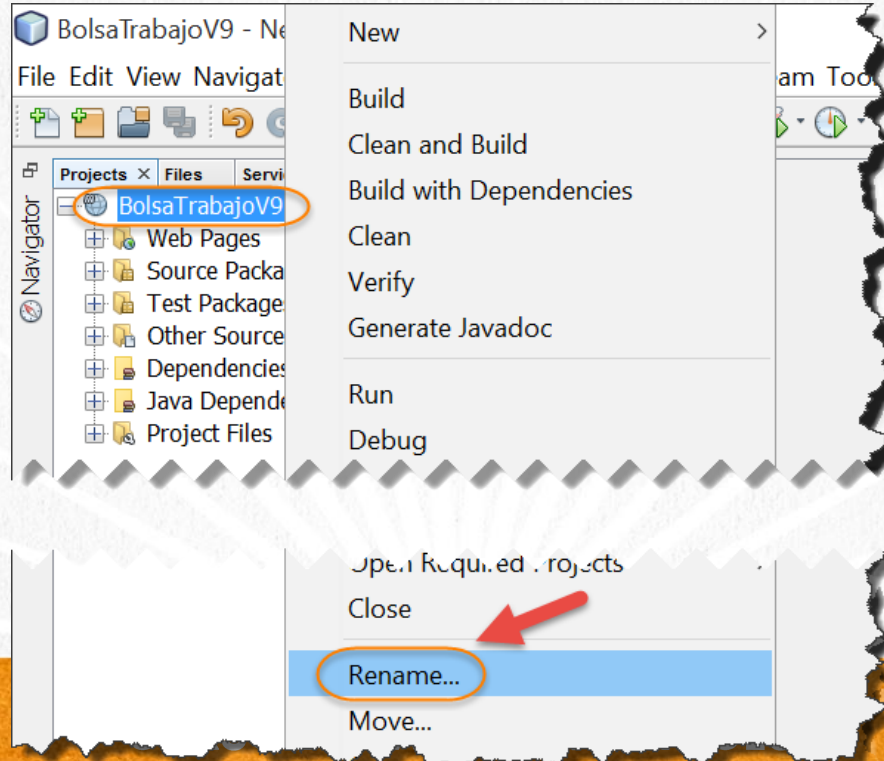
PASO 3. CERRAR PROYECTO

Cerramos el proyecto anterior, y dejamos solo el nuevo abierto:



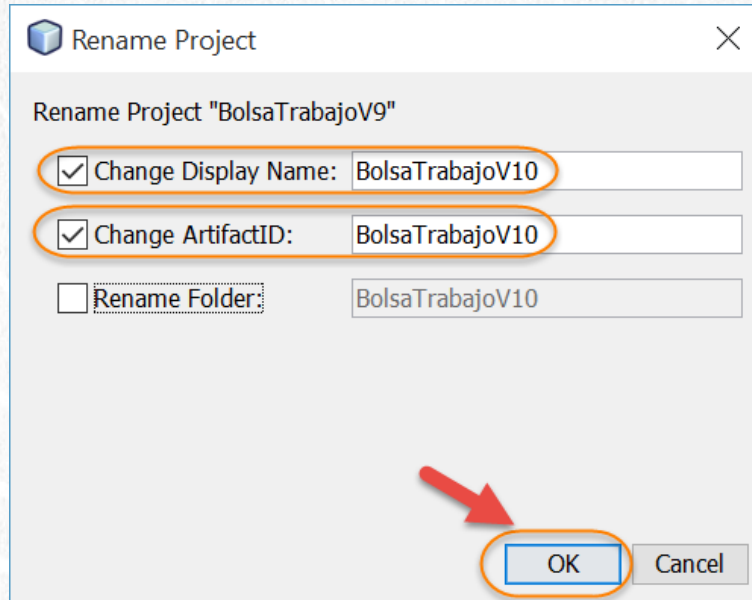
PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV10:



PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV10:



PASO 5. INSTALACIÓN DE MYSQL

Si aún no se tiene instalado MySql y MySql Workbench utilizar la siguiente guía:

<http://icursos.net/cursos/Instalaciones/CJ-B-Ejercicio-07-InstalacionMySql.pdf>

Nota: Si ya se tienen instaladas las herramientas de MySql omitir este paso.

