

JAKARTA EE9 66

domingo, 30 de abril de 2023 09:50 p. m.

:::::::::::::::::::SECCION 66::::::::::::::::::

-CLASE 549-



MIGRANDO PROYECTO B8_62_541_webapp-session-cdi-jpa-hibernate DESDE TOMCAT A WILDFLY

PROYECTO CON WILDFLY : C3_66_549_webapp-session-cdi-Jpa-hibernate-wildfly

LEVANTA TOMCAT :

```
PS C:\Users\Fanta\Documents\5.-JAVA_MASTER_JAKARTA\A4_429_Cookies> cd C:\developjava\apache-tomcat-10.1.5\bin  
PS C:\developjava\apache-tomcat-10.1.5\bin> .\startup.bat
```

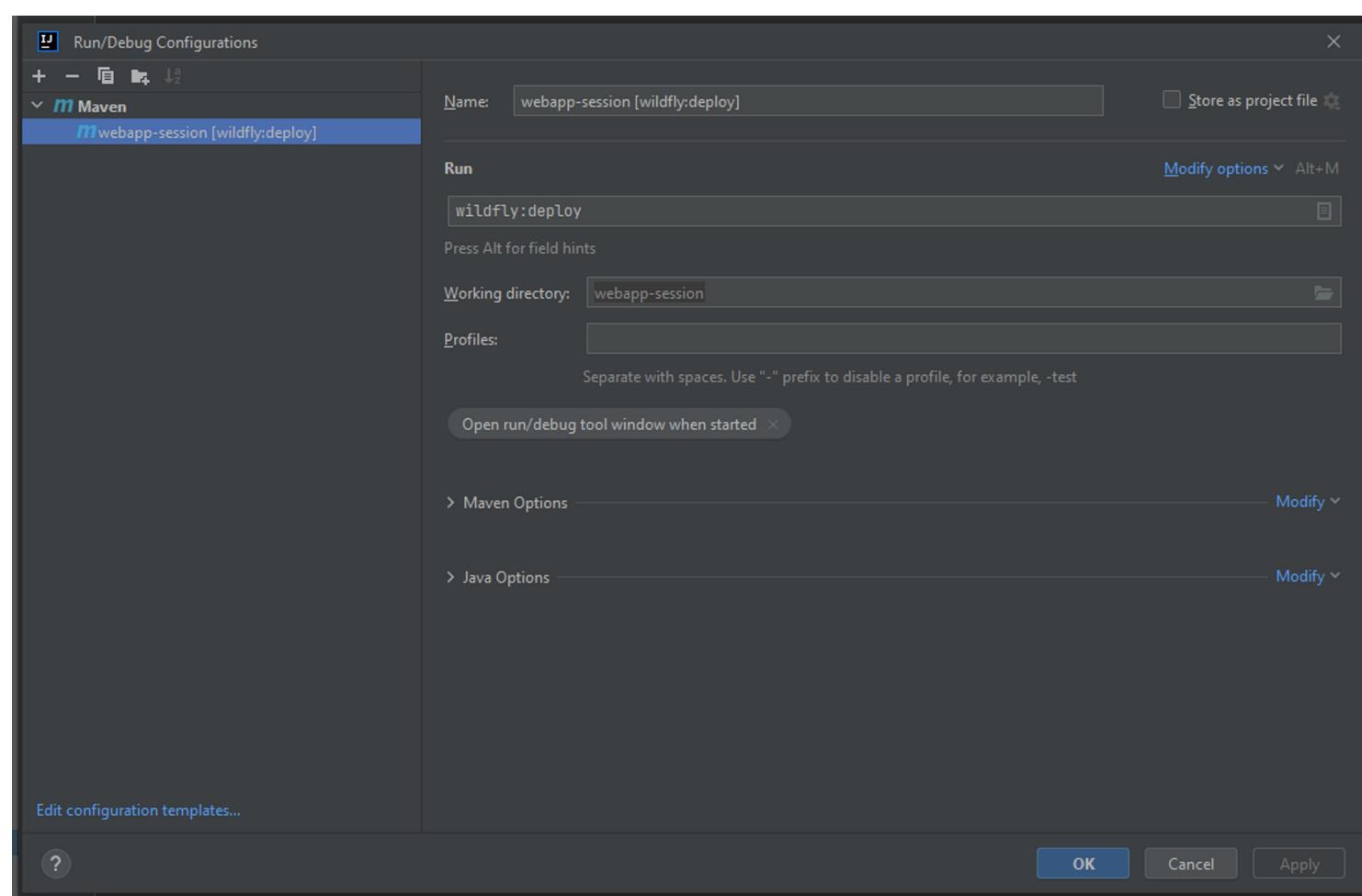
LEVANTA WILDFLY:

```
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> cd C:\developjava\wildfly-preview-26.1.3.Final\bin  
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> ./standalone.bat
```

