

**CURSO JAVA**

# **EJERCICIO**

## **CREACIÓN POOL DE CONEXIONES CON GLASSFISH**



Experiencia y Conocimiento para tu vida

**CURSO JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# OBJETIVO DEL EJERCICIO

- El objetivo del ejercicio es crear un pool de conexiones con MySql en Glassfish. Al finalizar deberemos observar el siguiente resultado desde nuestro servidor de Glassfish.

GlassFish™ Server Open Source Edition

Common Tasks

- Domain
- server (Admin Server)
- Clusters
- Standalone Instances
- Nodes
- Applications
- Lifecycle Modules
- Monitoring Data
- Resources
  - Concurrent Resources
  - Connectors
  - JDBC
    - JDBC Resources
      - jdbc/PersonaDb
      - jdbc/\_\_TimerPool
      - jdbc/\_\_default
      - jdbc/bolsatrabajo
      - jdbc/sample

### JDBC Resources

JDBC resources provide applications with a means to connect to a database.

Resources (5)

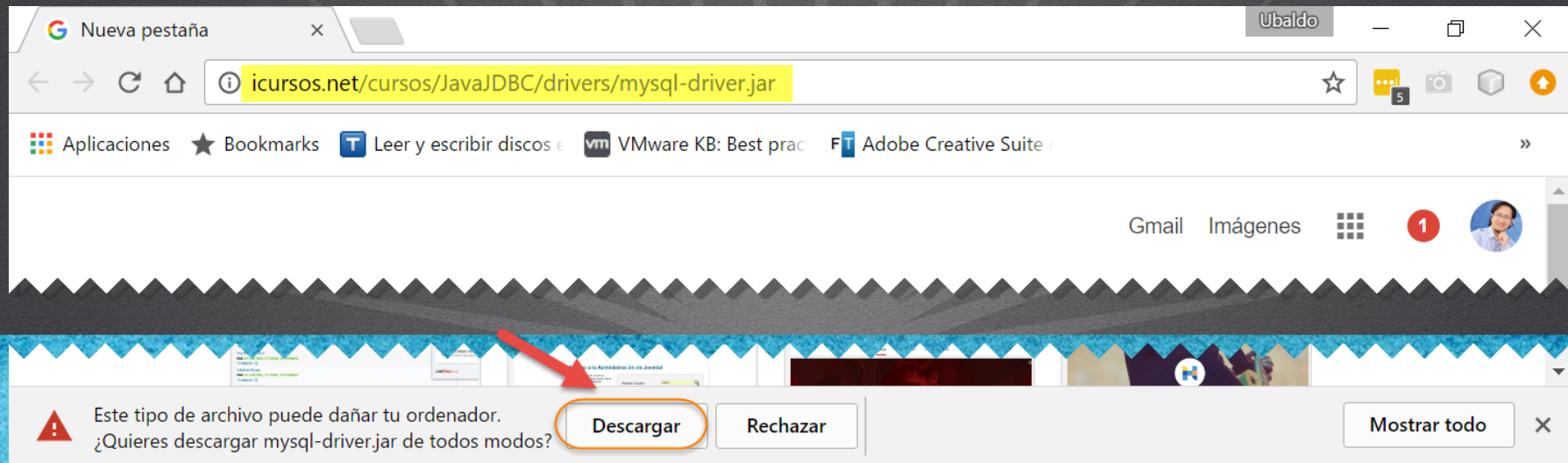
☒ ☐

Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Description
<input type="checkbox"/>	<a href="#">jdbc/PersonaDb</a>		✓	PersonaPool	
<input type="checkbox"/>	<a href="#">jdbc/__TimerPool</a>		✓	__TimerPool	
<input type="checkbox"/>	<a href="#">jdbc/__default</a>	java:comp/DefaultDataSource	✓	DerbyPool	
<input type="checkbox"/>	<a href="#">jdbc/bolsatrabajo</a>		✓	BolsaTrabajoPool	
<input type="checkbox"/>	<a href="#">jdbc/sample</a>		✓	SamplePool	

# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Configuramos la conexión de JTA en Glassfish. Agregamos el driver de mysql a Glassfish. Descargamos el .jar de mysql:

<http://icursos.net/cursos/JavaJDBC/drivers/mysql-driver.jar>

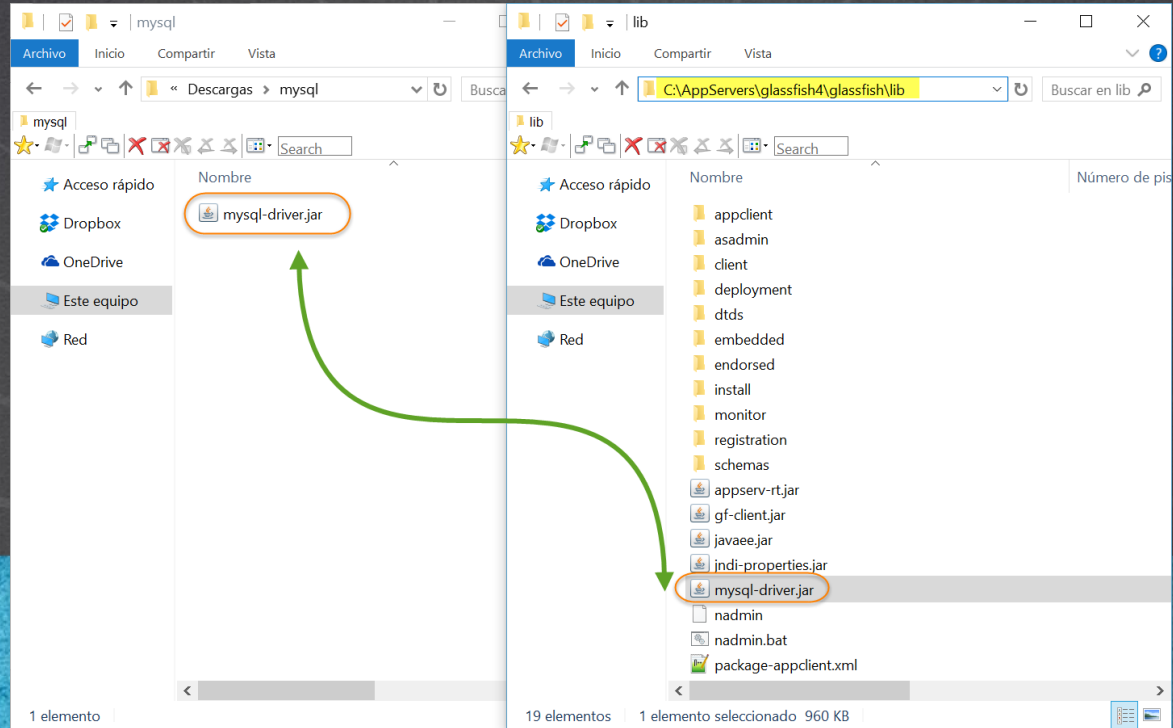




# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

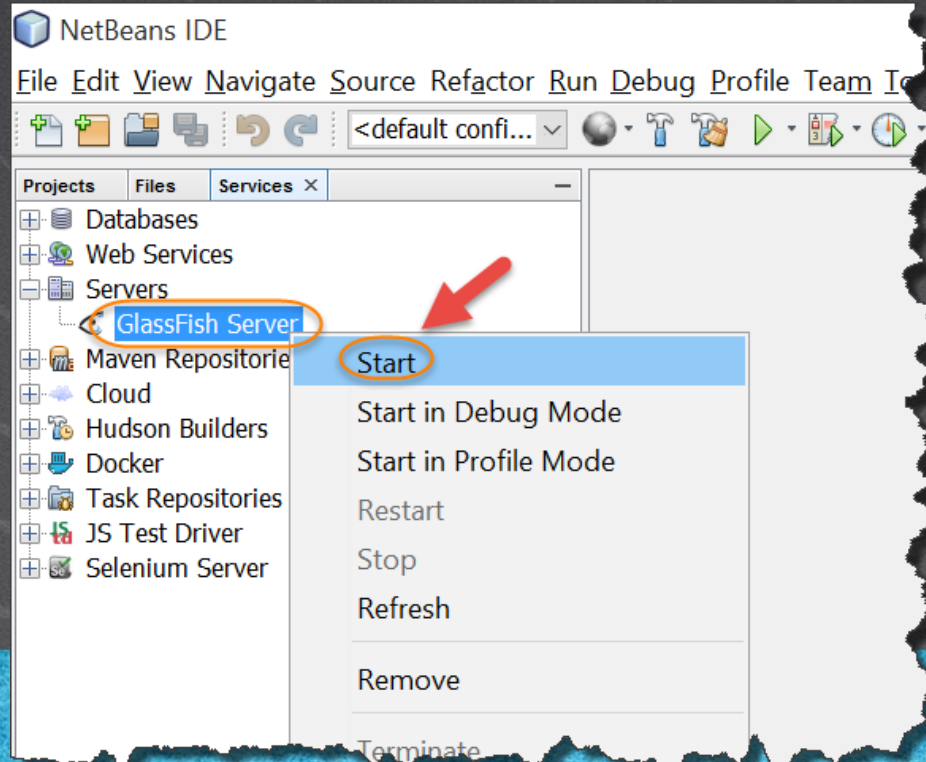
Copiamos el archivo recién descargado en la ruta de instalación de Glassfish. Ej. C:\AppServers\glassfish4\glassfish\lib