PASO 6. CREAR UNA BASE DE DATOS

Ahora vamos a crear la base de datos de **bolsatrabajo**, junto con la tabla de **colonia** utilizando MySql Workbench. Para crear la base de datos seguiremos la siguiente guía:

<http://icursos.net/cursos/Instalaciones/CJ-B-Ejercicio-17-CreacionBaseDatosMySqlJSF.pdf>



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 7. CREAR CONEXION JTA EN GLASSFISH

Crear una conexión vía JTA utilizando Glassfish. Vamos a seguir la siguiente guía para realizar esta tarea:

<http://icursos.net/cursos/Instalaciones/CJ-B-Ejercicio-16-CreacionPoolConexionesGlassfishJSF.pdf>



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 8. MODIFICAR UNA JAVA

Modificamos la clase Colonia.java. Esta clase la modificaremos para convertirla en una clase de Entidad y utilizará la tecnología de JPA.

Esta tecnología se ve a detalle en el curso de Java EE y Hibernate & JPA. En este ejercicio solo veremos la integración con Java EE y JSF.

Veamos como queda la clase de entidad Colonia.java ya con los cambios de JPA.



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 9. MODIFICAMOS EL CÓDIGO

Archivo Colonia.java:

Clic para descargar código

```
package beans.domain;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQuery;
import javax.persistence.Table;

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

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "colonia_id")
    private long coloniaId;

    @Column(name = "nombre_colonia")
    private String nombreColonia;
```

PASO 9. MODIFICAMOS EL CÓDIGO

Archivo Colonia.java:

Clic para descargar código

```
@Column(name = "codigo_postal")
private long codigoPostal;

public Colonia() {
}

public Colonia(long coloniaId, String nombreColonia, long codigoPostal) {
    this.coloniaId = coloniaId;
    this.nombreColonia = nombreColonia;
    this.codigoPostal = codigoPostal;
}

public long getColoniaId() {
    return coloniaId;
}

public void setColoniaId(long coloniaId) {
    this.coloniaId = coloniaId;
}

public String getNombreColonia() {
    return nombreColonia;
}

public void setNombreColonia(String nombreColonia) {
    this.nombreColonia = nombreColonia;
}
```

PASO 9. MODIFICAMOS EL CÓDIGO

Archivo Colonia.java:

Clic para descargar código

```
public long getCodigoPostal() {  
    return codigoPostal;  
}  
  
public void setCodigoPostal(long codigoPostal) {  
    this.codigoPostal = codigoPostal;  
}  
  
@Override  
public int hashCode() {  
    int hash = 7;  
    hash = 73 * hash + (int) (this.coloniaId ^ (this.coloniaId >>> 32));  
    return hash;  
}
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 9. MODIFICAMOS EL CÓDIGO

Archivo Colonia.java:

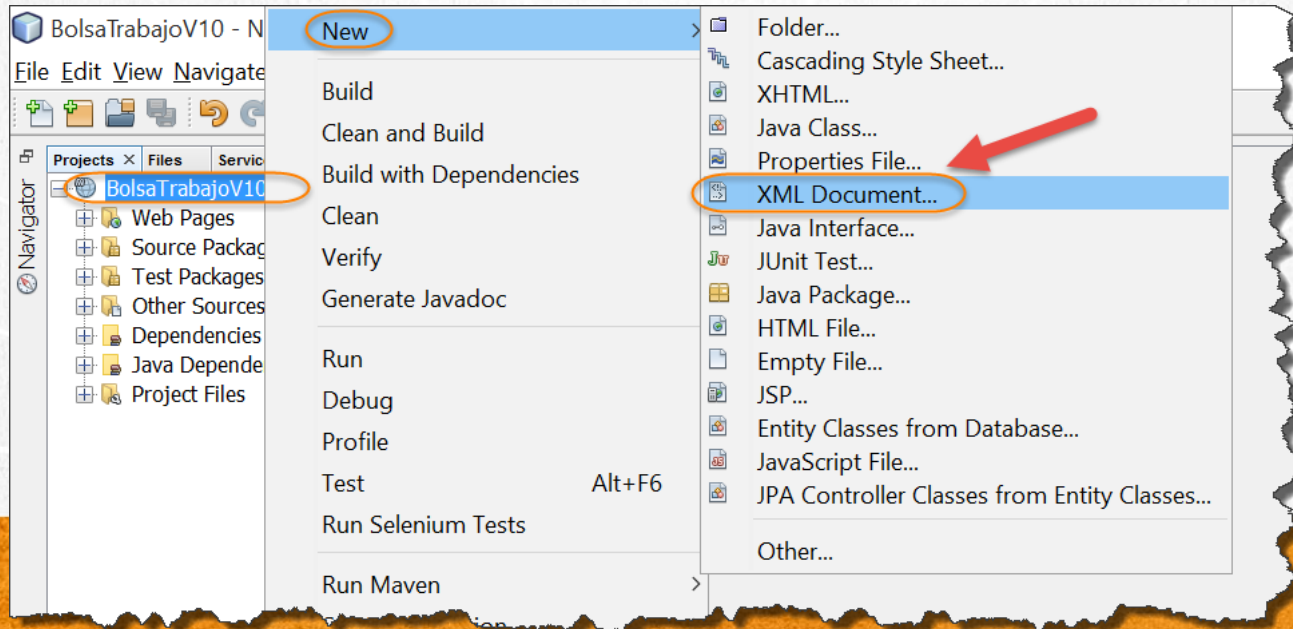
Clic para descargar código