REVISAR MODIFICACION EN CLASES, INTERFACES, XML, POM, ETC.

COMPARAR CAMBIOS ENTRE B8_62_541_webapp-session-cdi-jpa-hibernate y C3_66_549_webapp-session-cdi-Jpa-hibernate-wildfly

CONFIGURACION DEL RUN :



LUEGO DESPLIEGA WILDFLY

DALE RUN AL PROYECTO

REVISANDO DESPLIEGUE CON WILDFLY

A screenshot of the WildFly Management Console showing the 'Deployments' page. It lists five deployments: com.mysql:mysql-connector-j-8.0.32.jar, jee-master-ejb-remote.jar, jee-master-ejb.war, webapp-ear-ear.ear, and webapp-session.war (which is currently selected). The 'Main Attributes' for webapp-session.war are displayed: Name: webapp-session.war, Runtime Name: webapp-session.war, Context Root: /webapp-session, Hash: b4129a578b7e4573575014128ad5dc67f6c0d641, Enabled, Managed, Exploded: checked, Status: OK, Last enabled at: 4/30/23, 11:27 PM, and Last disabled at: n/a.

The screenshot shows the WildFly Management Console interface. The top bar displays the title 'msi | Management Console X Catalogo Servlet' and the URL '127.0.0.1:9990/console/index.html#deployment;de'. The left sidebar contains various icons for management tasks. The main content area is titled 'WildFly' and shows the deployment structure of 'webapp-session.war'. The structure is as follows:

- webapp-session.war
 - META-INF
 - MANIFEST.MF
 - context.xml
 - WEB-INF
 - classes
 - META-INF
 - persistence.xml
 - org
 - aguzman
 - apiservlet
 - webapp
 - headers
 - configs
 - controllers
 - filters
 - interceptors
 - listeners
 - models
 - repositories
 - services
 - beans.xml
 - web.xml
 - layout
 - carro.jsp
 - form.jsp
 - index.jsp
 - listar.jsp
 - login.jsp

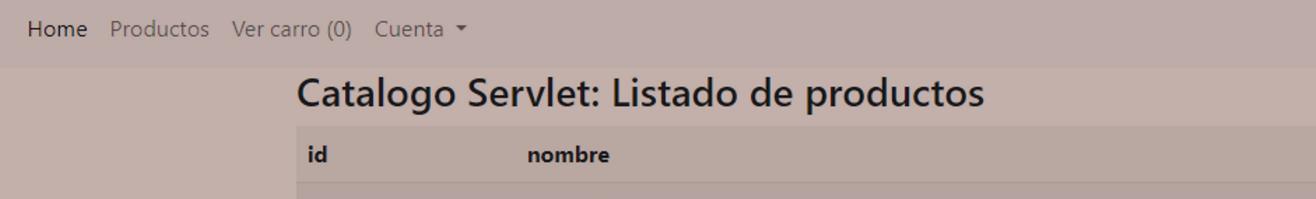
The screenshot shows the 'Catalogo Servlet' application running on WildFly. The browser address bar shows '127.0.0.1:8080/webapp-session/'. The page has a header with icons and a 'Navbar' menu. The main content area is titled 'Catalogo Servlet' and contains a blue header bar with the text 'Menu de opciones'. Below it is a list of links:

- [mostrar_productos.html](#)
- [login](#)
- [logout](#)
- [ver_carro](#)

PROBANDO CONEXIÓN

The screenshot shows the 'workbench' interface, specifically the 'SCHEMAS' section. The schema 'java-master-jakarta-2' is expanded, showing its structure:

- Tables
 - alumnos
 - categorias
 - Columns
 - id
 - nombre
 - Indexes
 - Foreign Keys
 - Triggers
 - clientes
 - clientes_detalles
 - curso
 - direccion
 - direcciones
 - facturas
 - productos
 - Columns
 - id
 - nombre
 - precio
 - categoria_id
 - sku
 - fecha_registro
 - Indexes
 - Foreign Keys
 - Triggers
 - tbl_alumnos_cursos
 - tbl_clientes_direcciones
 - usuarios
 - Columns
 - id
 - username
 - password
 - email
 - Indexes
 - Foreign Keys
 - Triggers
- Views



The screenshot shows a web application titled "Catalogo Servlet: Listado de productos" at the URL 127.0.0.1:8080/webapp-session/productos. The page includes a navigation bar with links for "Home", "Productos", "Ver carro (0)", and "Cuenta". The main content displays a table of products with columns for id, nombre, and tipo. The data is as follows:

| id | nombre | tipo |
|----|---------------|-------------|
| 1 | pc msi katana | computacion |
| 2 | teclado msi | computacion |
| 3 | lentes | computacion |
| 4 | pelota | deporte |
| 5 | raqueta | deporte |

Below the table, two messages are displayed: "algun valor global de la app!" and "guardando algun valor para el request".

A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:8080/webapp-session/login.html'. The main content area displays a large heading 'Hola admin has iniciado sesión con éxito!' in bold black font. Below it are two blue underlined links: 'volver' and 'cerrar sesión'. On the left side of the browser, there is a vertical toolbar with several icons: a red circle with a white letter 'O', a magnifying glass over a document, a key icon, a user profile icon, a person icon, a speech bubble icon, and a gear icon.

--CLASE 550--

550. EntityManager manejado por el contenedor con @PersistenceContext

PROYECTO : C4_66_550_webapp-session-cdi-Jpa-hibernate-wildfly-EntityManagerManejadoConContenedor

EN LA VERSION ANTERIOR DE ESTE PROYECTO, ESTA CON EL ENTITY MANAGER SE MANEJABA PROGRAMATICAMENTE.
MIENTRAS QUE EN ESTA VERSION, SE LE DICE AL CONTENEDOR DE JPA, EL CONTENEDOR DE PERSISTENCIA, QUE A SU VEZ ES MANEJADO POR EL CONTENEDOR DE WILDFLY, QUE NOS INYECTE UN ENTITY MANAGER

MORTIFICACIÓN DOM.

```
MODIFICACION_POM :  
src/main/java/manager.java           pom.xml (webapp-session)  ProducerResources.java  
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
         http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
    <modelVersion>4.0.0</modelVersion>  
  
    <groupId>org.aguzman.apiservlet.webapp.session</groupId>  
    <artifactId>webapp-session-2</artifactId> -----  
    <version>1.0-SNAPSHOT</version>  
    <packaging>war</packaging>  
  
    <properties>  
        <maven.compiler.source>17</maven.compiler.source>  
        <maven.compiler.target>17</maven.compiler.target>  
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  
    </properties>
```

MODIFICACIONES DEL CÓDIGO EN LAS CLASES DEL PACKAGE CONFIGS (`org.aguzman.apiservlet.webapp.headers.configs`) :

C:\Users\aguzman\Documents\NetBeansProjects\C4_66_550_webapp-session-cdi-Jpa-hibernate-wildfly-EntityManagerManejadoConContenedor - ProducerResources

```

16
17
18 //550
19
20 @ApplicationScoped
21 public class ProducerResources {
22
23     @Produces
24     @RequestScoped
25     @MySqlConnPrincipal
26     private Connection beanConnection() throws NamingException, SQLException {
27         //Context initContext = new InitialContext();
28         //Context envContext = (Context) initContext.lookup("java:/comp/env");
29         //DataSource ds = (DataSource) envContext.lookup("jdbc/mysqlDB");
30         return ds.getConnection();
31     }
32
33     @Produces
34     @RequestScoped
35     private Logger beanLogger(InjectionPoint injectionPoint){
36         return Logger.getLogger(injectionPoint.getMember().getDeclaringClass().getName());
37     }
38
39     @Disposes
40     public void close(@MySqlConnPrincipal Connection connection) throws SQLException {
41         connection.close();
42         log.info("Cerrando la conexión.");
43     }
44
45     //@Resource(name="jdbc/mysqlDB")
46     @Resource(lookup = "java:/MySqlDS")
47     private DataSource ds;
48
49     @Inject
50     private Logger log;
51 }
52

```

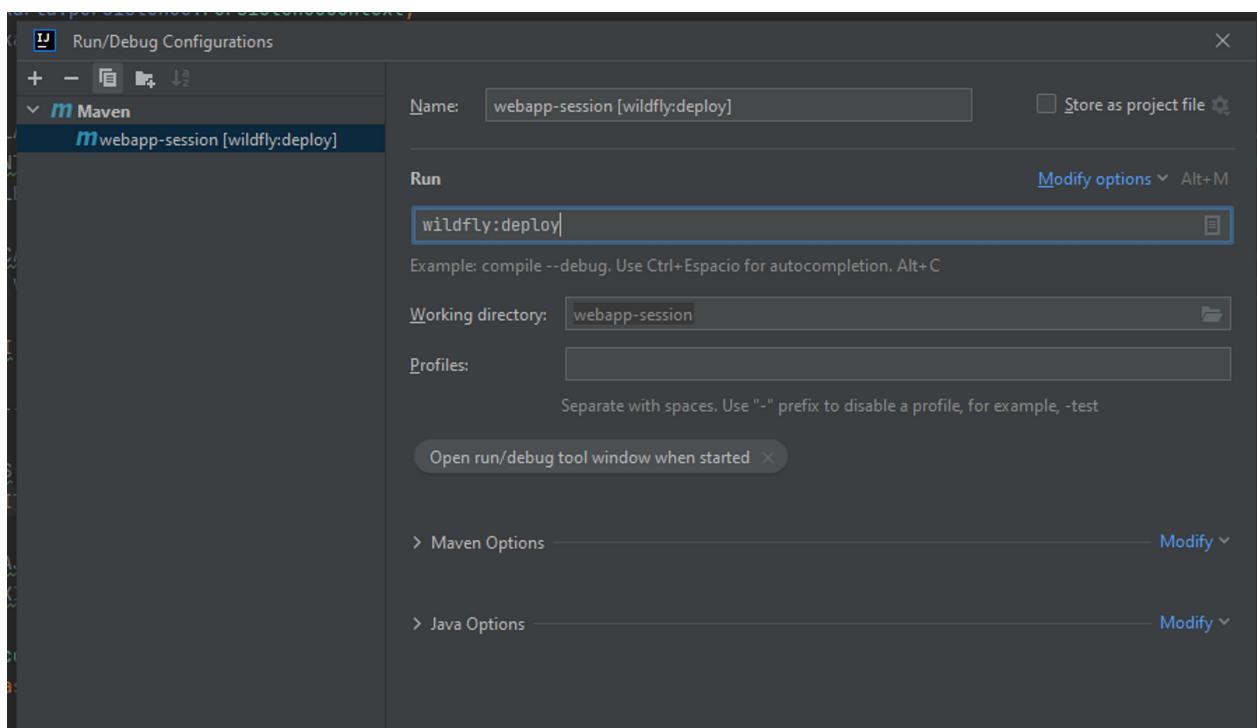
ProducerEntityManager.java

```

1 package org.aguzman.apiservlet.webapp.headers.configs;
2
3 import jakarta.enterprise.context.RequestScoped;
4 import jakarta.enterprise.inject.Produces;
5 import jakarta.persistence.EntityManager;
6 import jakarta.persistence.EntityManagerFactory;
7 import jakarta.persistence.PersistenceContext;
8 import jakarta.persistence.PersistenceUnit;
9
10 //550
11 //1 - EN LAS VERSIONES ANTERIORES DE ESTE PROYECTO, ESTA CON @ApplicationScoped Y SE CAMBIA A @RequestScoped PQ LE VAMOS A DECIR AL CONTENEDOR DE JPA, ..
12 //..EL CONTENEDOR DE PERSISTENCIA, QUE A SU VEZ ES MANEJADO POR EL CONTENEDOR DE WILDFLY, QUE NOS INYECTE UN ENTITY MANAGER,..
13 //..PERO LE DECIMOS QUE LO INYECTE DENTRO DEL REQUEST.
14
15 //2 - SE CAMBIA EL @PersistenceUnit POR EL @PersistenceContext : PQ ESTE MÉTODO YA NO VA A INYECTAR UN EntityManagerFactory, SINO QUE VA A INYECTAR UN EntityManager DIRECTAMENTE.
16 //.. A SU VEZ, EL EntityManager SE GUARDA EN EL PRODUCER QUE ES DEL TIPO DEL REQUEST. EN EL CONTEXTO DEL REQUEST. POR LO TANTO EL CICLO DE VIDA DE ESTE EntityManager ES EL REQUEST.
17
18 //3 - AQUÍ SE DEVUELVE EL ENTITY MANAGER, TAMBIÉN DEL REQUEST, PARA Q SE PUEDA INYECTAR DIRECTAMENTE.
19
20 //-----
21
22 //VENTAJAS :
23 // EL ENTITY MANAGER ES MANEJAD 100% POR EL CONTENEDOR. NO HAY Q PREOCUPARSE POR CREARLO O CERRAR LA CONEXIÓN.
24
25 //DESVENTAJA:
26 //LA CONEXIÓN SE CIERRA AUTOMATICAMENTE DESPUÉS DEL COMMIT.
27
28 @RequestScoped //@ApplicationScoped //1
29 public class ProducerEntityManager {
30
31     @PersistenceContext(name = "ejemploJpa") //@@PersistenceUnit(name = "ejemploJpa") //2
32     private EntityManager em; //private EntityManagerFactory emf;
33
34     @Produces
35     @RequestScoped
36     private EntityManager beanEntityManager() {
37         return em; //3
38     }
39 }

```

CONFIGURACIÓN DEL RUN (LO DE SIEMPRE) .



LEVANTA WILDFLY:

```
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> cd C:\developjava\wildfly-preview-26.1.3.Final\bin
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> ./standalone.bat
```

DALE RUN

| id | nombre |
|----|---------------|
| 1 | pc msi katana |
| 2 | teclado msi |
| 3 | lentes |

DESVENTAJA DE LA INYECCION DEL ENTITY MANAGER CON WILDFLY :

```
ERROR
panejado por el contenedor con @PersistenceContext
CONTEXTUAL /webapp-jpa-ejb2
Servlet Path: /listar.jsp
Path Info: null
Query String: null
Stack Trace:
org.apache.jasper.JasperException: jakarta.el.ELException: org.hibernate.LazyInitializationException: could not initialize proxy [org.aguzman.apiservlet.webapp.headers.models.entities.Categoría#1] - no Session
    at io.undertow.jsp@2.0.9.Final//org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:473)
    at io.undertow.jsp@2.0.9.Final//org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:403)
    at io.undertow.jsp@2.0.9.Final//org.apache.jasper.servlet.JspServlet.service(JspServlet.java:347)
    at jakarta.servlet.api@5.0.0//jakarta.servlet.http.HttpServlet.service(HttpServlet.java:587)
    at io.undertow.servlet@2.2.12.Final//io.undertow.servlet.handlers.ServletHandler.handleRequest(ServletHandler.java:74)
    at io.undertow.servlet@2.2.12.Final//io.undertow.servlet.handlers.FilterHandler.handleRequest(FilterHandler.java:81)
```

EN EL PROYECTO LA CONSULTA QUE DEVUELVE LA LISTA DESDE LA BBDD , ESTÁ HECHA CON FETCH JOIN, POR LO Q TRAE LA LISTA DE PRODUCTOS CON SUS CATEGORIAS EN UNA SOLA CONSULTA. EN LA CLASE ENTITY PRODUCTOS, EL ATRIBUTO DE LA CLASE "CATEGORIA" ESTA CON LAZY, POR LO TANTO AL USA FETCH JOIN, NO DA ERROR. PERO SI LE QUITAS EL FETCH JOIN EN LA CONSULTA Q TRAE LISTA, TE VA A TRAER EL ERROR DE "NO SESSION". YA Q DESPUÉS DE LA PRIMERA CONSULTA Q LISTA LOS PRODUTOS, SE CIERRA LA CONEXIÓN Y AL HACER LA CONSULTA POR LAS CATEGORIAS , YA NO HAY CONEXIÓN.

POR LO TANTO SI SE VA A USAR EL CONTENEDOR DE WILDFLY PARA INYECTAR EL ENTITY MANAGER, USA EL LEFT OUTER JOIN FETCH

```
Produced by the container with @PersistenceContext
CONTEXTUAL /webapp-jpa-ejb2
Servlet Path: /listar.jsp
Path Info: null
Query String: null
Stack Trace:
org.apache.jasper.JasperException: jakarta.el.ELException: org.hibernate.LazyInitializationException: could not initialize proxy [org.aguzman.apiservlet.webapp.headers.models.entities.Categoría#1] - no Session
    at io.undertow.jsp@2.0.9.Final//org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:473)
    at io.undertow.jsp@2.0.9.Final//org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:403)
    at io.undertow.jsp@2.0.9.Final//org.apache.jasper.servlet.JspServlet.service(JspServlet.java:347)
    at jakarta.servlet.api@5.0.0//jakarta.servlet.http.HttpServlet.service(HttpServlet.java:587)
    at io.undertow.servlet@2.2.12.Final//io.undertow.servlet.handlers.ServletHandler.handleRequest(ServletHandler.java:74)
    at io.undertow.servlet@2.2.12.Final//io.undertow.servlet.handlers.FilterHandler.handleRequest(FilterHandler.java:81)
```

```

Project ① CrudRepository
② ProductoRepositoryJdbcImpl
③ ProductoRepositoryJpaImpl
④ @RepositoryJdbc
⑤ @RepositoryJpa
⑥ UsuarioRepository
⑦ UsuarioRepositoryImpl
⑧ UsuarioRepositoryJpaImpl

@Project
⑨ services
⑩ resources
⑪ webapp
⑫ layout
⑬ META-INF
⑭ WEB-INF
⑮ carros.jsp

Run: m webapp-jpa-ejb2 [wildfly:deploy] ⑯ [INFO] BUILD SUCCESS
⑰ webapp 13 sec, 202 ms ⑱ [INFO] -----
```