**Nota:** Si ya se tiene agregada la librería de MySQL se puede omitir este paso.



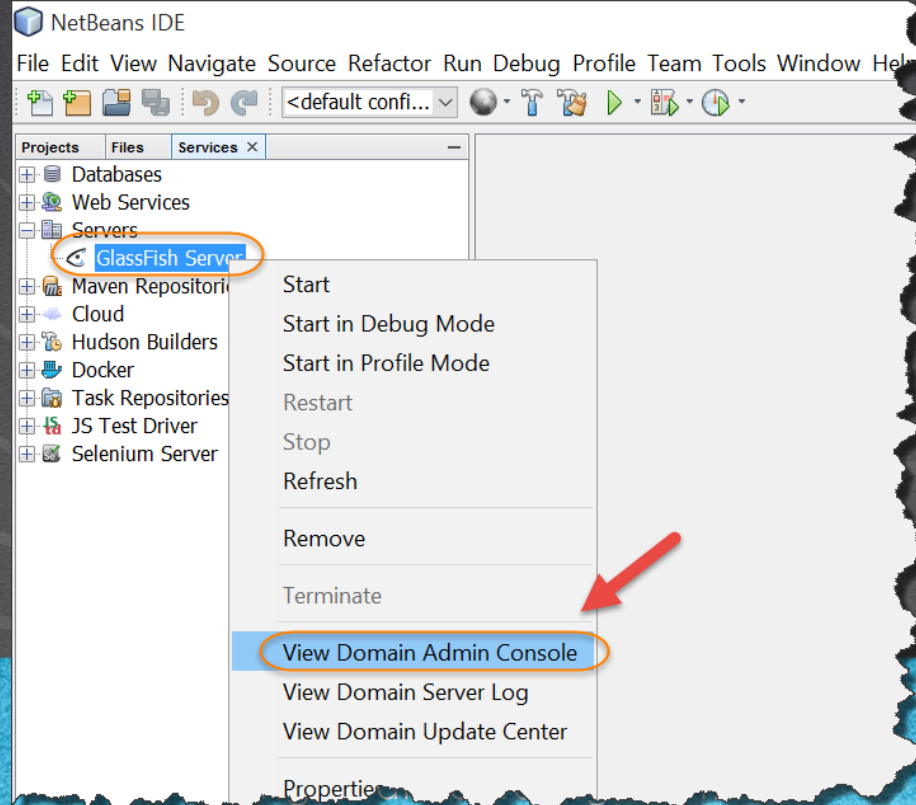
# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Abrimos Netbeans y levantamos el servidor de Glassfish:



# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Entramos a la consola de administración de Glassfish:





# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Creamos un nuevo pool de conexiones:

The screenshot shows the GlassFish Administration Console interface. The left-hand tree view displays the following structure:

- Server (Admin Server)
  - Clusters
  - Standalone Instances
  - Nodes
  - Applications
  - Lifecycle Modules
  - Monitoring Data
  - Resources** (1)
  - Concurrent Resources
  - Connectors
  - JDBC** (2)
  - JDBC Resources
    - JDBC Connection Pools** (3)
  - JMS Resources
  - JNDI

The main content area is titled "JDBC Connection Pools" and includes a description: "To store, organize, and retrieve data, most applications use relational databases. Java EE applications access relational databases through the JDBC API. Before an application can access a database, it must get a connection." Below the description, there is a section for "Pools (3)" with a "New..." button (4) and a "Delete" button. A table lists the existing pools:

Select	Pool Name	Resource Type	Classname	Description
<input type="checkbox"/>	DerbyPool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource	
<input type="checkbox"/>	SamplePool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource	
<input type="checkbox"/>	__TimerPool	javax.sql.XADataSource	org.apache.derby.jdbc.EmbeddedXADataSource	

# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Proporcionamos los siguientes datos para crear el pool de conexiones llamado BolsaTrabajoPool:

**New JDBC Connection Pool (Step 1 of 2)**  
Identify the general settings for the connection pool.

**General Settings**

1 **Pool Name:** \* BolsaTrabajoPool

2 **Resource Type:** javax.sql.ConnectionPoolDataSource  
Must be specified if the datasource class implements more than 1 of the interface.

3 **Database Driver Vendor:** MySql  
Select or enter a database driver vendor

**Introspect:** ☐ **Enabled**  
If enabled, data source or driver implementation class names will enable introspection.

4 **Next** **Cancel**  
\* Indicates required field

**CURSO JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Dejamos los valores por default:

**New JDBC Connection Pool (Step 2 of 2)**

PreviousFinishCancel

Identify the general settings for the connection pool. Datasource Classname or Driver Classname must be specified for the connection pool.

\* Indicates required field

### General Settings

Pool Name:	BolsaTrabajoPool
Resource Type:	javax.sql.ConnectionPoolDataSource
Database Driver Vendor:	MySql
Datasource Classname:	<div>com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource</div> <div>Select or enter vendor-specific classname that implements the DataSource and/or XADataSource APIs</div>
Driver Classname:	<div></div> <div>Select or enter vendor-specific classname that implements the java.sql.Driver interface.</div>
Ping:	<input checked="" type="checkbox"/> <b>Enabled</b> When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes
Description:	

**CURSO JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Dejamos los valores por default:

## Pool Settings

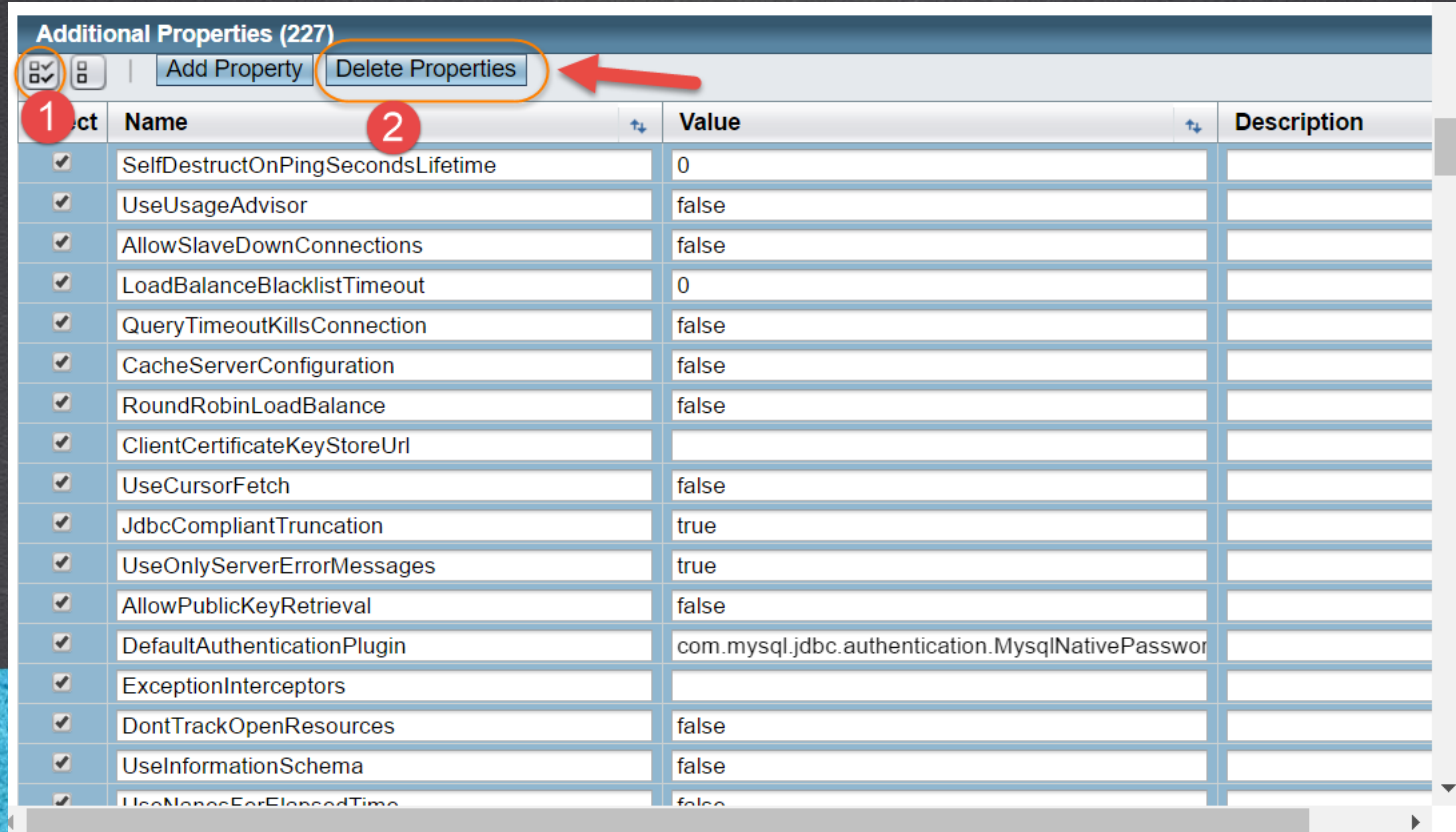
- Initial and Minimum Pool Size:**  Connections  
Minimum and initial number of connections maintained in the pool
- Maximum Pool Size:**  Connections  
Maximum number of connections that can be created to satisfy client requests
- Pool Resize Quantity:**  Connections  
Number of connections to be removed when pool idle timeout expires
- Idle Timeout:**  Seconds  
Maximum time that connection can remain idle in the pool
- Max Wait Time:**  Milliseconds  
Amount of time caller waits before connection timeout is sent

## Transaction

- Non Transactional Connections:** ☐ **Enabled**  
Returns non-transactional connections
- Transaction Isolation:**   
If unspecified, use default level for JDBC Driver
- Isolation Level:** ☒ **Guaranteed**  
All connections use same isolation level; requires Transaction Isolation

# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Borramos todas las propiedades:



**Additional Properties (227)**

1 ☒ ☐ | Add Property Delete Properties 2

Select	Name	Value	Description
<input checked="" type="checkbox"/>	SelfDestructOnPingSecondsLifetime	0	
<input checked="" type="checkbox"/>	UseUsageAdvisor	false	
<input checked="" type="checkbox"/>	AllowSlaveDownConnections	false	
<input checked="" type="checkbox"/>	LoadBalanceBlacklistTimeout	0	
<input checked="" type="checkbox"/>	QueryTimeoutKillsConnection	false	
<input checked="" type="checkbox"/>	CacheServerConfiguration	false	
<input checked="" type="checkbox"/>	RoundRobinLoadBalance	false	
<input checked="" type="checkbox"/>	ClientCertificateKeyStoreUrl		
<input checked="" type="checkbox"/>	UseCursorFetch	false	
<input checked="" type="checkbox"/>	JdbcCompliantTruncation	true	
<input checked="" type="checkbox"/>	UseOnlyServerErrorMessages	true	
<input checked="" type="checkbox"/>	AllowPublicKeyRetrieval	false	
<input checked="" type="checkbox"/>	DefaultAuthenticationPlugin	com.mysql.jdbc.authentication.MysqlNativePasswor	
<input checked="" type="checkbox"/>	ExceptionInterceptors		
<input checked="" type="checkbox"/>	DontTrackOpenResources	false	
<input checked="" type="checkbox"/>	UseInformationSchema	false	
<input checked="" type="checkbox"/>	UseNonzeroForElapsedTime	false	



# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Agregamos las siguientes propiedades:

portNumber	3306
databaseName	bolsatrabajo
datasourceName	com.mysql.jdbc.Driver
serverName	localhost
user	root
password	admin

El usuario y password pueden cambiar según la instalación de MySql. El serverName puede ser una IP del servidor MySql.

**CURSO JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Agregamos las siguientes propiedades, proporcionando los valores mostrados y después damos click en Finish. Los valores pueden variar según la base de datos y demás valores que utilicemos para conectarnos a MySql:

**Transaction**

**Non Transactional Connections:** ☐ Enabled  
Returns non-transactional connections

**Transaction Isolation:**   
If unspecified, use default level for JDBC Driver

**Isolation Level:** ☒ Guaranteed  
All connections use s

**Additional Properties (6)**

☒ ☐

Select	Name	Value	Description
<input type="checkbox"/>	portNumber	3306	
<input type="checkbox"/>	databaseName	bolsatrabajo	
<input type="checkbox"/>	datasourceName	com.mysql.jdbc.Driver	
<input type="checkbox"/>	serverName	localhost	
<input type="checkbox"/>	user	root	
<input type="checkbox"/>	password	admin	

**Ir agregando cada una de las propiedades**



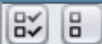
# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Verificamos la conexión del nuevo pool desde Glassfish:

## JDBC Connection Pools

To store, organize, and retrieve data, most applications use relational databases. Java EE applications access relational databases through JTA. To access a database, an application can access a database, it must get a connection.

### Pools (5)



New...

Delete

Select	Pool Name	Resource Type	Classname
<input type="checkbox"/>	BolsaTrabajoPool	javax.sql.ConnectionPoolDataSource	com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource
<input type="checkbox"/>	DerbyPool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource
<input type="checkbox"/>	PersonaPool	javax.sql.ConnectionPoolDataSource	com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource
<input type="checkbox"/>	SamplePool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource
<input type="checkbox"/>	__TimerPool	javax.sql.XADataSource	org.apache.derby.jdbc.EmbeddedXADataSource

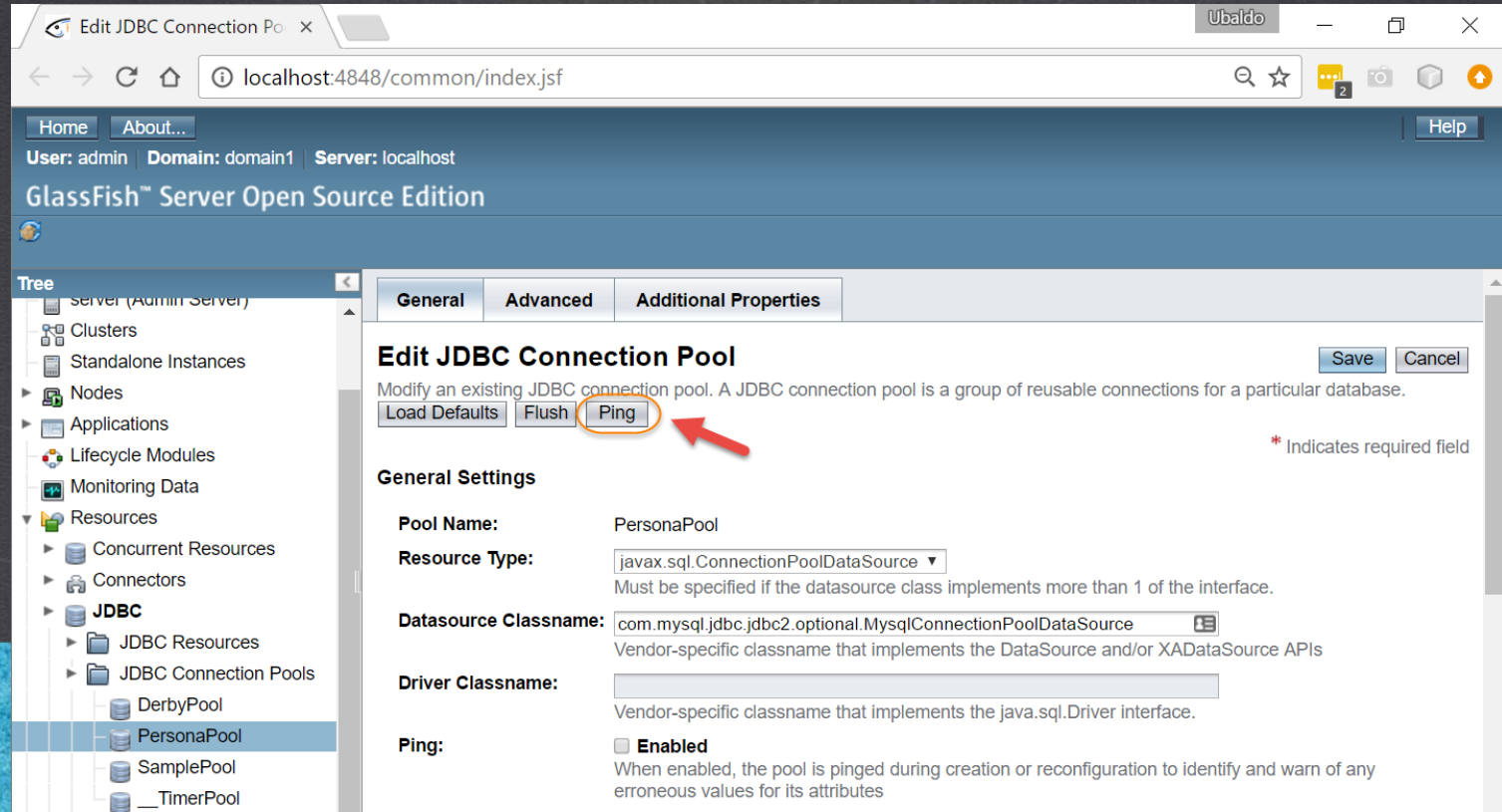
**CURSO JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Verificamos la conexión del nuevo pool desde Glassfish:



The screenshot shows the GlassFish Admin Console interface. The browser address bar displays `localhost:4848/common/index.jsf`. The page title is "GlassFish™ Server Open Source Edition". The left sidebar shows a tree view of the server configuration, with "JDBC" expanded and "PersonaPool" selected. The main content area is titled "Edit JDBC Connection Pool" and has three tabs: "General", "Advanced", and "Additional Properties". The "General" tab is active. Below the tabs, there are buttons for "Load Defaults", "Flush", and "Ping". A red arrow points to the "Ping" button. The "General Settings" section contains the following fields:

- Pool Name:** PersonaPool
- Resource Type:** `javax.sql.ConnectionPoolDataSource` (dropdown menu)
- Datasource Classname:** `com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource` (text field)
- Driver Classname:** (empty text field)
- Ping:** ☐ Enabled

Below the "Ping" field, there is a description: "When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes".

# PASO 1. CONFIGURACIÓN CONEXIÓN JTA

Verificamos la conexión del nuevo pool desde Glassfish. Si hace ping la conexión ha sido exitosa. De lo contrario revisamos que los valores sean correctos (en el tabulador de Additional Properties se puede corregir)

GlassFish™ Server Open Source Edition

Common Tasks

- Domain
  - server (Admin Server)
  - Clusters
  - Standalone Instances
  - Nodes
  - Applications
  - Lifecycle Modules
  - Monitoring Data
  - Resources
    - Concurrent Resources
    - Connectors
    - JDBC
      - JDBC Resources
      - JDBC Connection Pools
        - BolsaTrabajoPool
        - DerbyPool
        - PersonaPool
        - SamplePool

General Advanced Additional Properties

**Ping Succeeded**

### Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

[Load Defaults](#) [Flush](#) [Ping](#)

#### General Settings

**Pool Name:** BolsaTrabajoPool

**Resource Type:**    
Must be specified if the datasource class implements more than 1 of the interface.

**Datasource Classname:**    
Vendor-specific classname that implements the DataSource and/or XADataSource APIs

**Driver Classname:**    
Vendor-specific classname that implements the java.sql.Driver interface.

**Ping:** ☒ **Enabled**   
When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any errors.

# PASO 2. CONFIGURACIÓN CONEXIÓN JTA

Creamos ahora el recurso de JDBC:

The screenshot shows the GlassFish Server Open Source Edition administration console. The browser address bar displays `localhost:4848/common/index.jsf`. The page title is "JDBC Resources". Below the title, there is a description: "JDBC resources provide applications with a means to connect to a database." The left sidebar shows the "Tree" view with the following structure:

- Server (Admin Server)
  - Clusters
  - Standalone Instances
  - Nodes
  - Applications
  - Lifecycle Modules
  - Monitoring Data
  - Resources
    - Concurrent Resources
    - Connectors
    - JDBC** (1)
    - JDBC Resources (2)
      - jdbc/\_\_TimerPool
      - jdbc/\_\_default
      - jdbc/sample

The main content area shows a table of existing JDBC resources. The table has the following columns: Select, JNDI Name, Logical JNDI Name, Enabled, Connection Pool, and Description. The table contains three rows:

Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Description
<input type="checkbox"/>	jdbc/__TimerPool		✓	__TimerPool	
<input type="checkbox"/>	jdbc/__default	java:comp/DefaultDataSource	✓	DerbyPool	
<input type="checkbox"/>	jdbc/sample		✓	SamplePool	

A red arrow points to the "New..." button in the toolbar above the table. The toolbar also includes "Delete", "Enable", and "Disable" buttons.



# PASO 2. CONFIGURACIÓN CONEXIÓN JTA

Creamos ahora el recurso de **jdbc/bolsatrabajo**. Este nombre puede ser cualquiera, sin embargo es el que utilizaremos vía JTA desde nuestras aplicaciones Java:

GlassFish™ Server Open Source Edition

Common Tasks

- Domain
  - server (Admin Server)
    - Clusters
    - Standalone Instances
    - Nodes
    - Applications
    - Lifecycle Modules
    - Monitoring Data
    - Resources
      - Concurrent Resources
      - Connectors
      - JDBC
        - JDBC Resources
          - jdbc/PersonaDb
          - jdbc/\_\_\_TimerPool
          - jdbc/\_\_\_default

### New JDBC Resource

Specify a unique JNDI name that identifies the JDBC resource you want to create. The name must contain only alphanumeric, underscore, dash, and dot characters.

1 JNDI Name: \* jdbc/bolsatrabajo

2 Pool Name: BolsaTrabajoPool ▼

Use the [JDBC Connection Pools](#) page to create new pools

Description:

Status: ☒ Enabled

#### Additional Properties (0)

[Add Property](#) [Delete Properties](#)

Select	Name	Value	Description
No items found.			

3 OK Cancel

# PASO 2. CONFIGURACIÓN CONEXIÓN JTA

Con esto ya tenemos la conexión de JDBC llamada **jdbc/bolsatrabajo** y el pool de conexiones de MySql. Ya podemos utilizarlo para conectarnos desde nuestra aplicación de Java vía JTA (Java Transaction API).

GlassFish™ Server Open Source Edition

## JDBC Resources

JDBC resources provide applications with a means to connect to a database.

Resources (5)

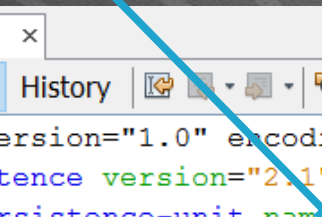
☒ ☐

Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Description
<input type="checkbox"/>	jdbc/PersonaDb		✓	PersonaPool	
<input type="checkbox"/>	jdbc/__TimerPool		✓	__TimerPool	
<input type="checkbox"/>	jdbc/__default	java:comp/DefaultDataSource	✓	DerbyPool	
<input checked="" type="checkbox"/>	jdbc/bolsatrabajo		✓	BolsaTrabajoPool	
<input type="checkbox"/>	jdbc/sample		✓	SamplePool	

# PASO 2. CONFIGURACIÓN CONEXIÓN JTA

Ejemplo de Uso: Podemos observar que el mismo nombre configurado en Glassfish, es el nombre usado en el archivo persistence.xml:

Resources (5)						
<div><input checked="" type="checkbox"/> <input type="checkbox"/>   <input type="button" value="New..."/> <input type="button" value="Delete"/> <input type="button" value="Enable"/> <input type="button" value="Disable"/></div>						
Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Desc	
<input type="checkbox"/>	jdbc/PersonaDb		✓	PersonaPool		
<input type="checkbox"/>	jdbc/__TimerPool		✓	__TimerPool		
<input type="checkbox"/>	jdbc/_default	java:comp/DefaultDataSource	✓	DerbyPool		
<input type="checkbox"/>	jdbc/bolsatrabajo		✓	BolsaTrabajoPool		
<input type="checkbox"/>	jdbc/sample		✓	SamplePool		



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
3   <persistence-unit name="BolsaTrabajoPU" transaction-type="JTA">
4     <jta-data-source>jdbc/bolsatrabajo</jta-data-source>
5   </persistence-unit>
6 </persistence>
7
```



# CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos creado un pool de conexiones y creamos un recurso jdbc que será el que podamos utilizar desde nuestras aplicaciones Java, por ejemplo con JPA en su archivo persistence.xml
- De esta manera podemos comunicarnos con la base de datos de MySQL, además de acceder vía JTA (Java Transaction API), el cual nos permite delegar los datos de conexión a Glassfish y así evitar configurar la conexión a base de datos desde nuestra aplicación.



Experiencia y Conocimiento para tu vida

**CURSO JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

**CURSO ONLINE**

# **CURSO DE JAVA**

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

**CURSO JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)