```
@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Colonia other = (Colonia) obj;
    if (this.coloniaId != other.coloniaId) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Colonia{" + "coloniaId=" + coloniaId + ", nombreColonia=" + nombreColonia + ", codigoPostal=" +
codigoPostal + '}';
}
}
```

PASO 10. CREAR ARCHIVO XML

Creamos el archivo persistence.xml. Para que la clase de entidad Colonia.java pueda conectarse a la base de datos, vamos a crear un archivo que será la conexión entre nuestra aplicación de Java y el pool de conexiones que creamos anteriormente en Glassfish. Este archivo se llama persistence.xml y es parte de la tecnología JPA.



PASO 10. CREAR ARCHIVO XML

Creamos el archivo persistence.xml:

New XML Document

Steps

1. Choose File Type
2. **Name and Location**
3. Select Document Type
4. ...

Name and Location

File Name: persistence

Project: BolsaTrabajoV10

Folder: src/main/resources/META-INF Browse...

Created File: BolsaTrabajoV10\src/main/resources/META-INF\persistence.xml

Si no existe la carpeta de META-INF, solo con escribir la ruta mostrada se creará de manera automática.

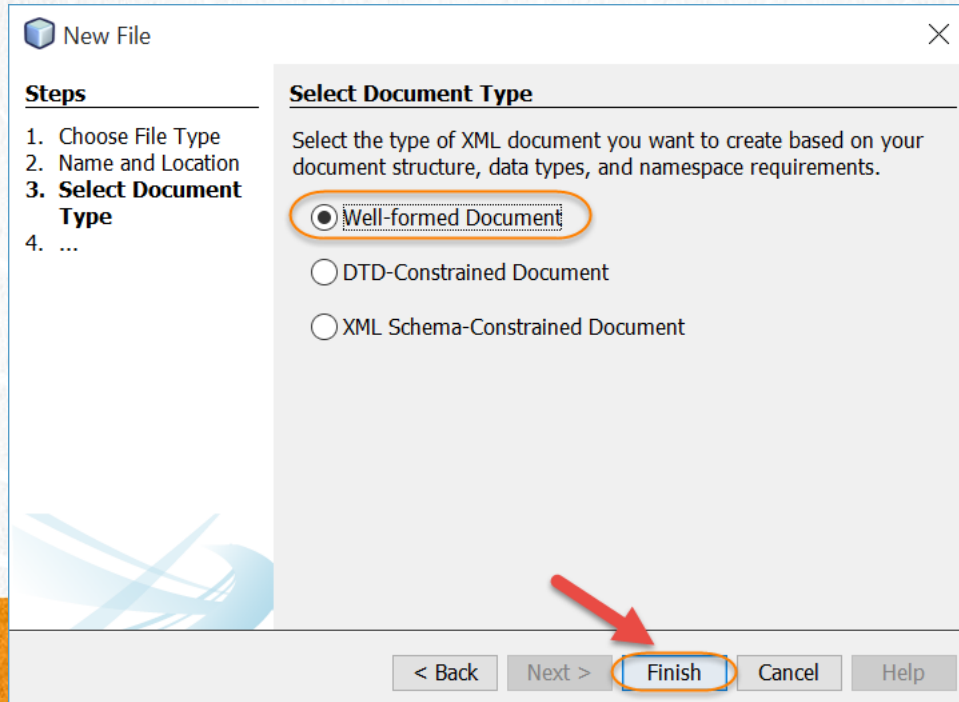
< Back **Next >** Finish Cancel Help

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 10. CREAR ARCHIVO XML

Creamos el archivo persistence.xml. Aquí podemos escoger cualquier opción, no importa, ya que vamos a sobrescribir el contenido del archivo xml:



PASO 11. MODIFICAMOS EL CÓDIGO

Archivo persistence.xml:

Clic para descargar código

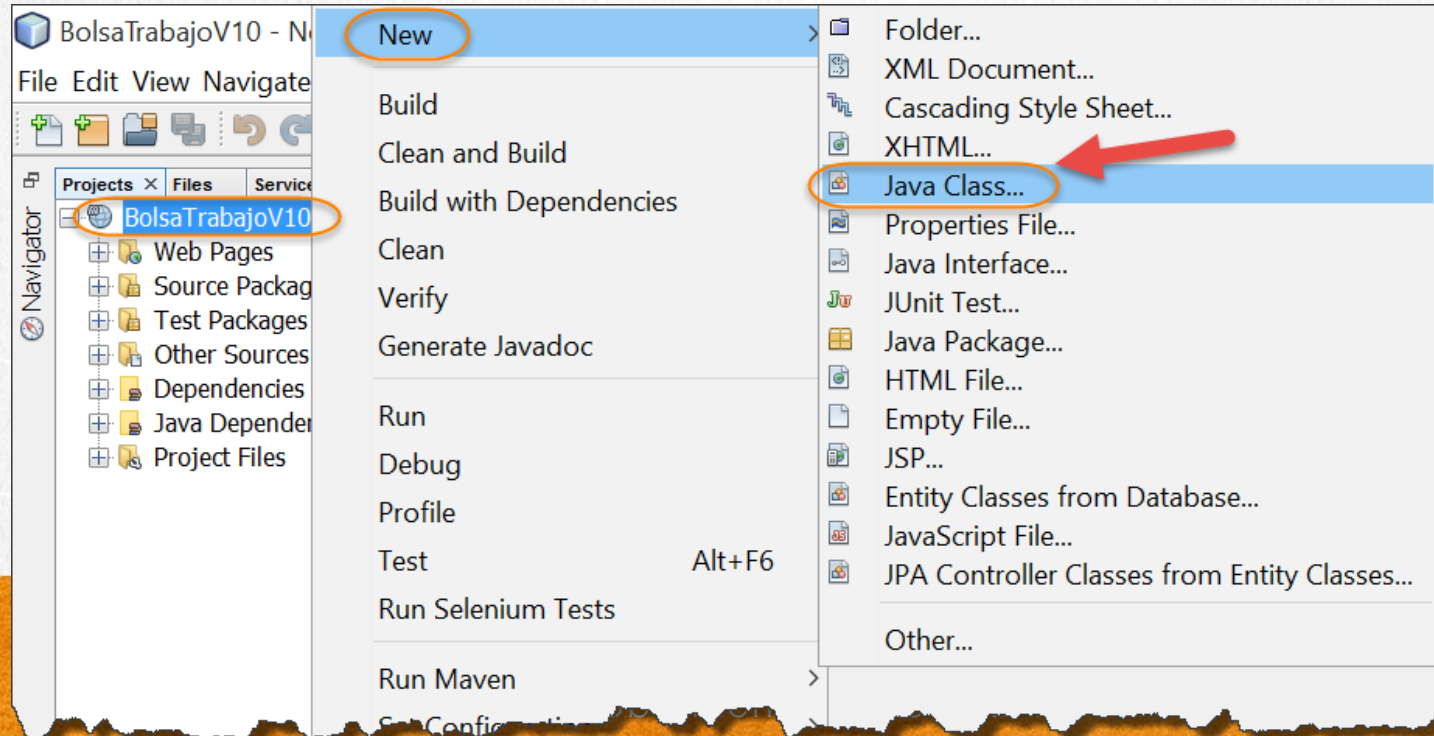
```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="BolsaTrabajoPU" transaction-type="JTA">
    <jta-data-source>jdbc/bolsatrabajo</jta-data-source>
    <properties>
      <property name="eclipselink.logging.level.sql" value="FINE"/>
      <property name="eclipselink.logging.parameters" value="true"/>
    </properties>
  </persistence-unit>
</persistence>
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 12. CREAR UN ARCHIVO JAVA

Creamos la clase ColoniaService.java. Esta clase utilizará la tecnología de EJB's. Esta tecnología se estudia a detalle en el curso de Java EE. En este ejercicio solo veremos la integración de Java EE y JSF.



PASO 12. CREAR UN ARCHIVO JAVA

Creamos la clase ColoniaService.java:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: ColoniaService

Project: BolsaTrabajoV10

Location: Source Packages

Package: beans.servicio

Created File: J:\SF\Leccion11\BolsaTrabajoV10\src\main\java\beans\servicio\ColoniaService.java

< Back Next > **Finish** Cancel Help

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 13. MODIFICAMOS EL CÓDIGO

[Archivo ColoniaService.java:](#)

[Clic para descargar código](#)

```
package beans.servicio;

import beans.domain.Colonia;
import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

@Stateless
public class ColoniaService {

    @PersistenceContext
    EntityManager em;

    public List<Colonia> findAll() {
        return em.createNamedQuery("Colonia.findAll").getResultList();
    }
}
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 14. MODIFICAR UN ARCHIVO JAVA

Modificamos la clase ColoniaHelper.java para inyectar la referencia del EJB ColoniaService, y así recuperar el listado de colonias desde el EJB, el cual regresa una lista de objetos de Entidad de tipo Colonia. Inyectamos la dependencia como sigue:

```
@Inject  
ColoniaService coloniaService;
```

Veamos como queda la clase ColoniaHelper.java:



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 14. MODIFICAMOS EL CÓDIGO

Archivo ColoniaHelper.java:

Clic para descargar código

```
package beans.helper;

import beans.domain.Colonia;
import beans.servicio.ColoniaService;
import java.util.ArrayList;
import java.util.List;
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
import javax.faces.model.SelectItem;
import javax.inject.Inject;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

@RequestScoped
@Named
public class ColoniaHelper {

    @Inject
    ColoniaService coloniaService;

    Logger log = LogManager.getRootLogger();

    public List<Colonia> getColonias() {
        List<Colonia> colonias = coloniaService.findAll();
        log.info("Recuperamos el listado de personas del EJB");
        return colonias;
    }
}
```

PASO 14. MODIFICAMOS EL CÓDIGO

Archivo ColoniaHelper.java:

Clic para descargar código

```
public long getColoniaIdPorNombre(String nombreColonia) {
    long coloniaId = 0;
    List<Colonia> colonias = getColonias();//Lista de colonias
    for (Colonia colonia : colonias) {
        if (colonia.getNombreColonia().equals(nombreColonia)) {
            coloniaId = colonia.getColoniaId();
            break;
        }
    }
    return coloniaId;
}

public long getColoniaIdPorCP(long codigoPostal) {
    long coloniaId = 0;
    List<Colonia> colonias = getColonias();
    for (Colonia colonia : colonias) {
        if (colonia.getCodigoPostal() == codigoPostal) {
            coloniaId = colonia.getColoniaId();
            break;
        }
    }
    return coloniaId;
}
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 14. MODIFICAMOS EL CÓDIGO

Archivo ColoniaHelper.java:

Clic para descargar código

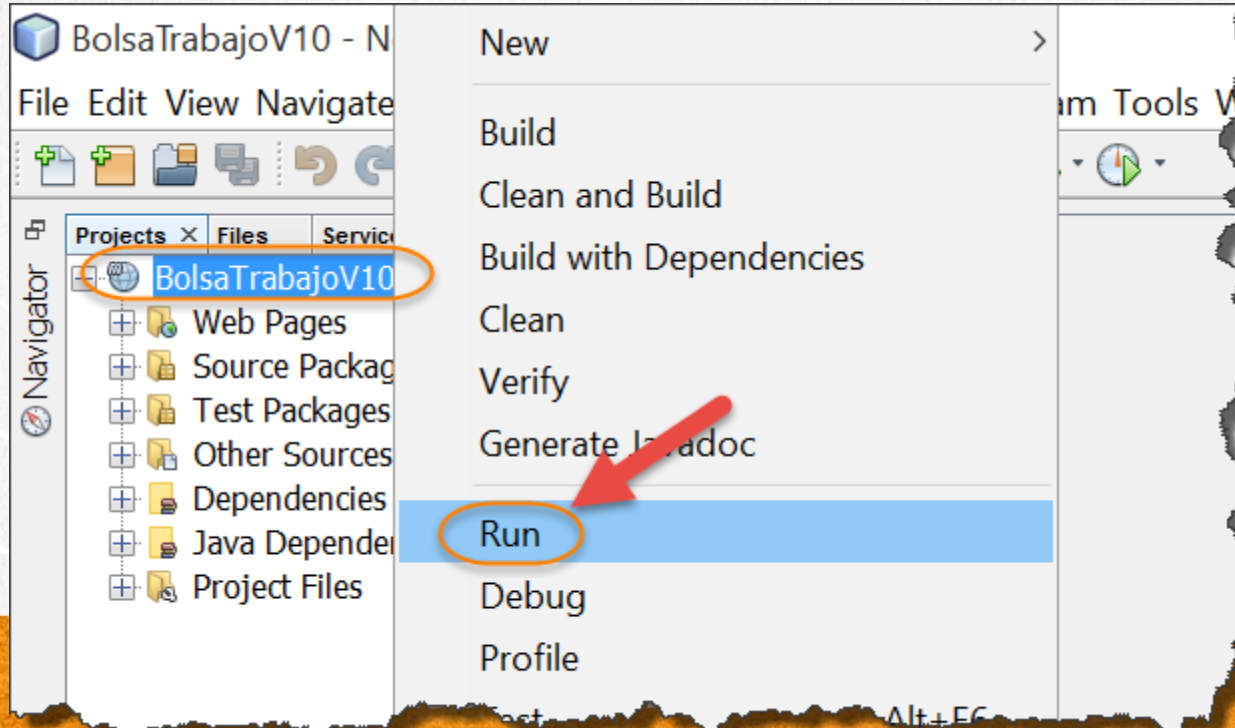
```
public List<SelectedItem> getSelectItems() {  
    List<SelectedItem> selectItems;  
    selectItems = new ArrayList<>();  
    List<Colonia> colonias = getColonias();  
    for (Colonia colonia : colonias) {  
        SelectedItem selectItem = new SelectedItem(colonia.getColoniaId(),  
            colonia.getNombreColonia());  
        selectItems.add(selectItem);  
    }  
    return selectItems;  
}
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 15. EJECUTAR LA APLICACIÓN

Ejecutamos la aplicación. Se recomienda detener el servidor de Glassfish si está levantado, hacer un clean & build de la aplicación y ejecutarla como sigue:



PASO 15. EJECUTAR LA APLICACIÓN

Ejecutamos la aplicación. El resultado es como sigue:



The screenshot displays a web application titled "Bolsa de Trabajo" (Job Market) with a purple header. Below the header is a registration form with the following fields and values:

Label	Value
Nombre	Introduce tu nombre
Apellido	Introduce tu Apellido
Sueldo deseado	0
Fecha de Nacimiento	
Código Postal	03810
Colonia	Nápoles
Ciudad	Ciudad de Mexico

Below the form fields are two buttons: "Mostrar" and "Enviar". The "Código Postal" field is highlighted with an orange oval, and a green arrow points from it to the "Enviar" button. A black "Enter" key icon is also shown next to the "Código Postal" field.

Derechos Reservados - GlobalMentoring.com.mx - UniversidadJava.com

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

TAREAS EXTRAS

Se deja como tarea extra revisar el log del servidor Java para observar que todo funcione según lo esperado.

Además, se deja como ejercicio opcional hacer una depuración paso a paso para observar como se va pasando por cada uno de los componentes de la aplicación de 3 capas que hemos aplicado con Java EE, utilizando las tecnologías de JSF, EJB y JPA.



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

En esta lección estudiamos el tema de integración de JSF y Java EE.

Se creó una aplicación de tres capas utilizando las tecnologías de JSF, EJB y JPA. El detalle de las tecnologías de EJB y JPA se estudia en el curso de Java EE. En este ejercicio se estudió solamente cómo integrarlo con JSF.

Con esto ya tenemos las bases para crear aplicaciones web con Java EE, utilizando como capa de presentación la tecnología de JavaServer Faces.



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

CURSO ONLINE

JAVASERVER FACES (JSF)

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx