Introducción

- En muchas ocasiones se requiere que una aplicación desplegada en un servidor de aplicaciones pueda acceder a los datos de una base de datos.
- Para ello, se debe utilizar el conector correspondiente entre Java y la base de datos.
  - El conector es dependiente de la base de datos que se utilice.

Conexión MySQL - Glassfish (I)

- En primer lugar vamos a realizar una práctica con la base de datos MySQL.
  - Crearemos inicialmente una base de datos y un usuario, ambos con nombre curso:

```
CREATE DATABASE curso;
CREATE USER curso IDENTIFIED BY 'curso-araw';
GRANT ALL PRIVILEGES ON curso.* TO curso;
```

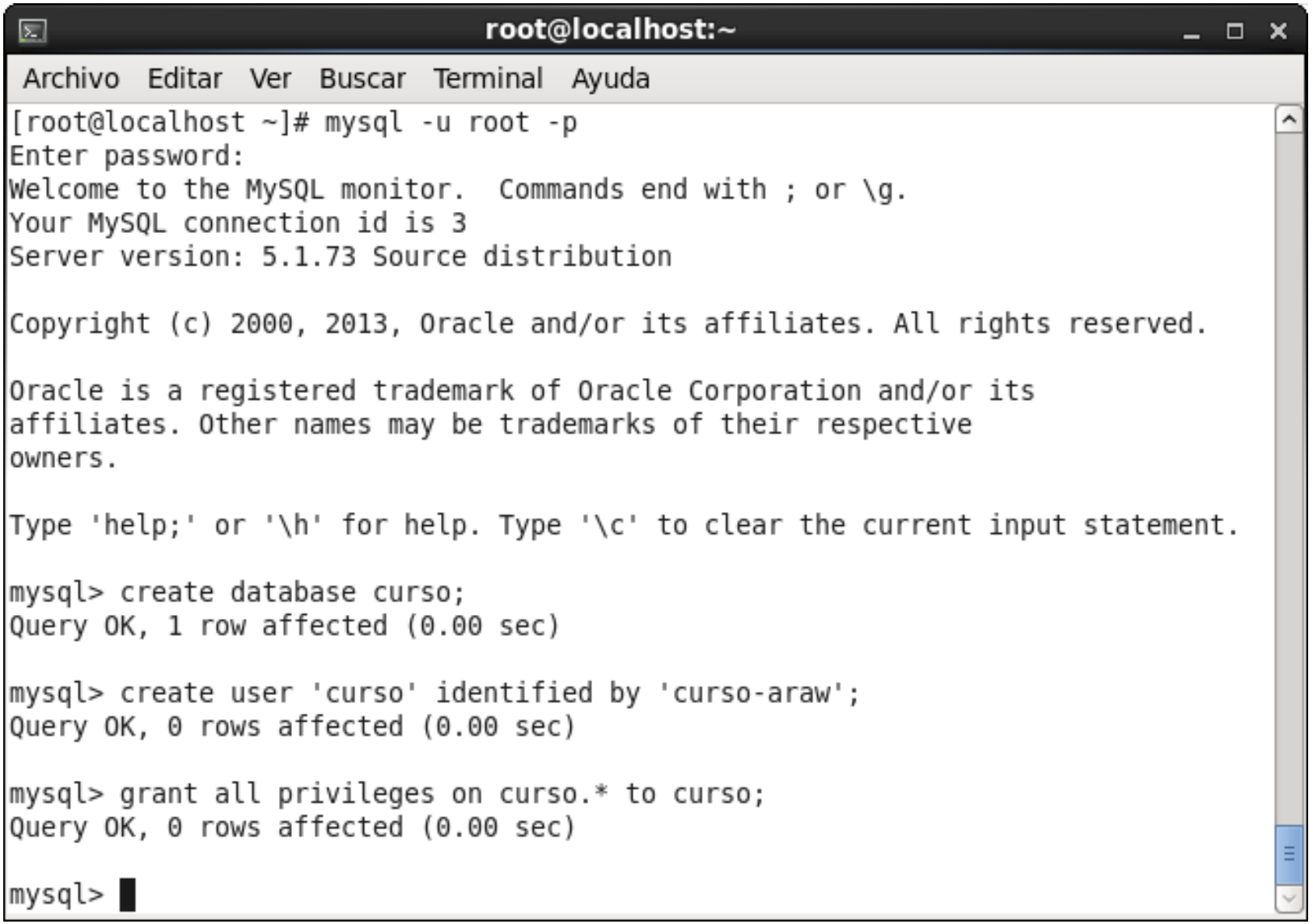
  - Debemos descargar el conector de Java para la base de datos desde la URL:

```
http://dev.mysql.com/downloads/connector/j/
```

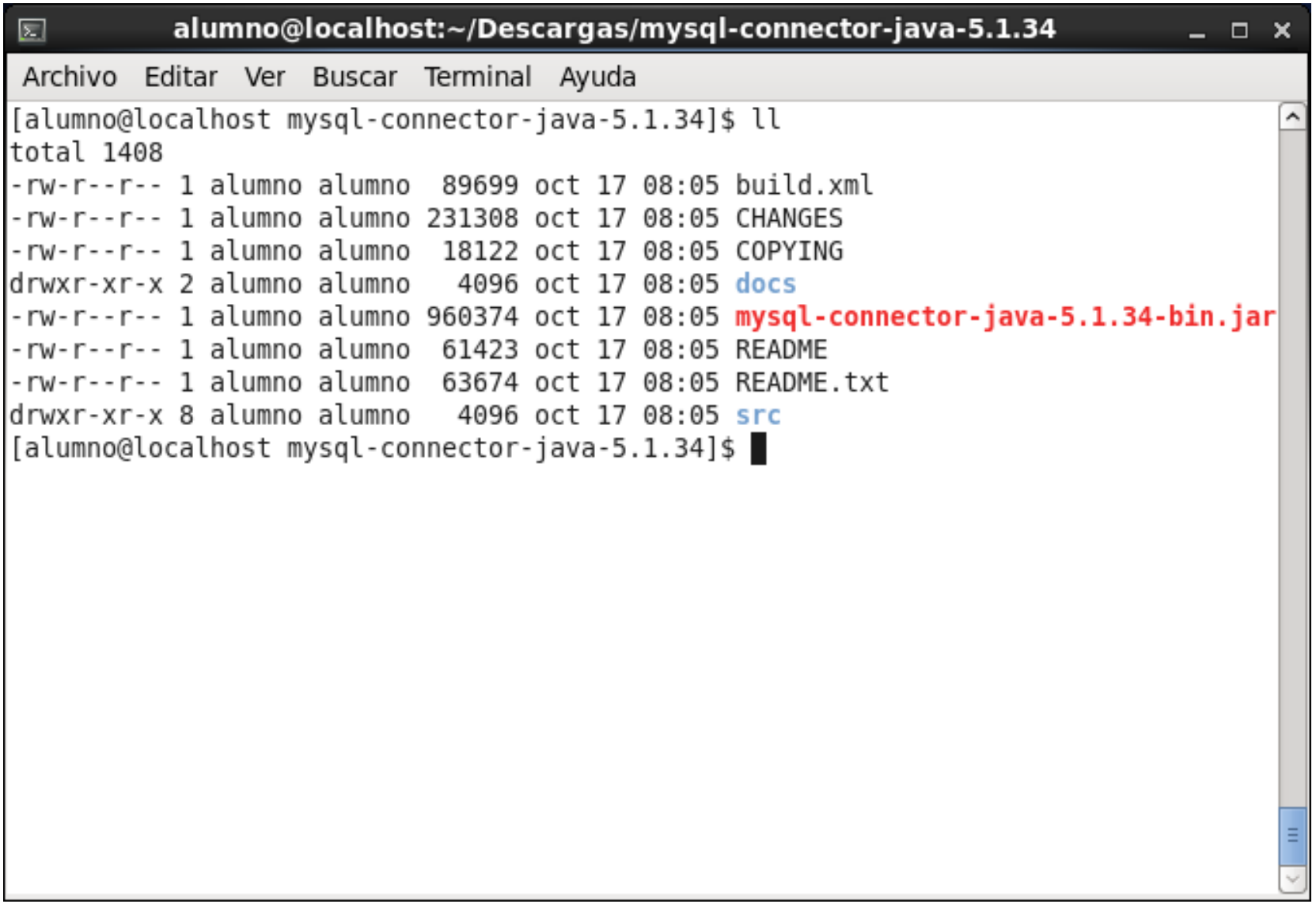
  - Descomprimir el conector para obtener el fichero **mysql-connector-java-<version>-bin.jar**.

```
unzip mysql-connector-java-<version>.zip
```

Conexión MySQL - Glassfish (II)

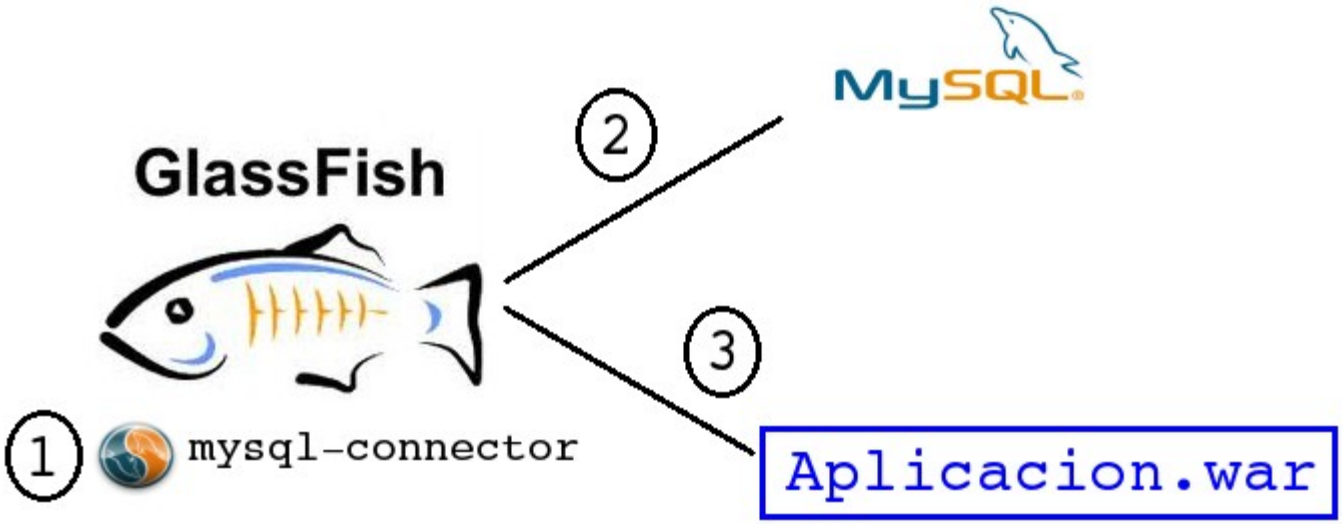


Conexión MySQL - Glassfish (III)



Conexión MySQL – Glassfish (IV)

- Para que una aplicación desplegada en Glassfish acceda a una base de datos MySQL, en la parte del servidor hay que seguir 3 pasos:

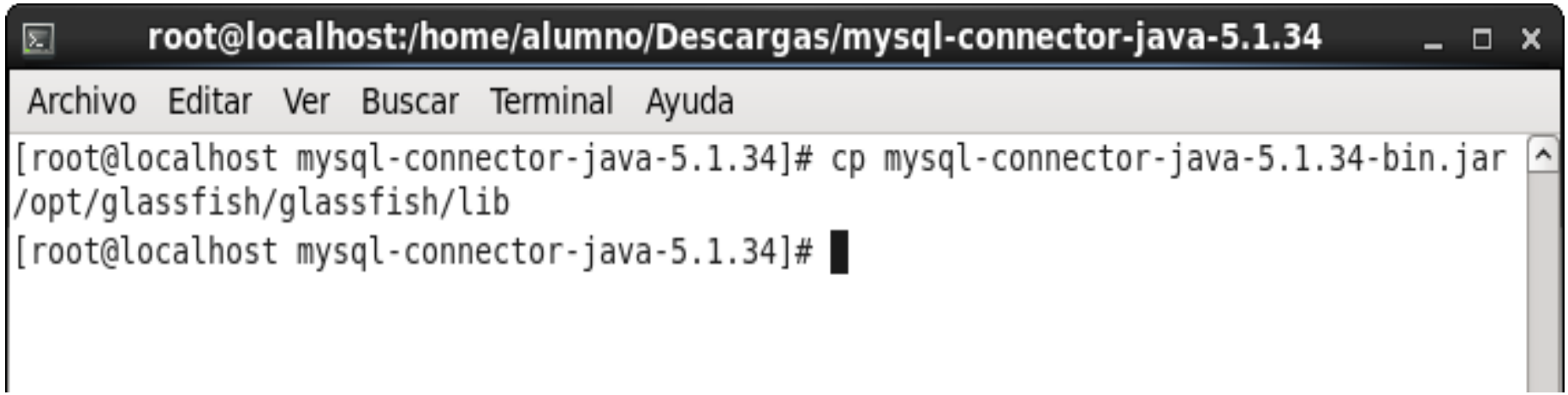


1. Instalar el conector MySQL
2. Crear un Connection Pool
3. Crear un recurso JDBC

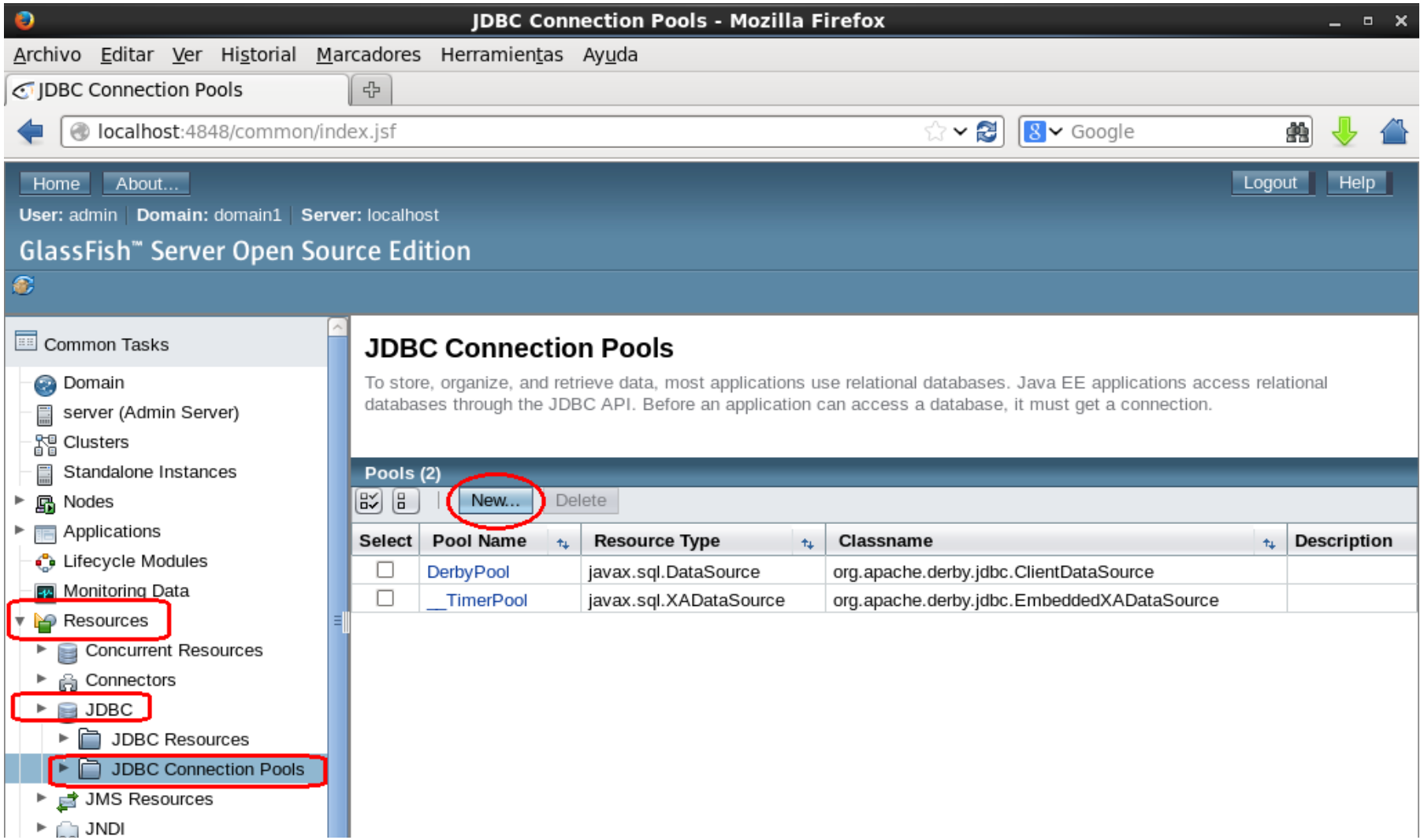
Instalar el conector de MySQL

- Copiar el fichero **mysql-connector-java-<version>-bin.jar** en la carpeta en que Glassfish almacena las librerías.

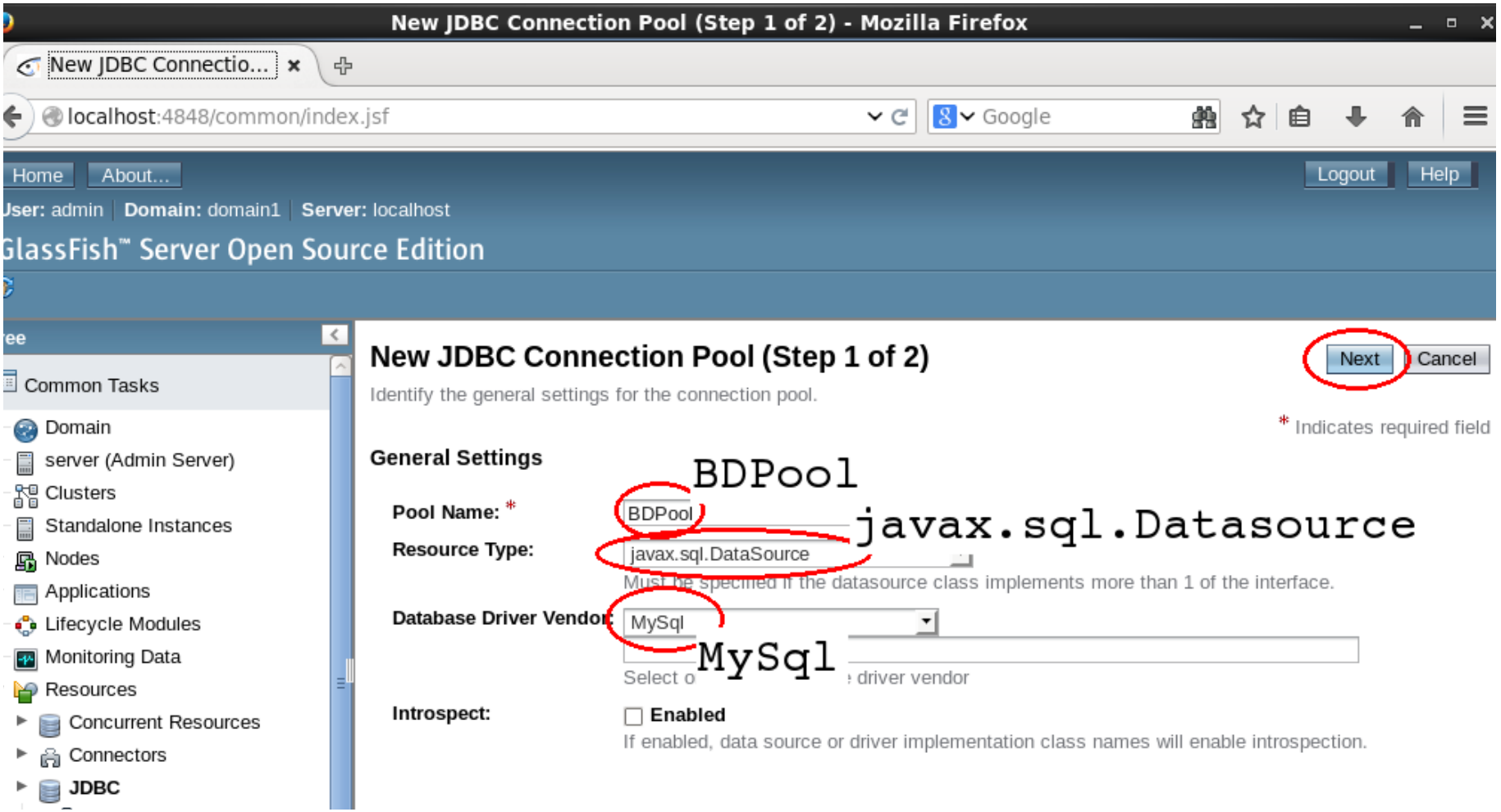
```
cp mysql-connector-java-5.1.34-bin.jar
/opt/glassfish/glassfish/lib/
```



Crear un Connection Pool a MySQL (I)



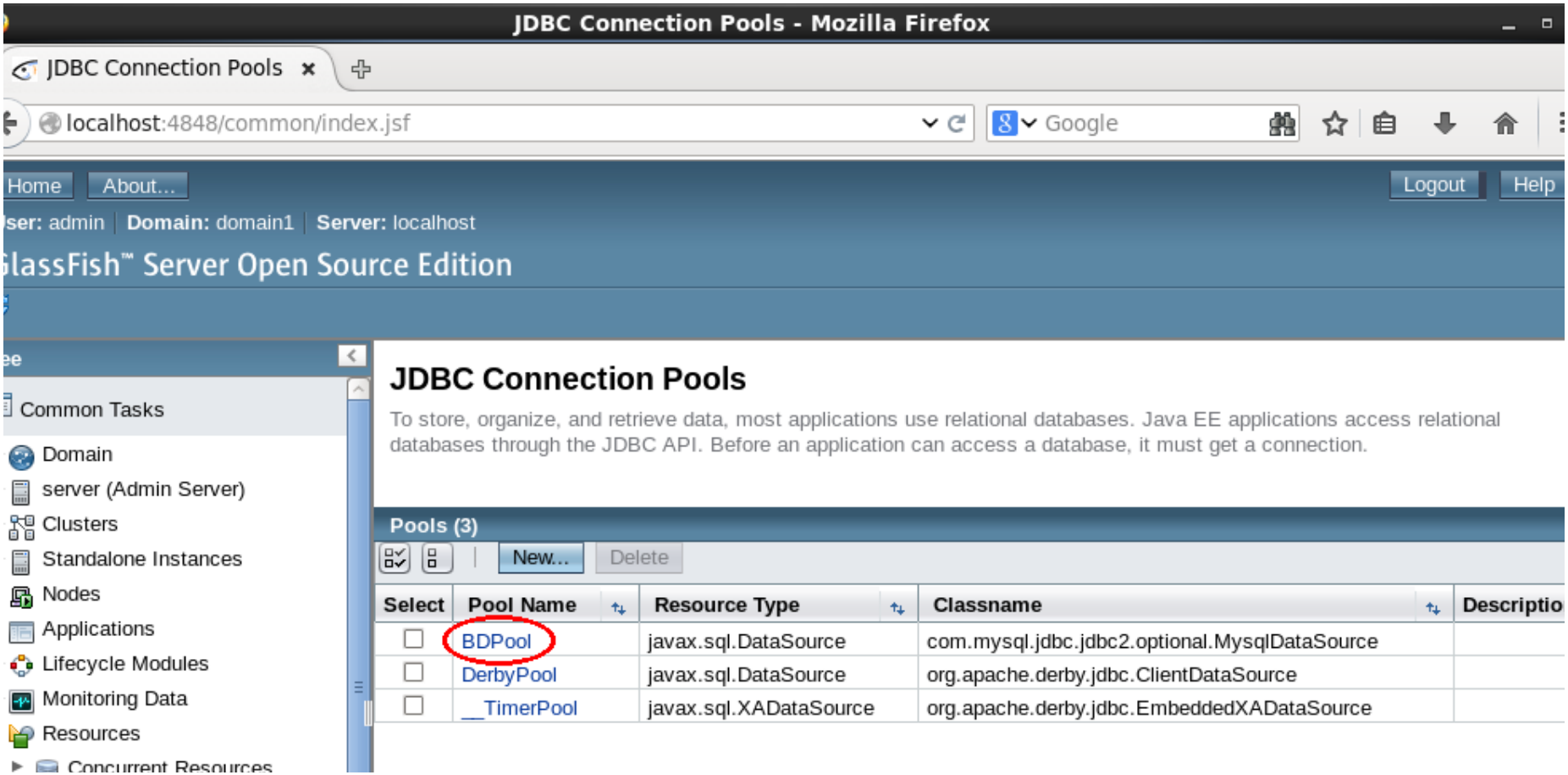
Crear un Connection Pool a MySQL (II)



Crear un Connection Pool a MySQL (III)

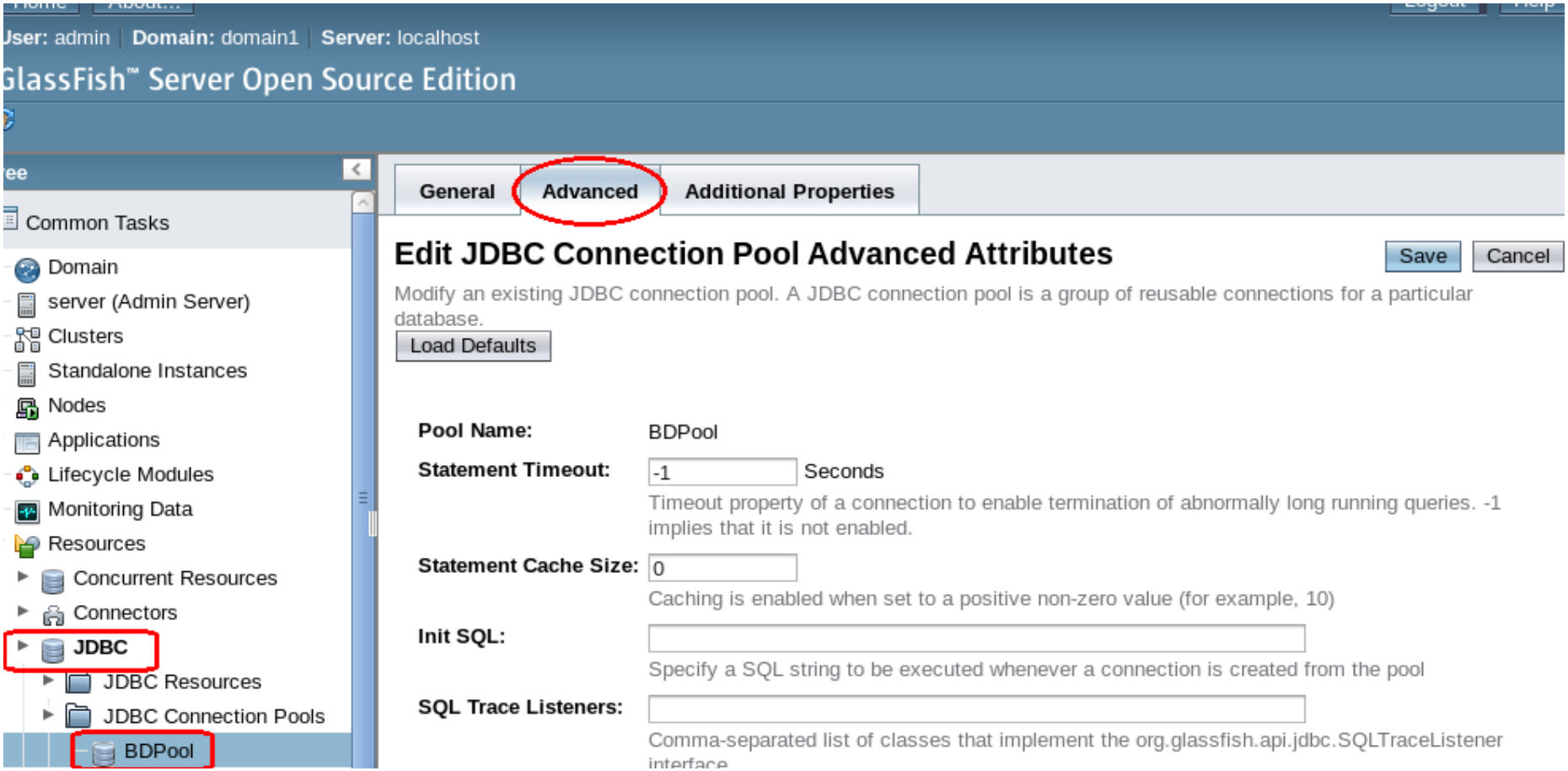
- Dejar por defecto los valores de “General Settings”.
- Dejar por defecto los valores de “Pool Settings”.
- Añadir en “Additional Properties” los siguientes valores:
  - DatabaseName: curso
  - ServerName: localhost
  - User: curso
  - Password: curso-araw
  - Url y URL: jdbc:mysql://localhost:3306/curso
- Pulsar en “Finish”.

Crear un Connection Pool a MySQL (IV)

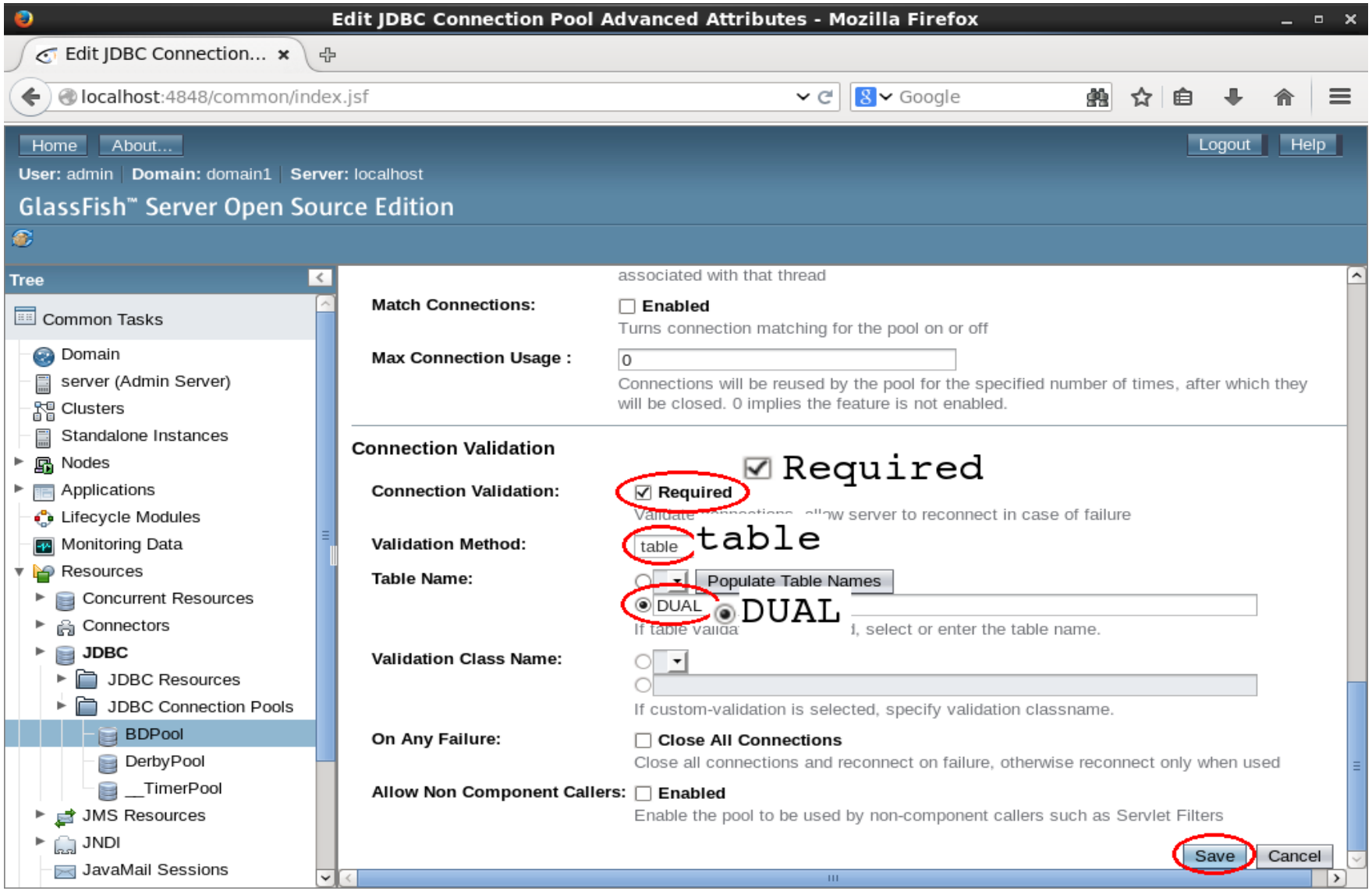


Crear un Connection Pool a MySQL (V)

- En opciones Avanzadas, editar los valores de la sección “Connection Validation”.

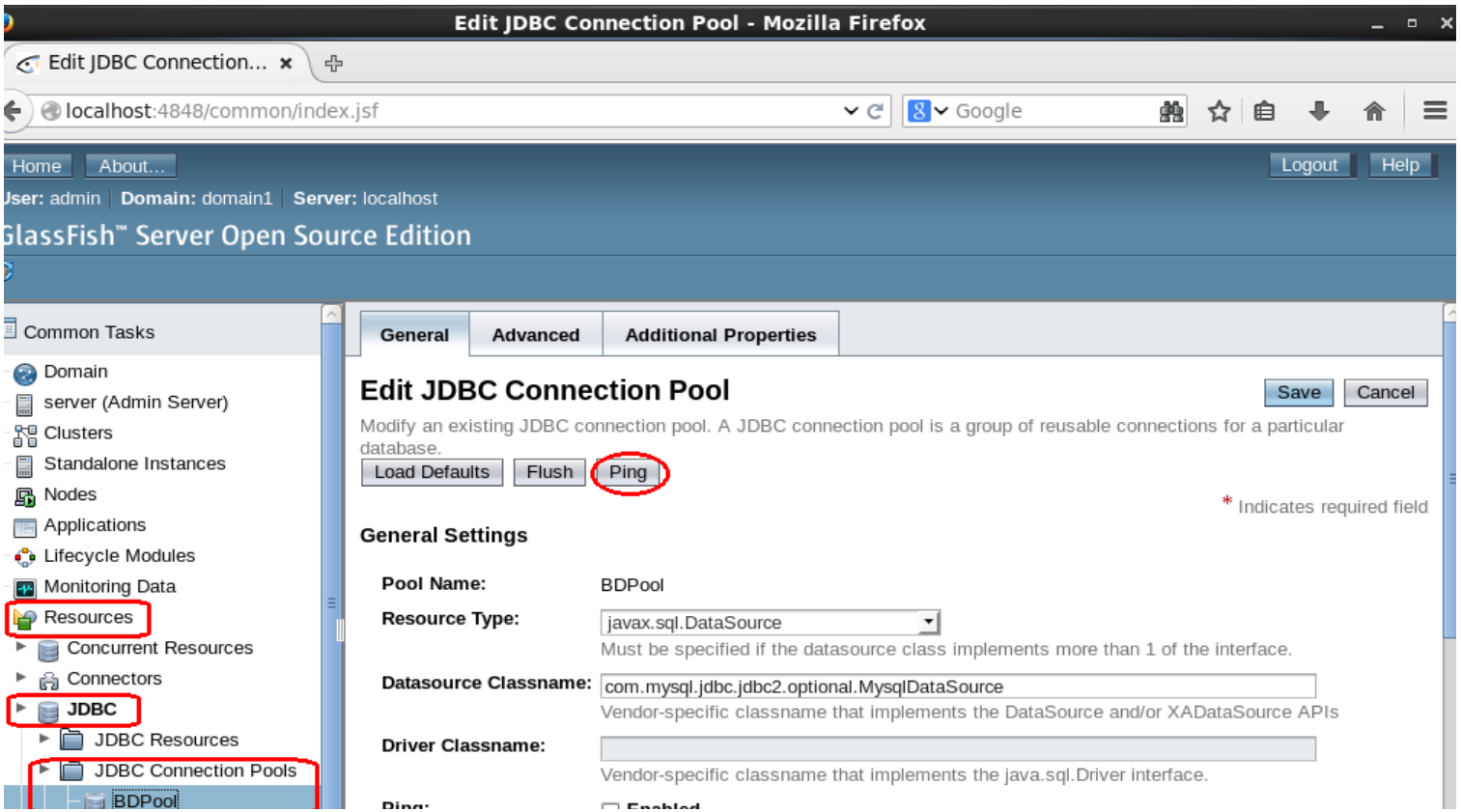


Crear un Connection Pool a MySQL (VI)

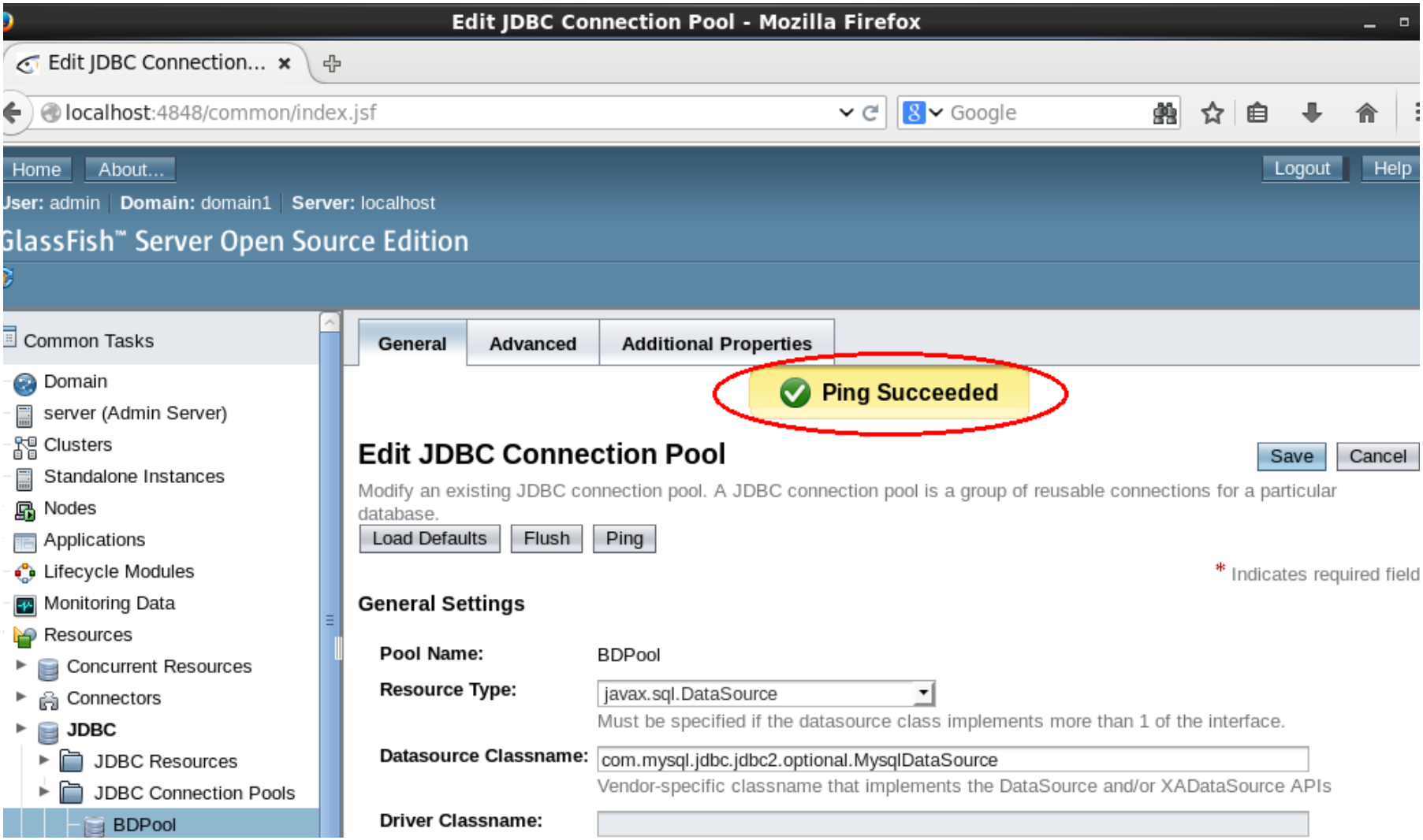


Crear un Connection Pool a MySQL (VII)

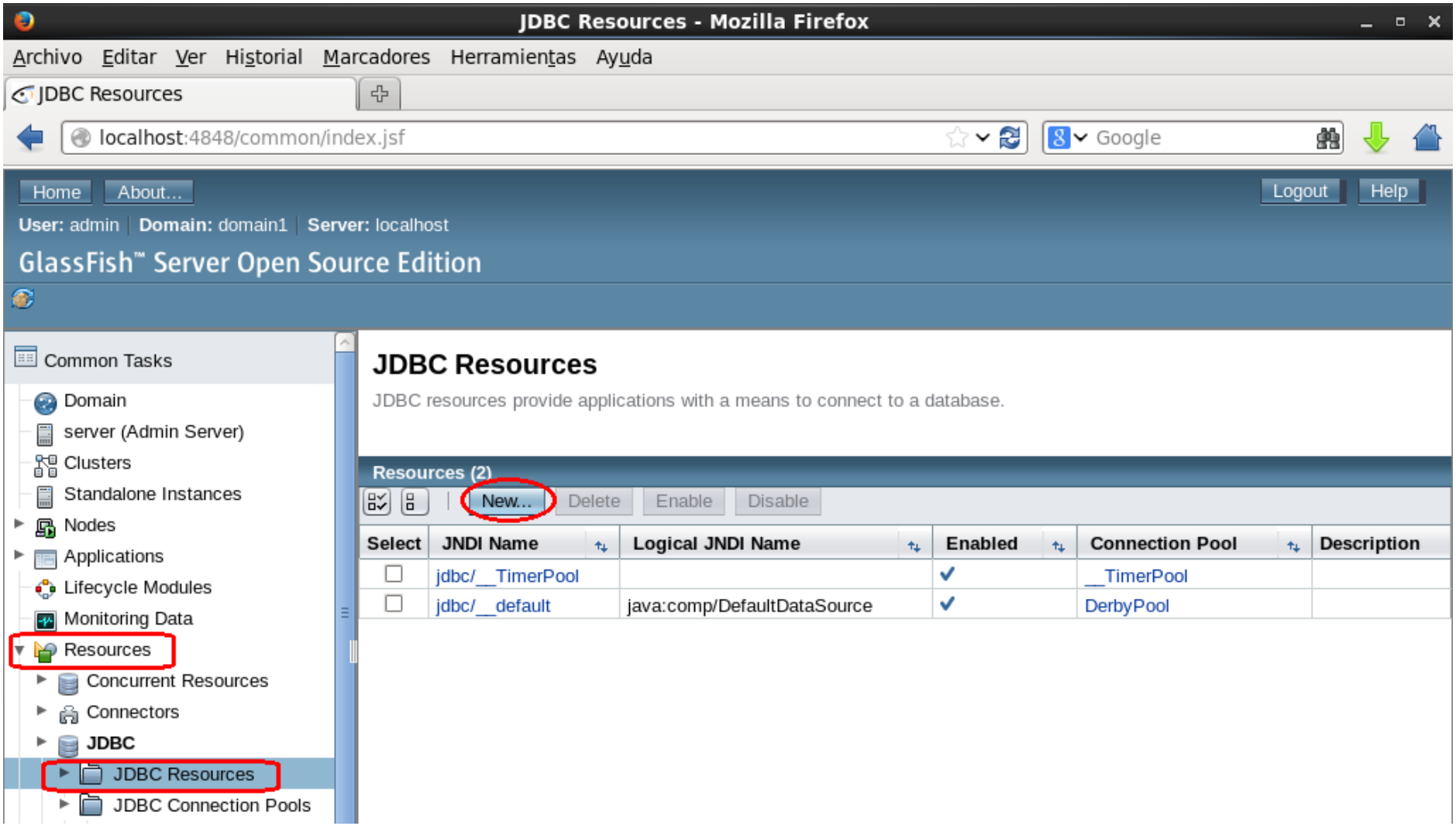
- Para comprobar que funciona el Connection Pool, se puede hacer un ping (es necesario reiniciar el GlassFisch y el servidor de MySQL debe estar arrancado).



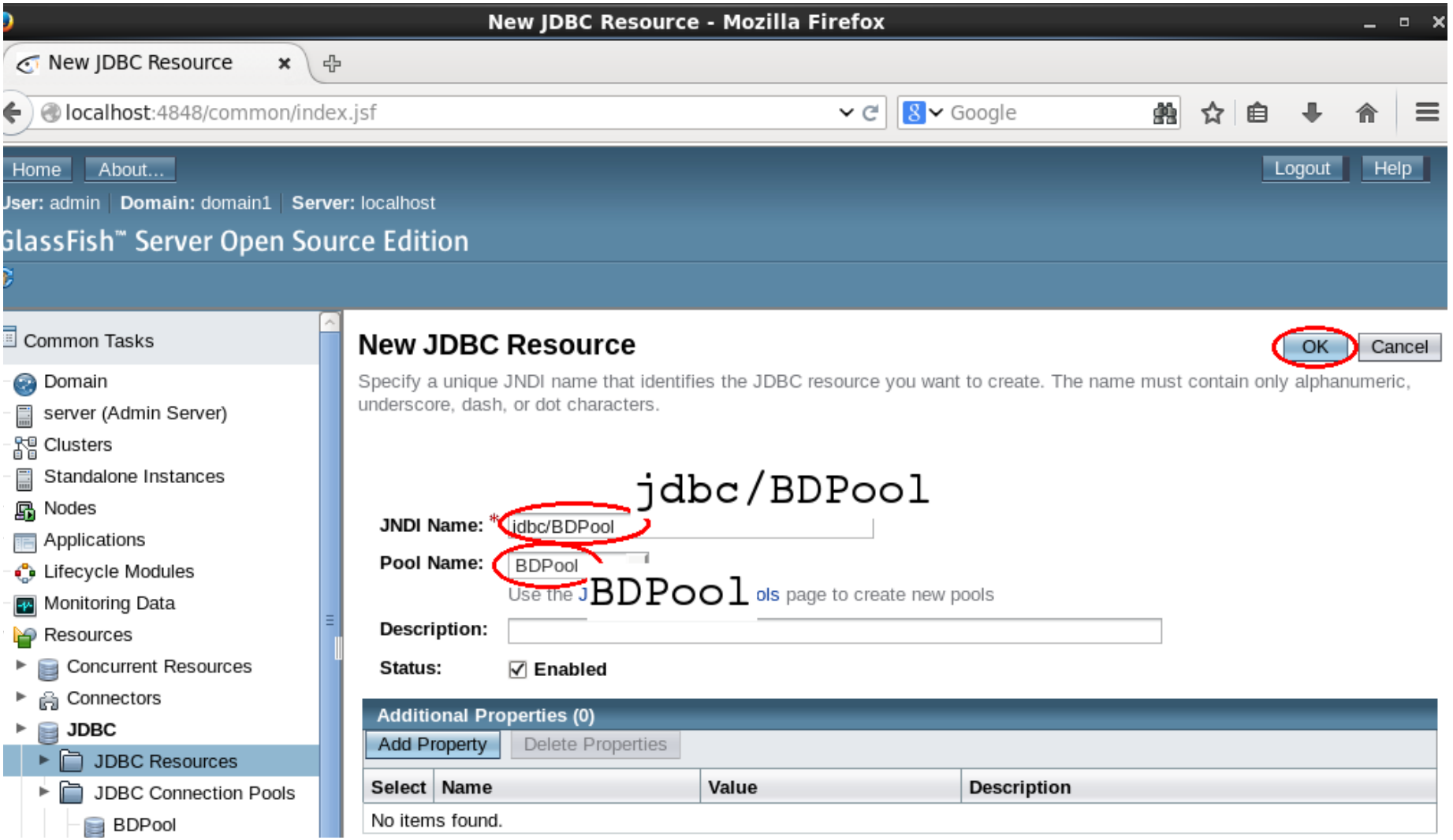
Crear un Connection Pool a MySQL (VIII)



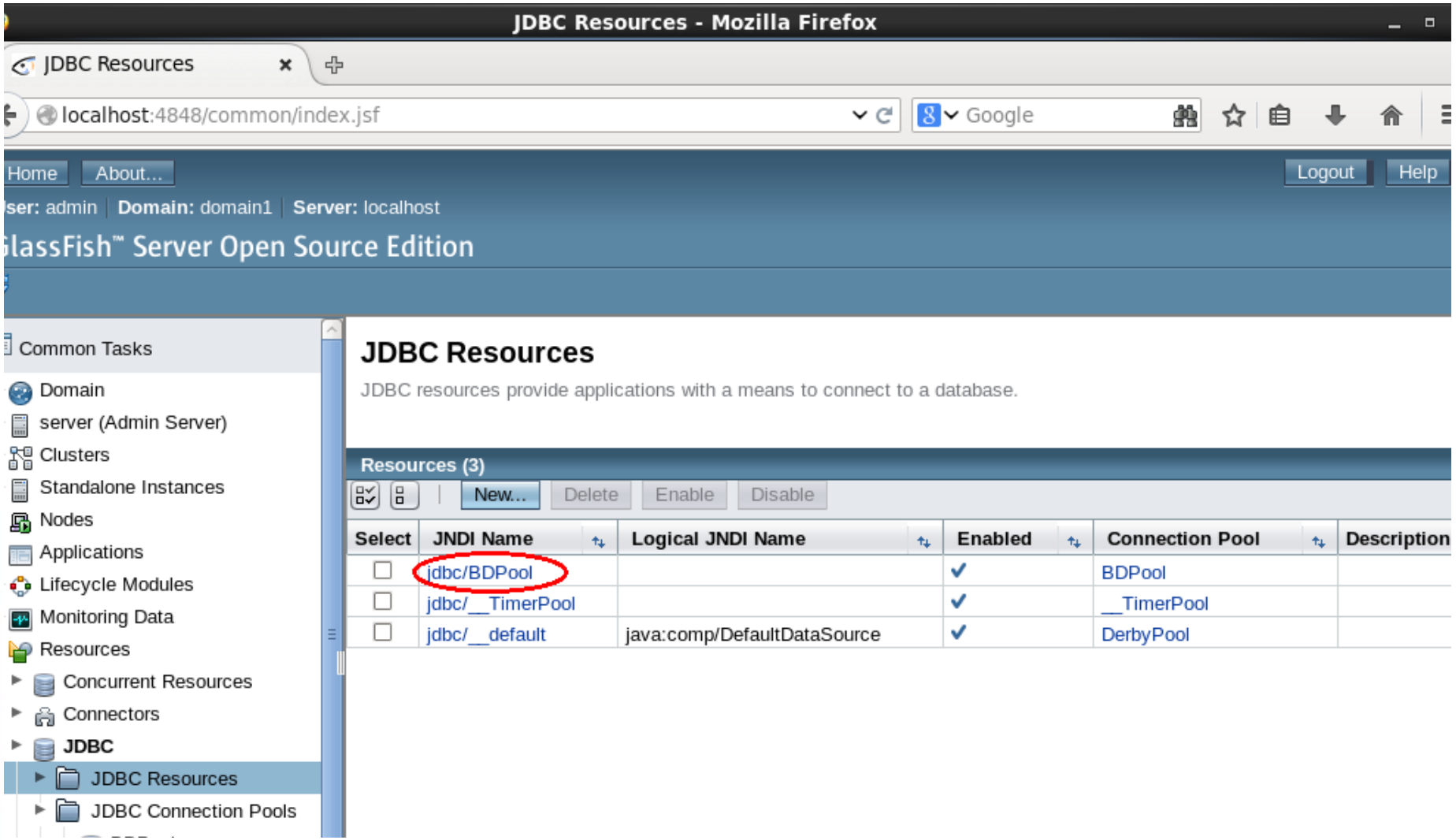
Crear un recurso JDBC a MySQL (I)



Crear un recurso JDBC a MySQL (II)



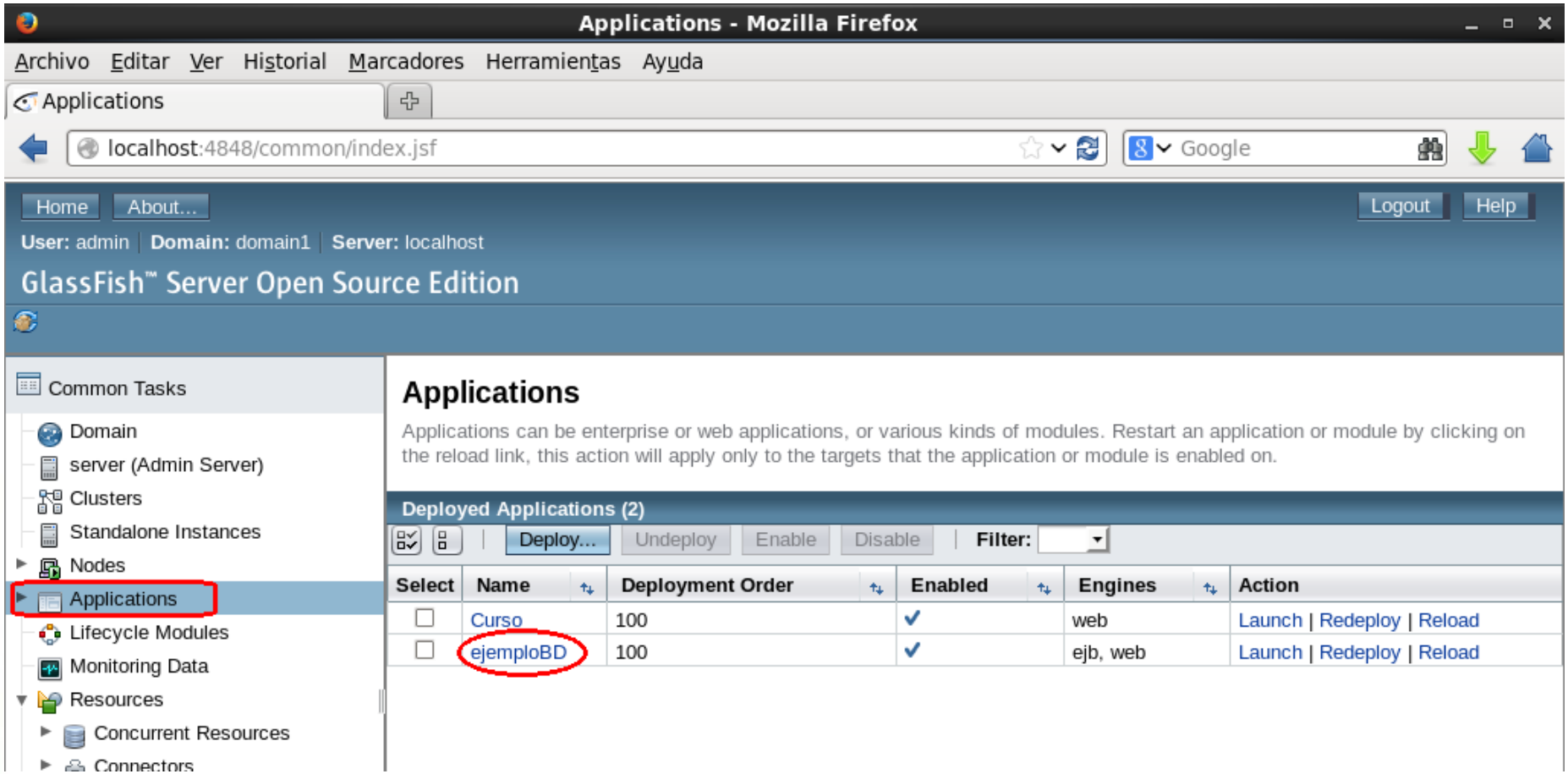
Crear un recurso JDBC a MySQL (III)



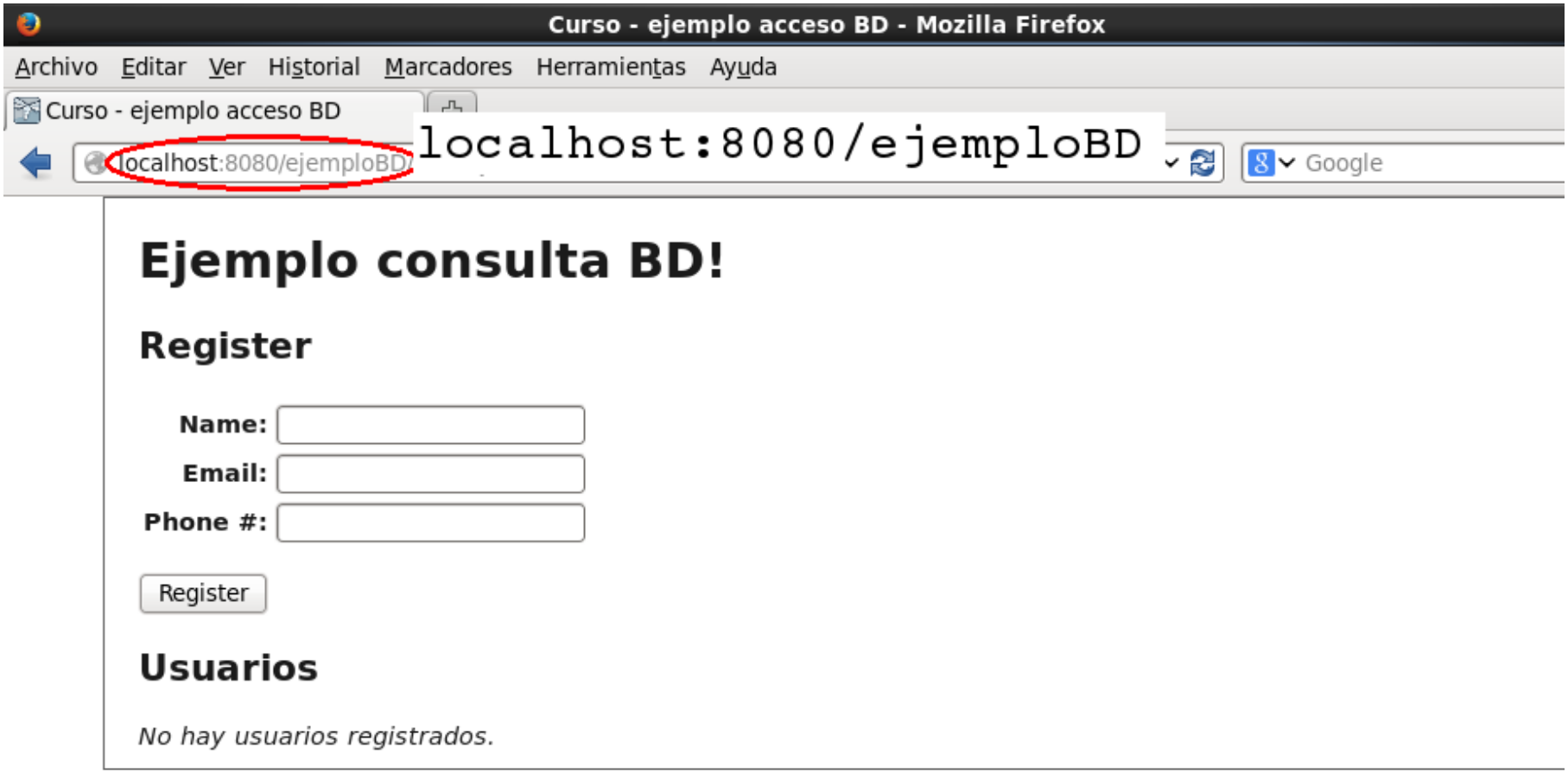
Acceso a MySQL desde Glassfish (I)

- Llegados a este punto ya está configurada la conexión del Glassfish con MySQL.
  - Es necesario reiniciar el servidor de Glassfish para utilizar el recurso JDBC creado.
- Para comprobar que funciona vamos a desplegar la aplicación EjemploBD.war.
- Esta aplicación utiliza el recurso JDBC creado anteriormente (jdbc/BDPool).
  - No es necesario definir los datos de usuario, contraseña y nombre de la base de datos en la aplicación, puesto que ya están definidos en el Connection Pool que utiliza el recurso JDBC.

Acceso a MySQL desde Glassfish (II)



Acceso a MySQL desde Glassfish (III)



Acceso a MySQL desde Glassfish (IV)

Acceso a MySQL desde Glassfish (V)

Acceso a MySQL desde Glassfish (VI)

- Antes de realizar el acceso a PostgreSQL, eliminamos lo que hemos creado en el servidor de GlassFish:
  - Eliminar la aplicación ejemploBD
  - Eliminar el recurso jdbc/BDPool
  - Eliminar el Connection Pool BDPool

Conexión PostgreSQL - Glassfish (I)

- Ahora vamos a realizar la misma práctica pero conectaremos con la base de datos PostgreSQL.
  - Crearemos inicialmente una base de datos y un usuario, ambos con nombre curso:

```
CREATE DATABASE curso;
CREATE ROLE curso LOGIN ENCRYPTED PASSWORD
'curso-araw';
ALTER DATABASE curso OWNER TO curso;
```

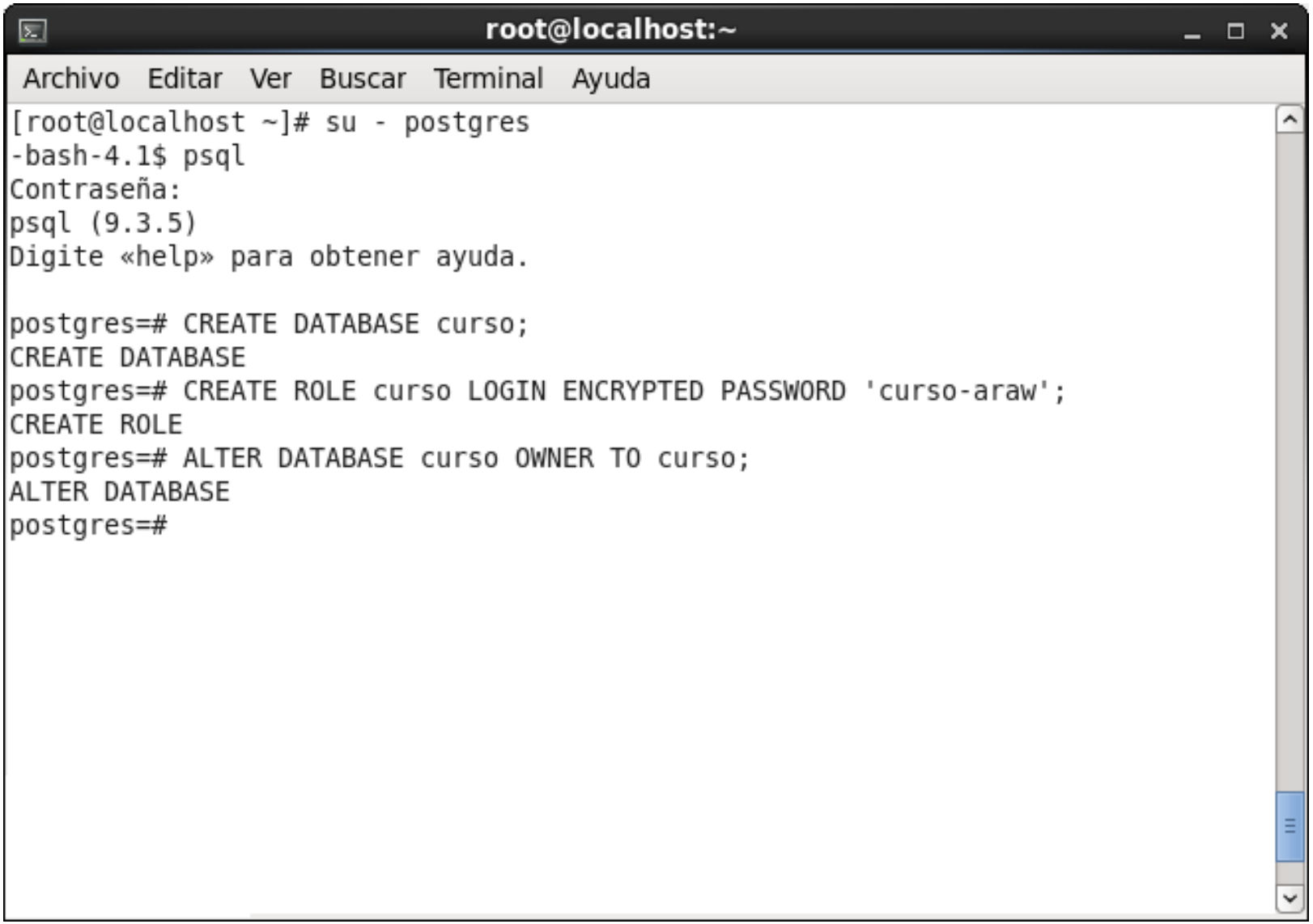
  - Debemos descargar el conector de Java para la base de datos desde la URL:

```
http://jdbc.postgresql.org/download.html
```

  - Para JVM 1.7 descargamos la versión JDBC41.

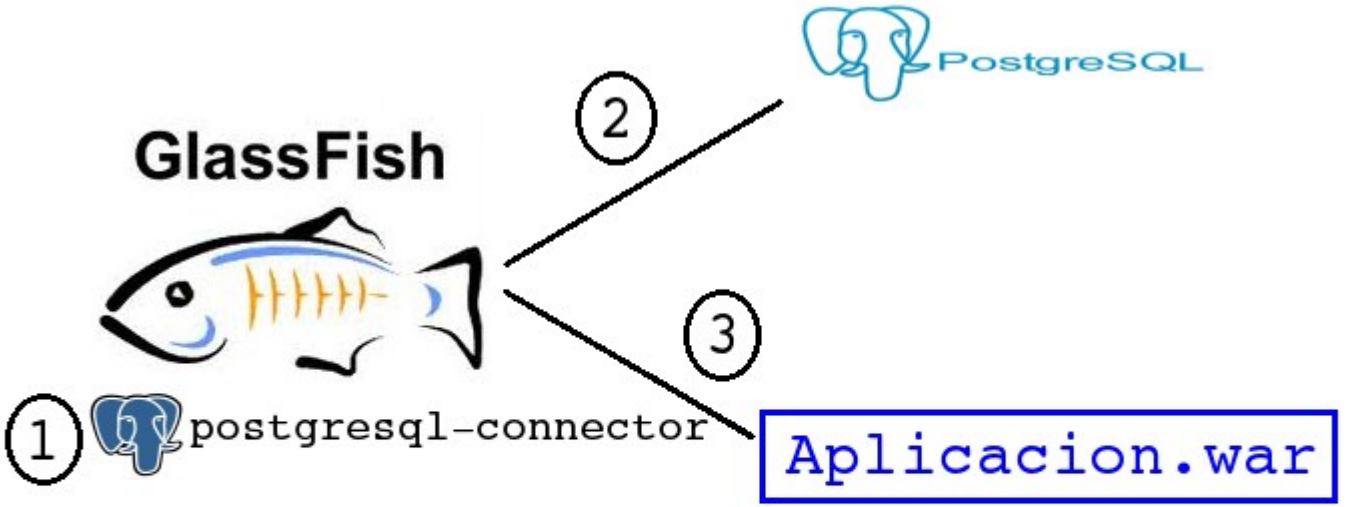
```
postgresql-<version>.jdbc41.jar
```

Conexión PostgreSQL - Glassfish (II)



Conexión PostgreSQL – Glassfish (III)

- Para que la aplicación desplegada en Glassfish acceda a una base de datos PostgreSQL, se deben seguir los mismos pasos que en el caso anterior:

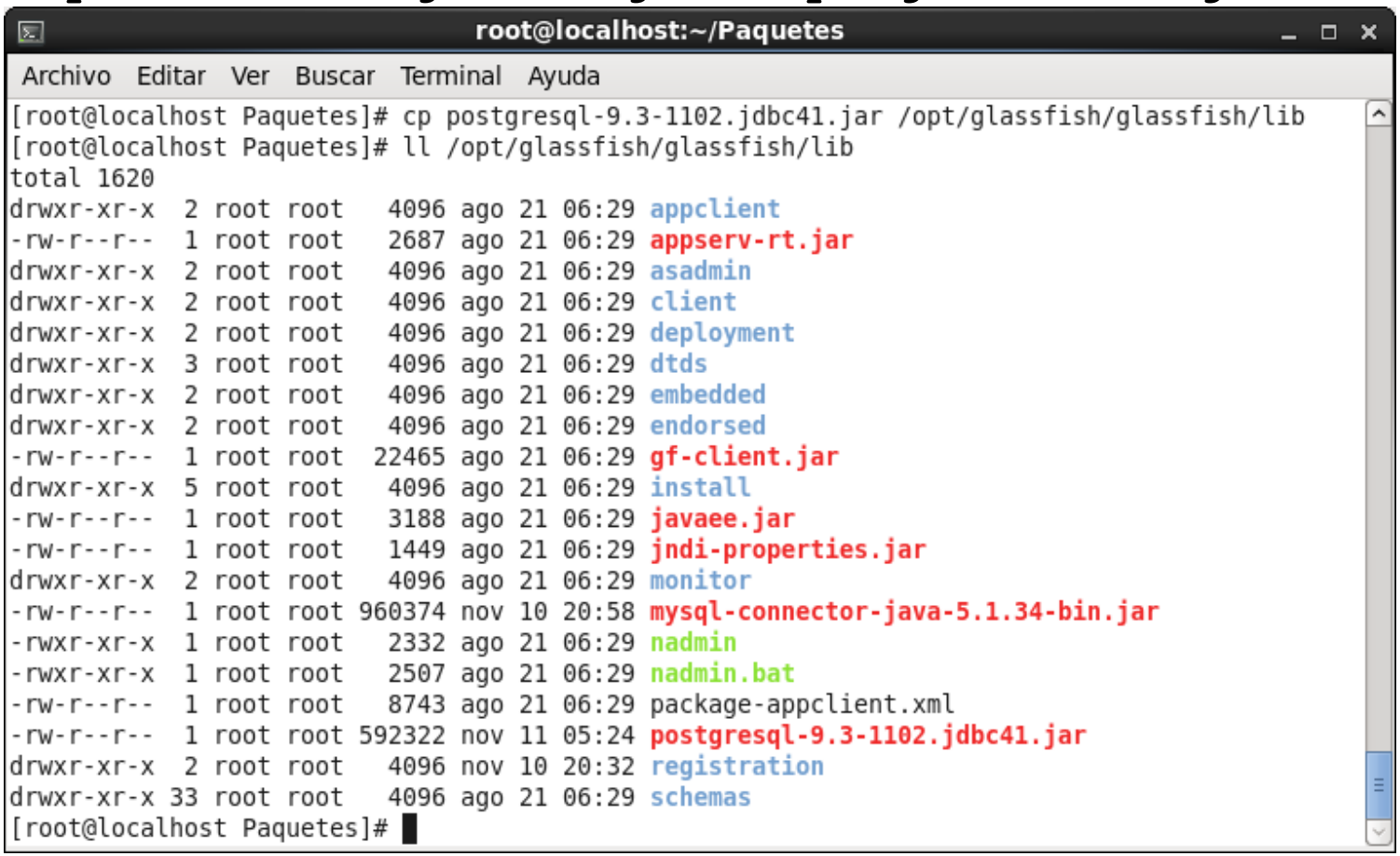


1. Instalar el conector PostgreSQL
2. Crear un Connection Pool
3. Crear un recurso JDBC

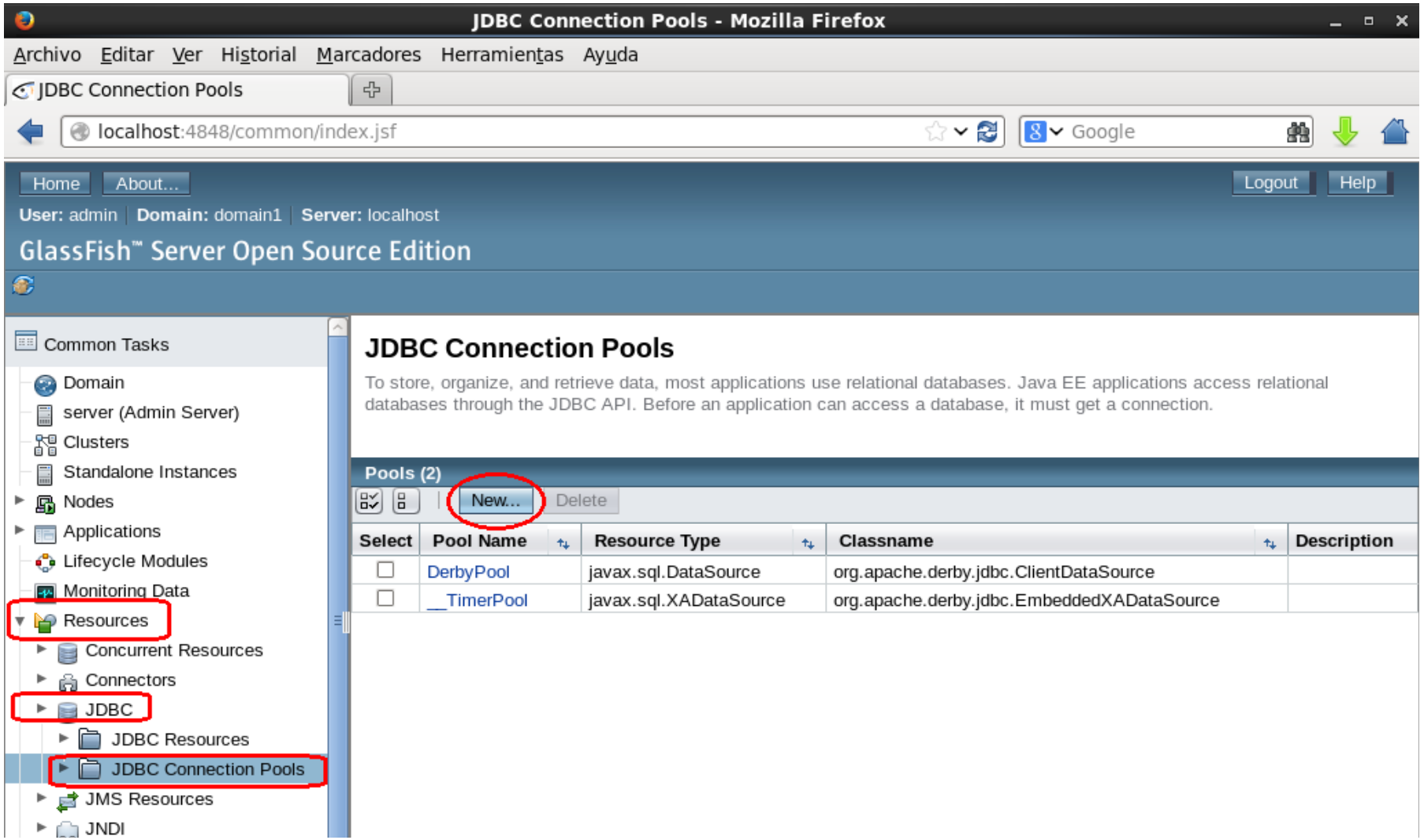
Instalar el conector de PostgreSQL

- Copiar el fichero **postgresql-<version>.jdbc41.jar** en la carpeta en que Glassfish almacena las librerías.

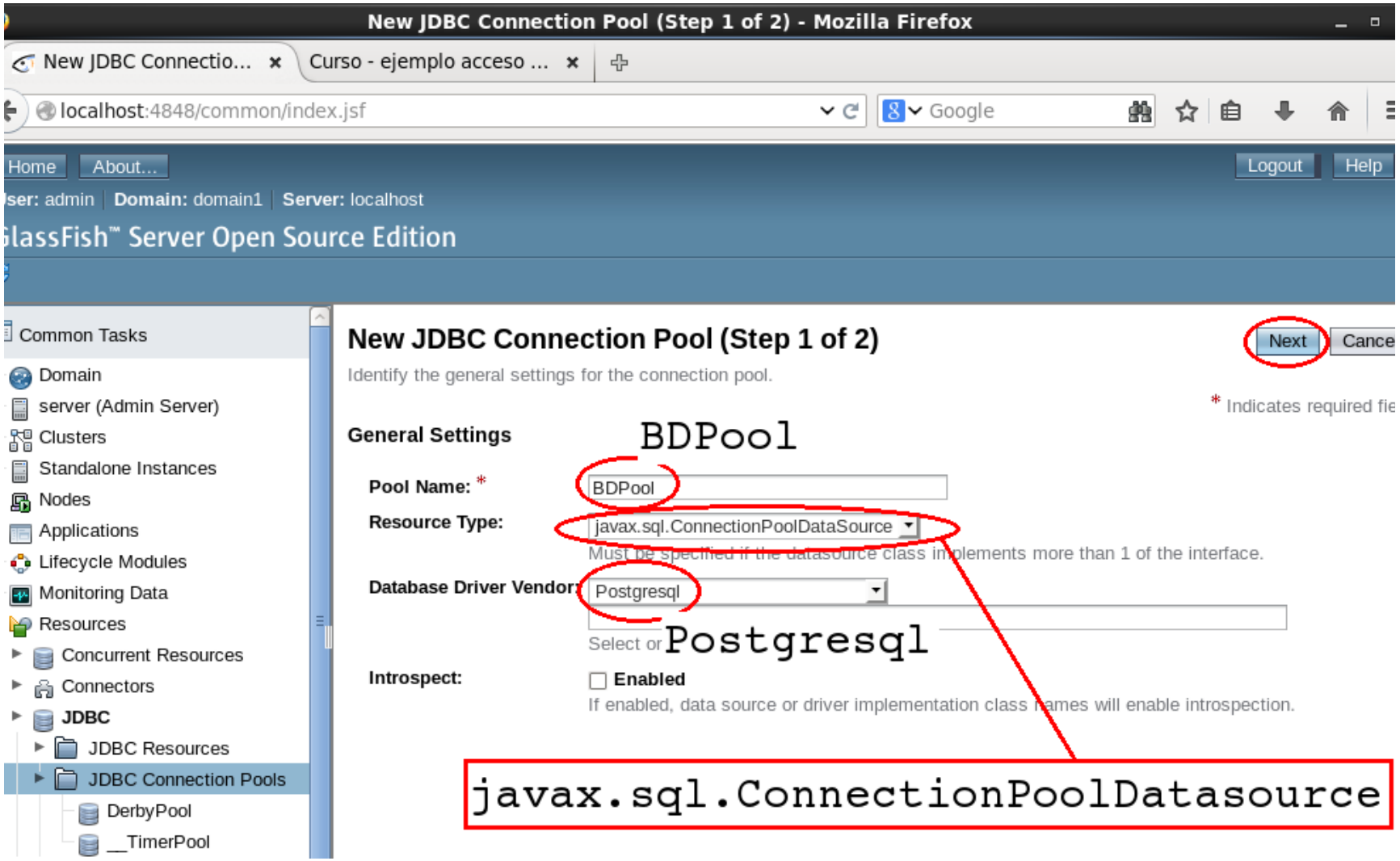
```
cp postgresql-9.3-1102.jdbc41.jar /opt/glassfish/glassfish/lib/
```



Crear un Connection Pool a PostgreSQL (I)



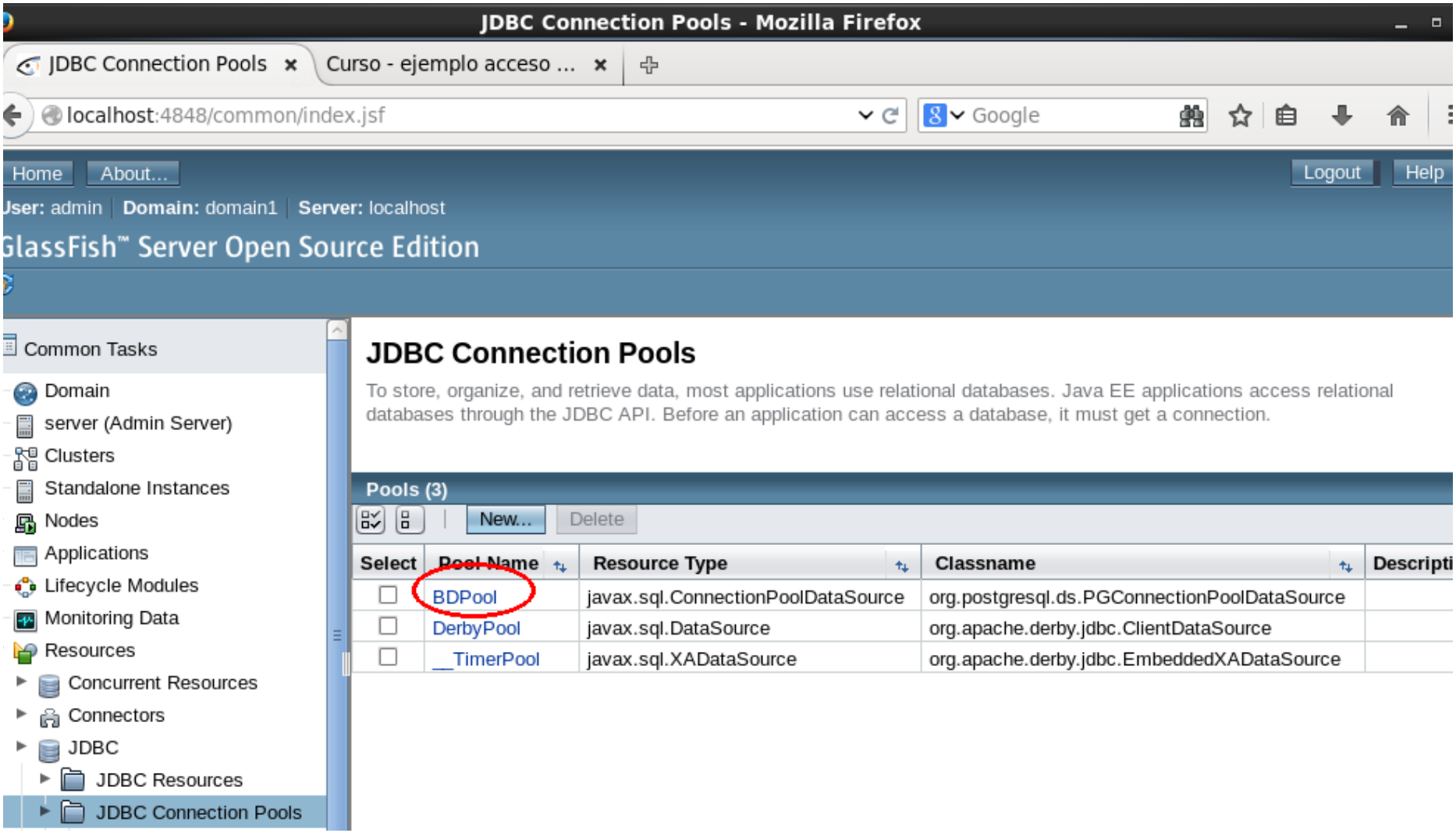
Crear un Connection Pool a PostgreSQL (II)



Crear un Connection Pool a PostgreSQL (III)

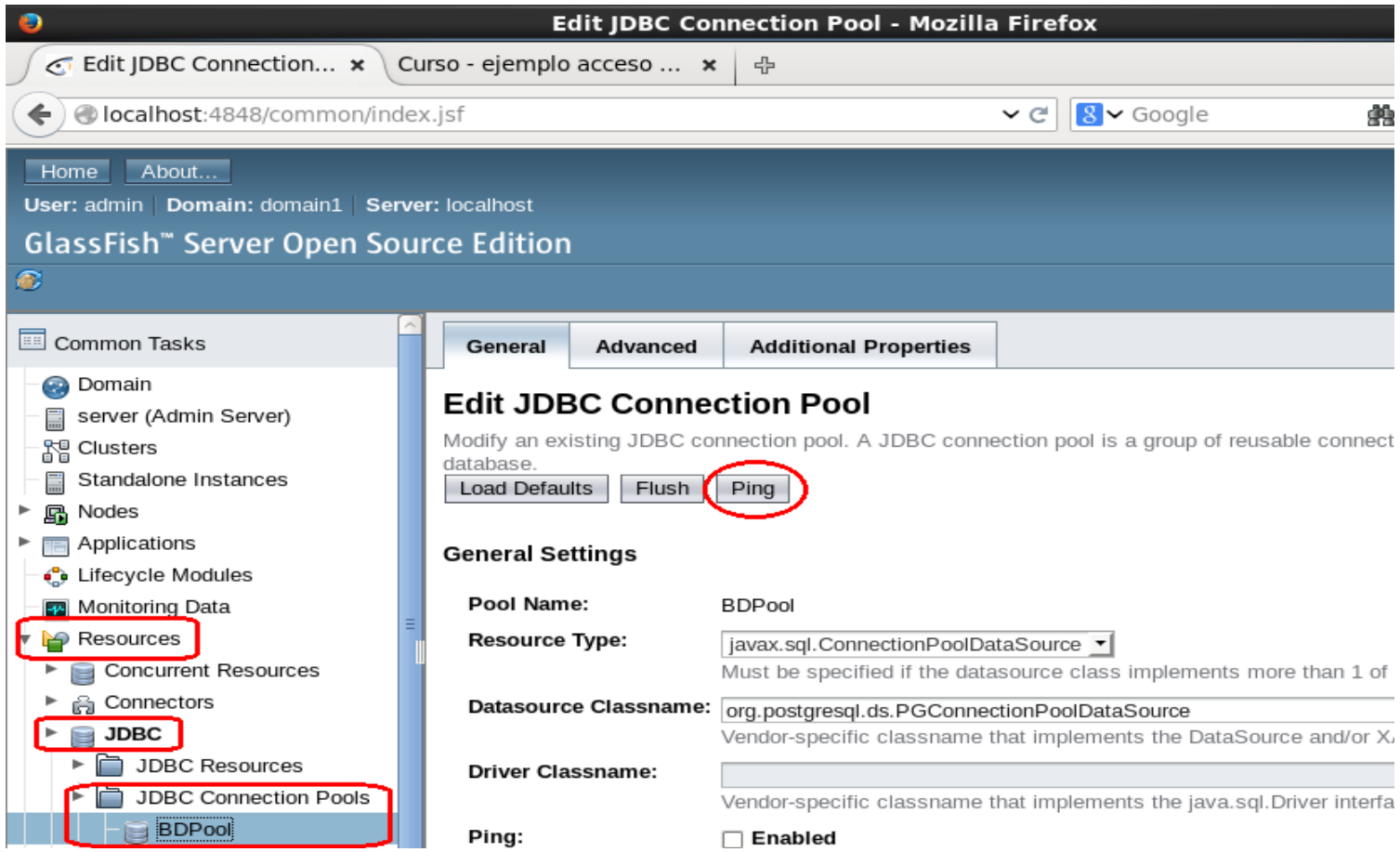
- Dejar por defecto los valores de “General Settings”.
- Dejar por defecto los valores de “Pool Settings”.
- Añadir en “Additional Properties” los siguientes valores:
  - DatabaseName: curso
  - User: curso
  - Password: curso-araw
  - PortNumber: 5432
- Pulsar en “Finish”.

Crear un Connection Pool a PostgreSQL (IV)

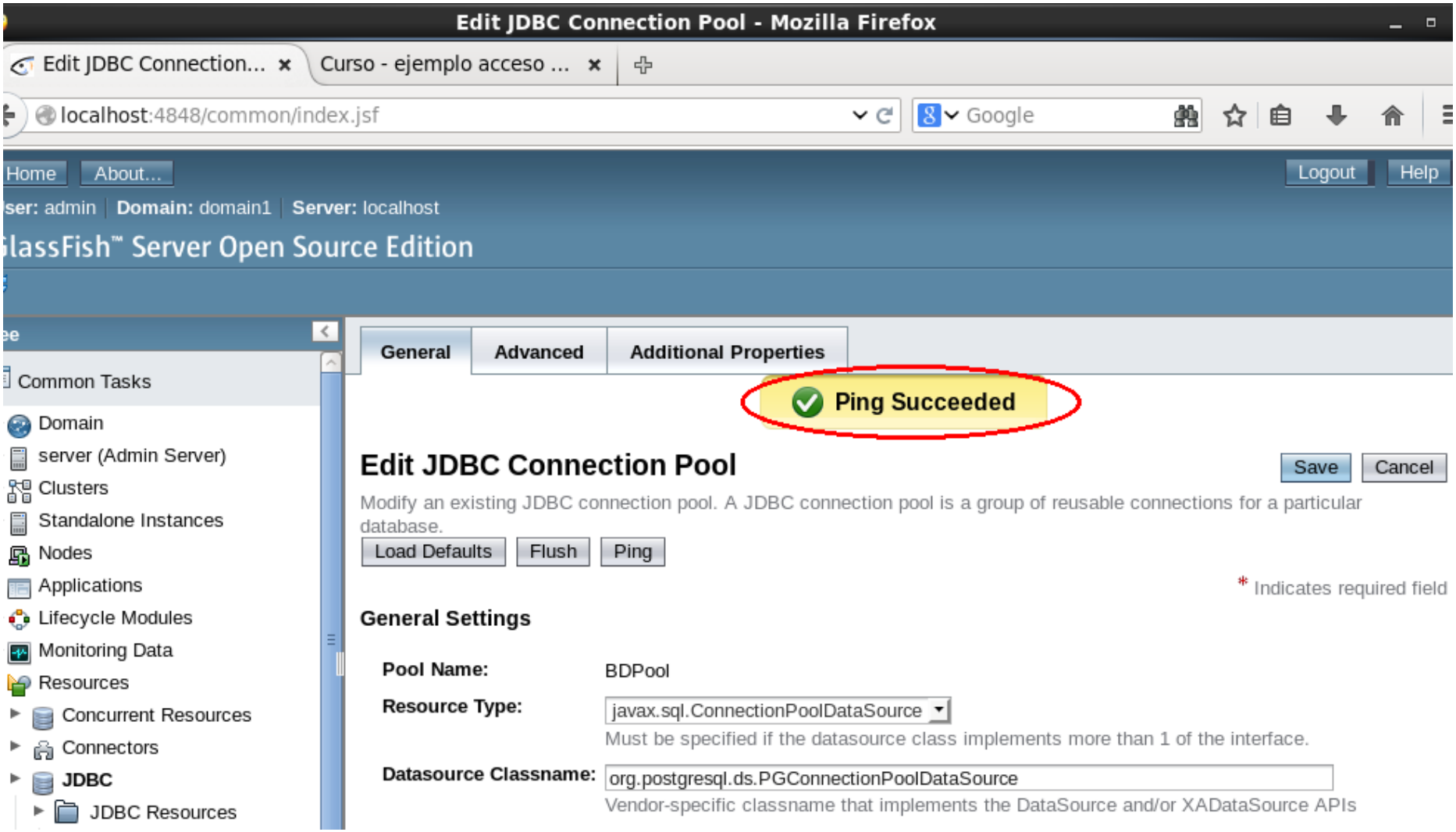


Crear un Connection Pool a PostgreSQL (V)

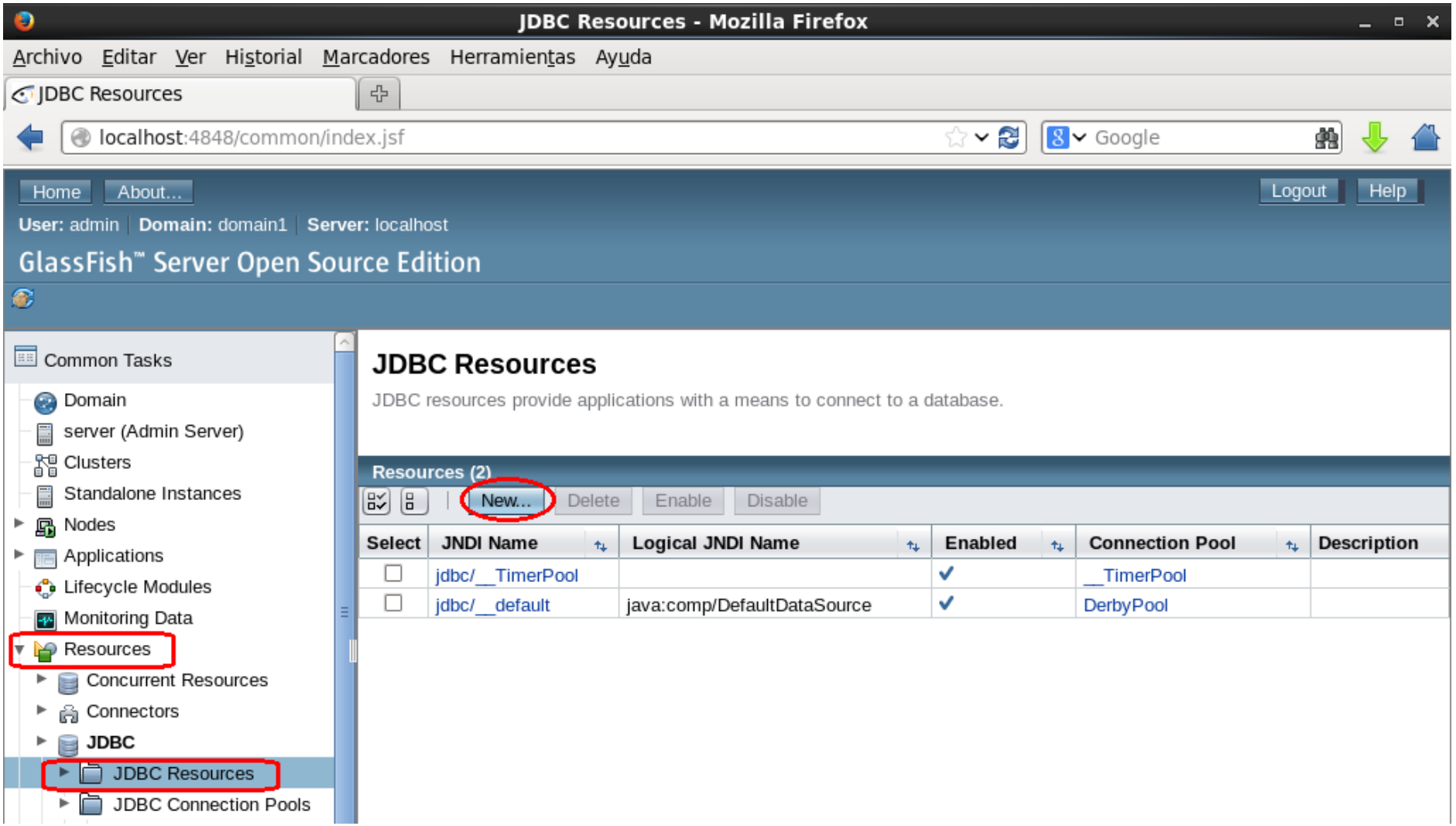
- Para comprobar que funciona el Connection Pool, se puede hacer un ping (es necesario reiniciar el GlassFisch y el servidor de PostgreSQL debe estar iniciado).



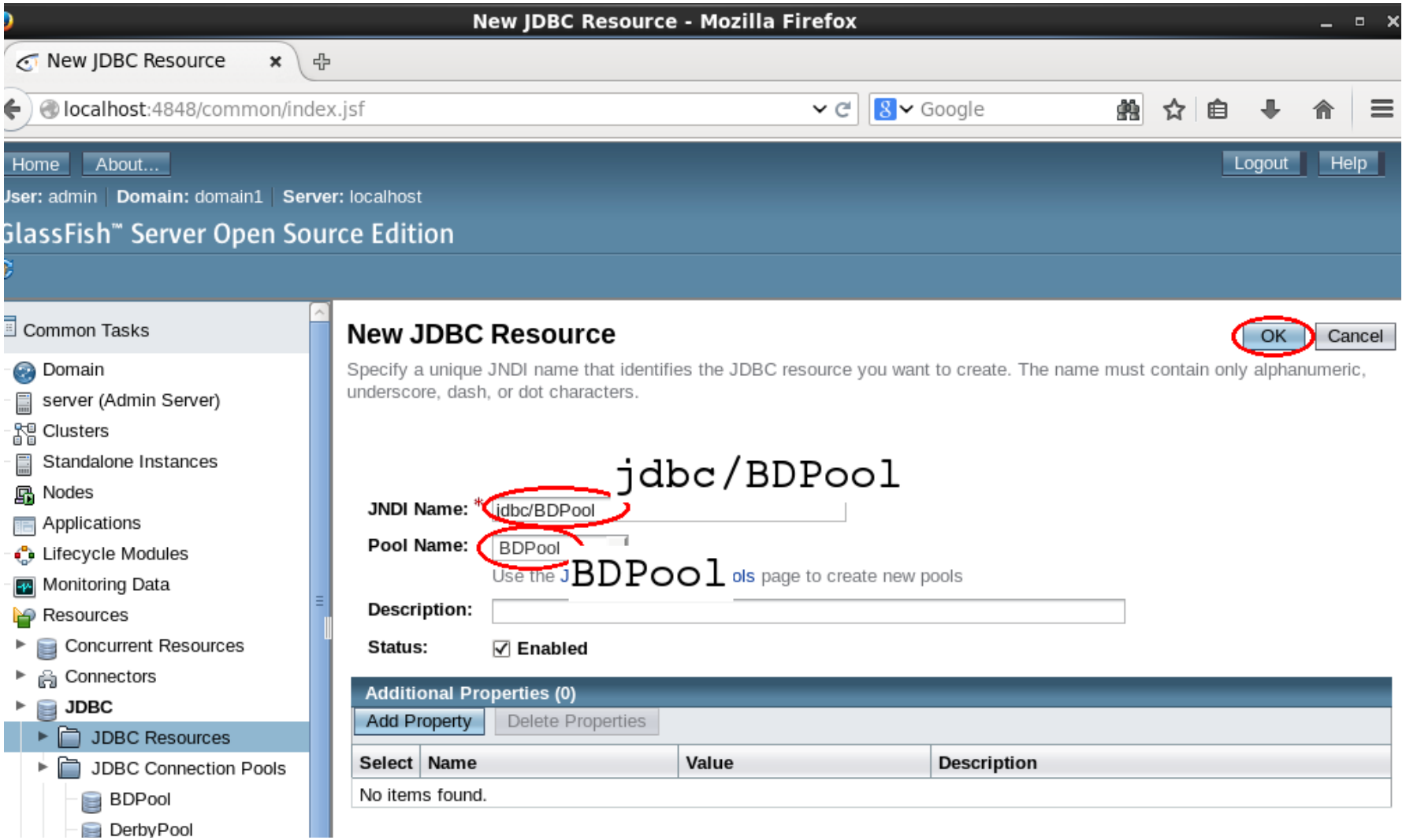
Crear un Connection Pool a PostgreSQL (VI)



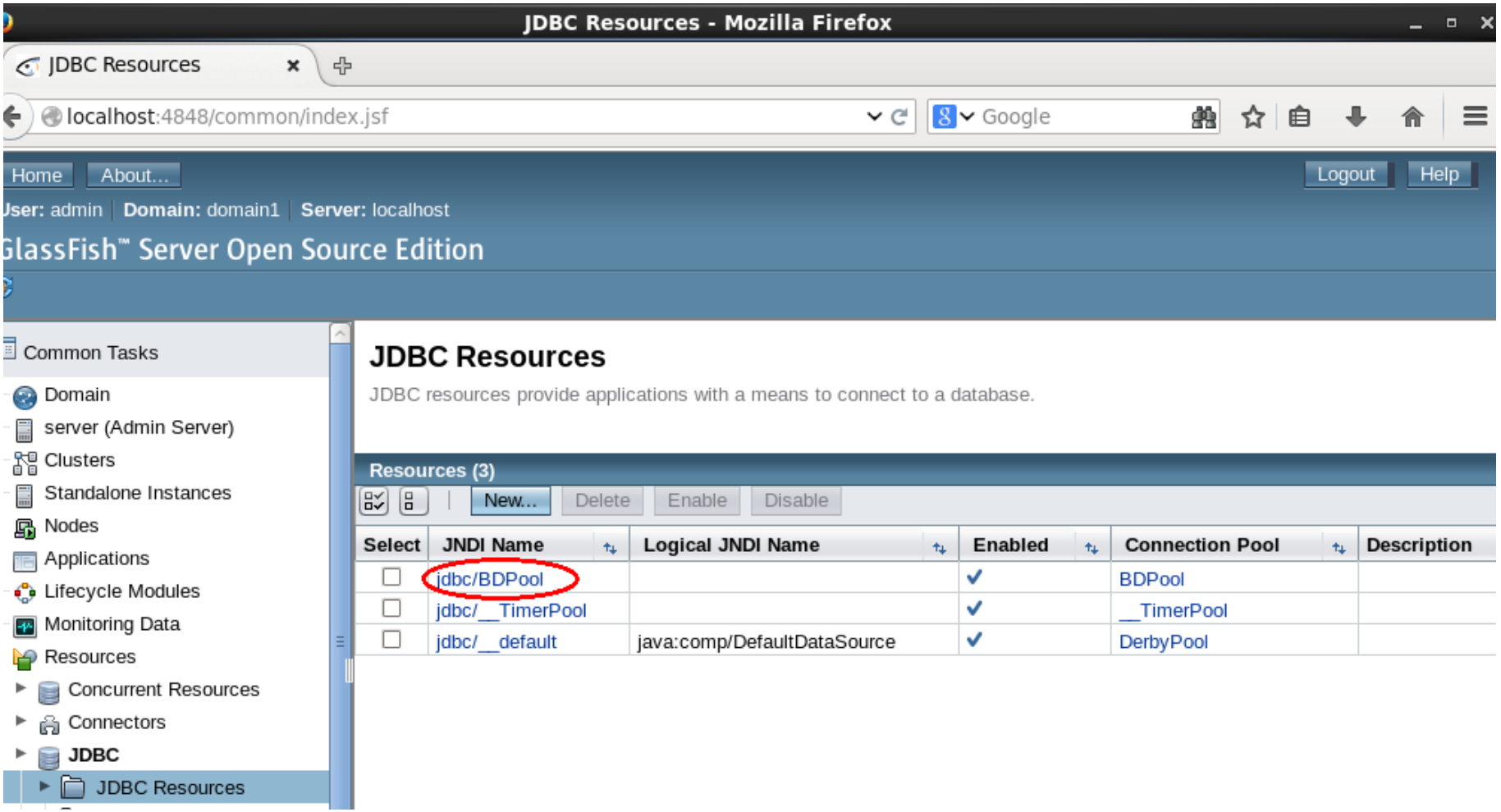
Crear un recurso JDBC a PostgreSQL (I)



Crear un recurso JDBC a PostgreSQL (II)



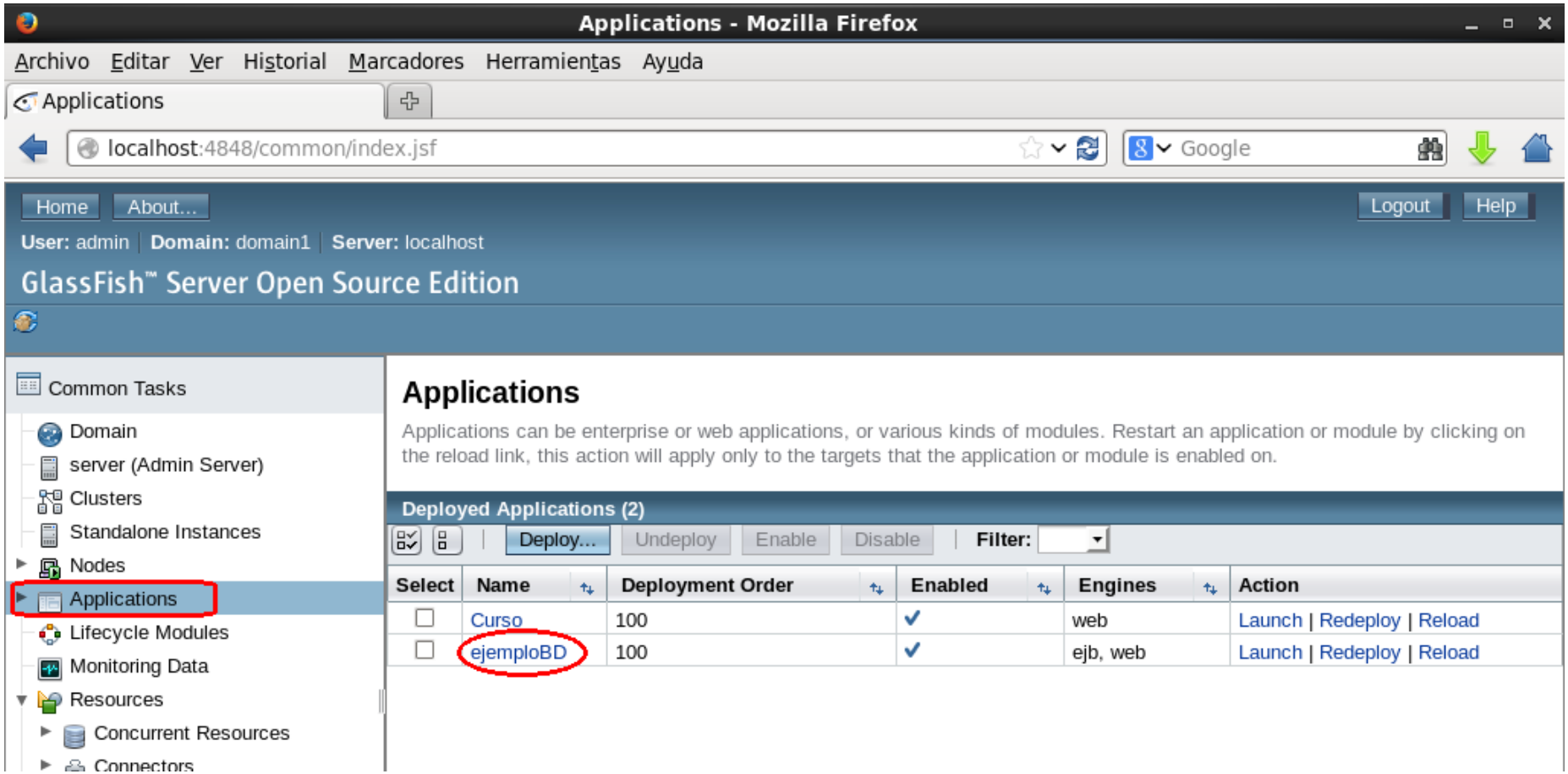
Crear un recurso JDBC a PostgreSQL (III)



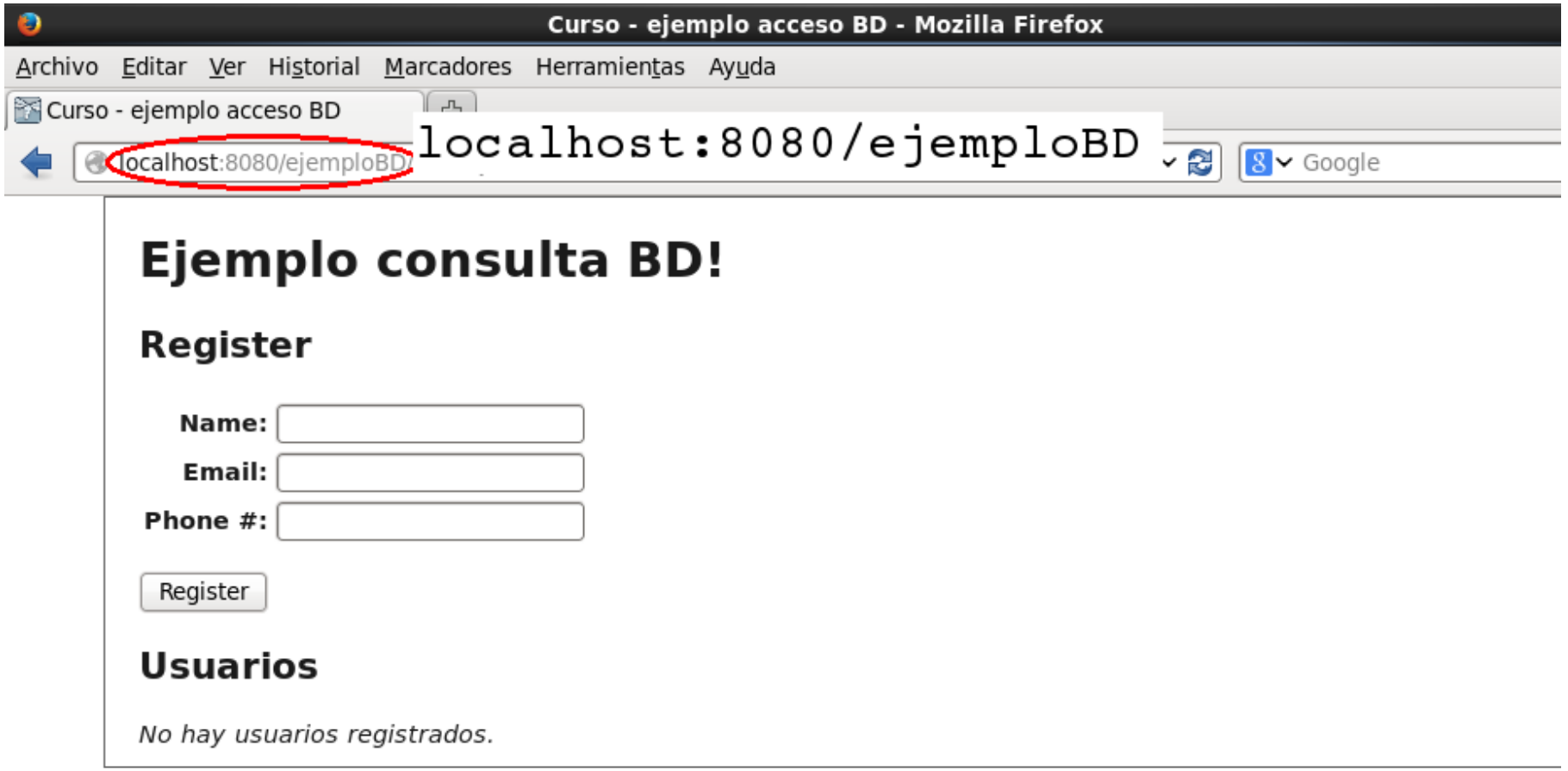
Acceso a PostgreSQL desde Glassfish (I)

- Una vez configurada la conexión de GlassFish con PostgreSQL es necesario reiniciar el servidor de GlassFish para poder utilizar el nuevo recurso JDBC.
- Para comprobar que funciona vamos a volver a desplegar la aplicación EjemploBD.war.
- Esta aplicación utiliza el recurso JDBC creado anteriormente (jdbc/BDPool), en este caso con los valores apuntando a la base de datos PostgreSQL.
  - No es necesario definir los datos de usuario, contraseña y nombre de la base de datos en la aplicación, puesto que ya están definidos en el Connection Pool que utiliza el recurso JDBC.

Acceso a PostgreSQL desde Glassfish (II)



Acceso a PostgreSQL desde Glassfish (III)

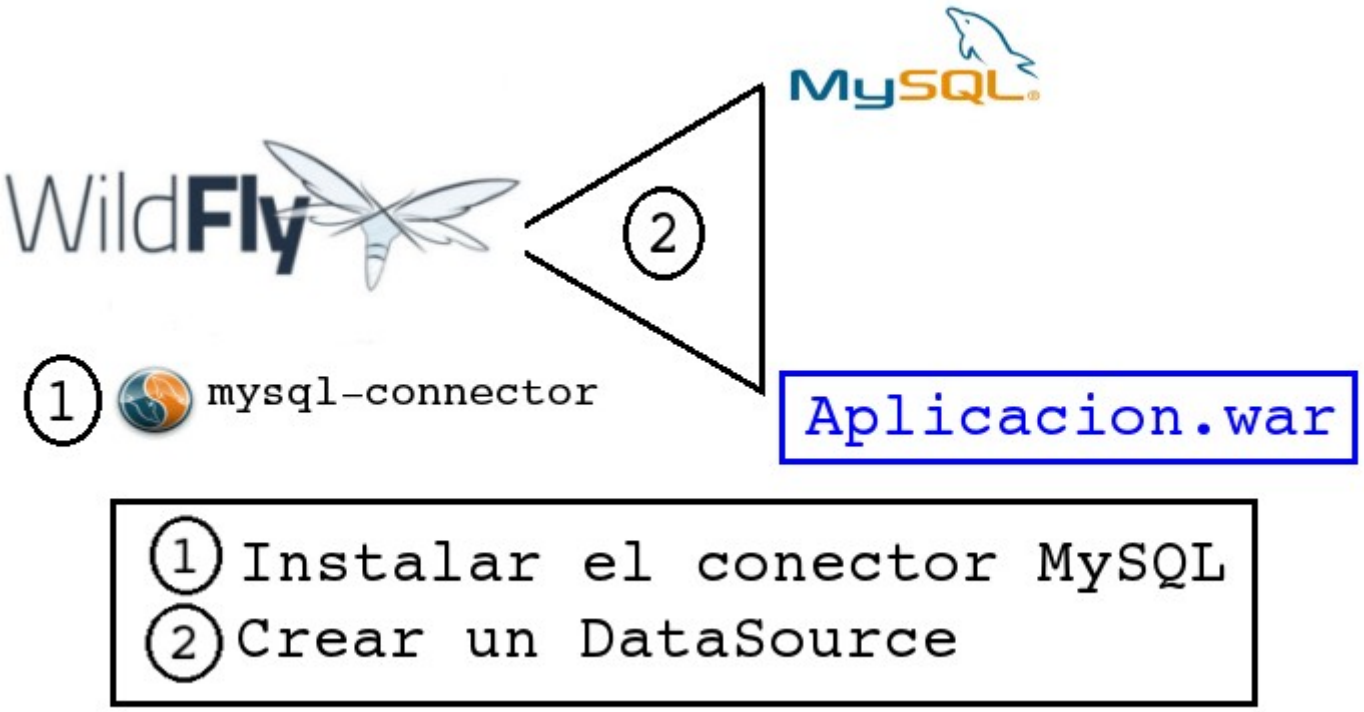


Acceso a PostgreSQL desde Glassfish (IV)

Acceso a PostgreSQL desde Glassfish (V)

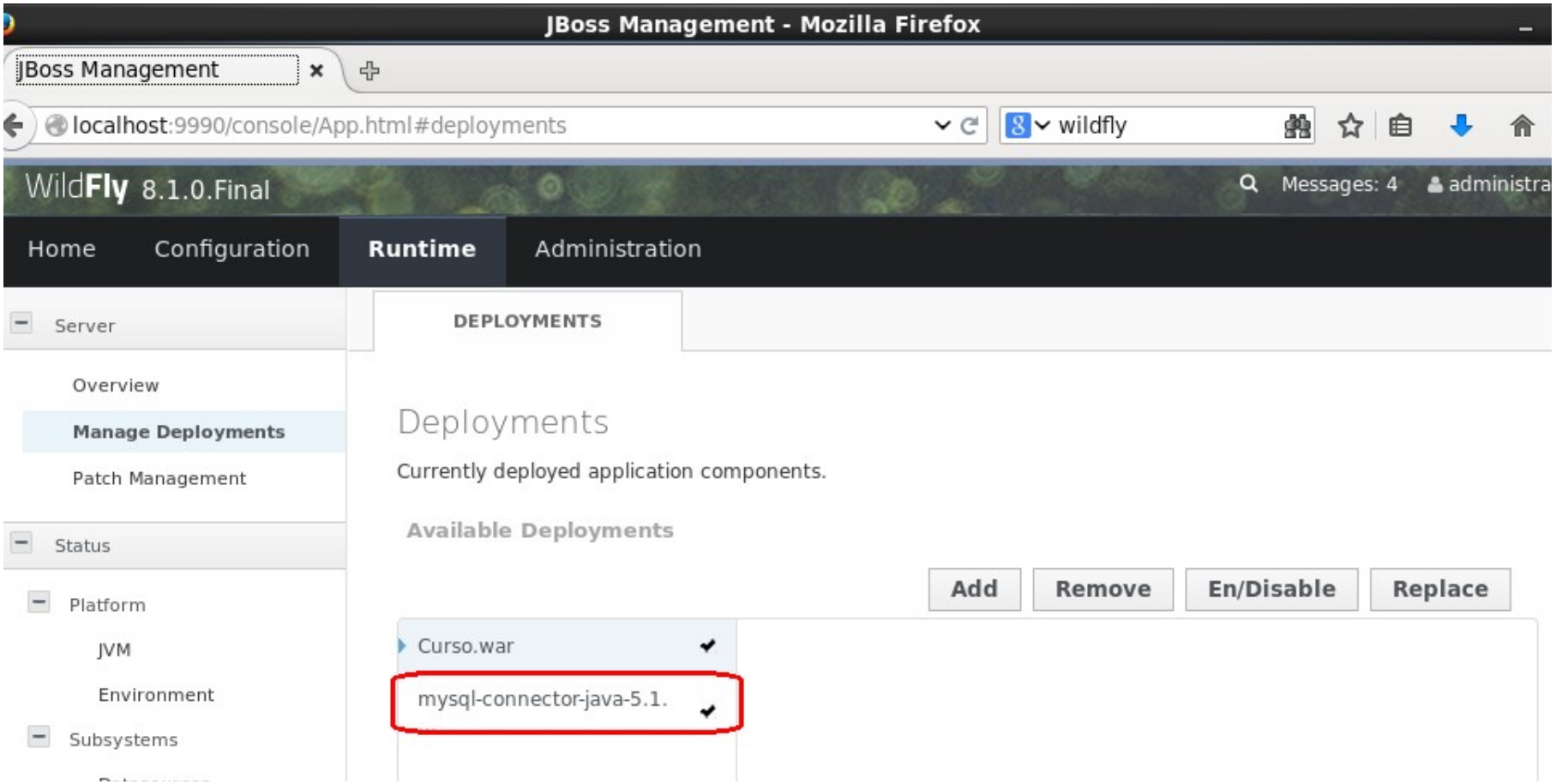
Conexión MySQL - WildFly

- Para que una aplicación desplegada en WildFly acceda a una base de datos MySQL, en la parte del servidor hay que seguir 2 pasos:

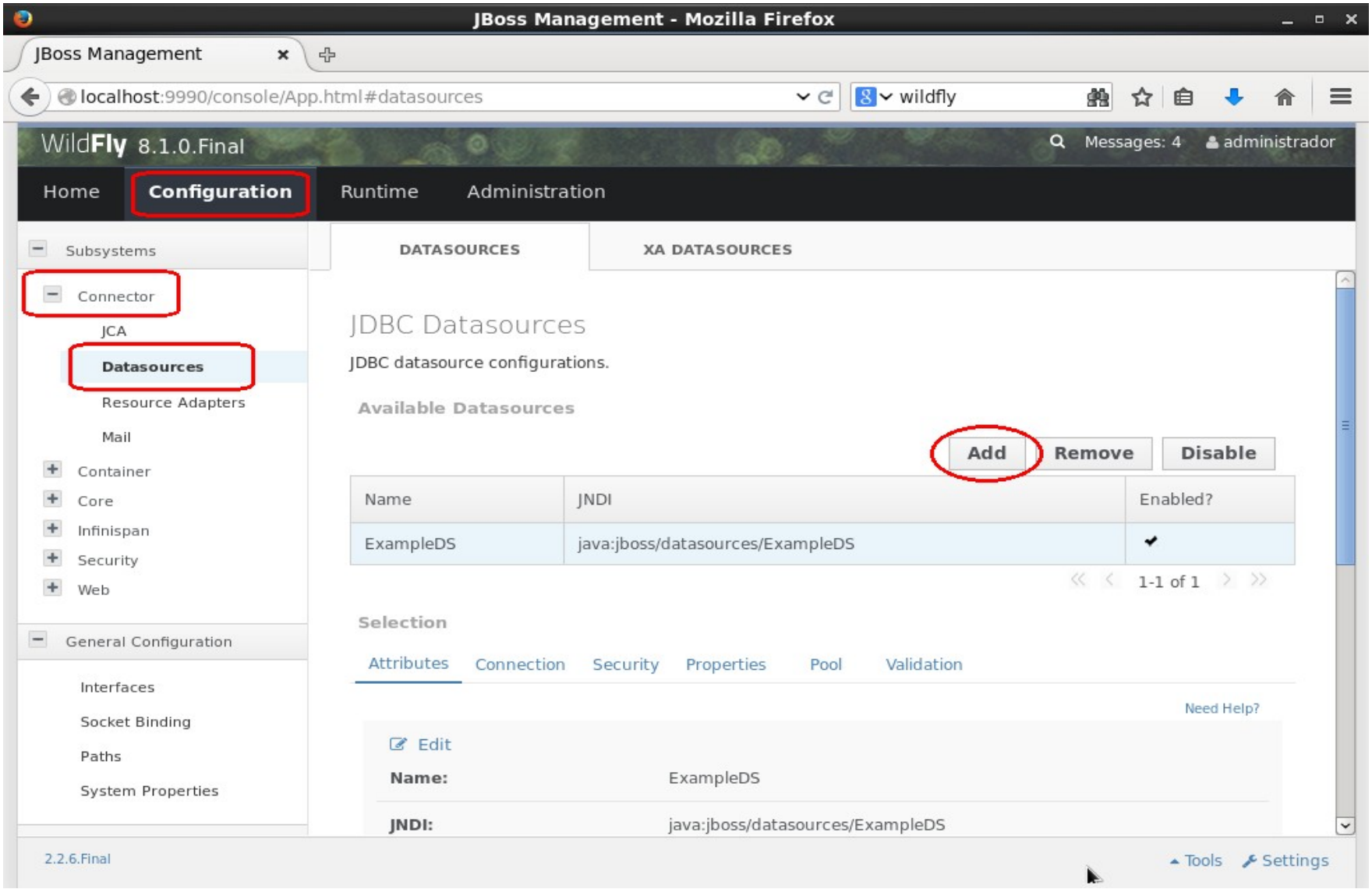


Instalar el conector de MySQL

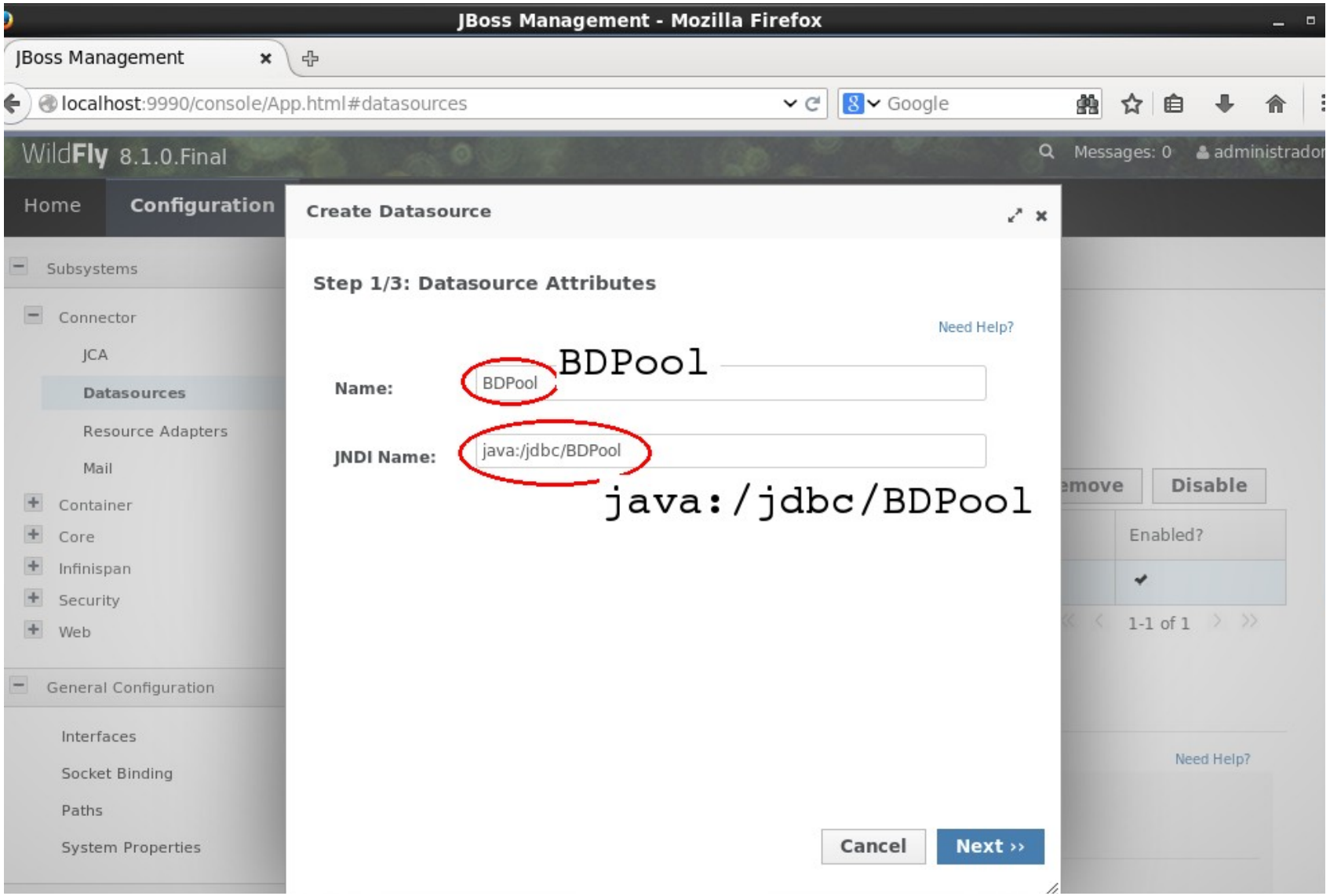
- Desplegar el **mysql-connector-java-<version>-bin.jar** como cualquier aplicación de WildFly.



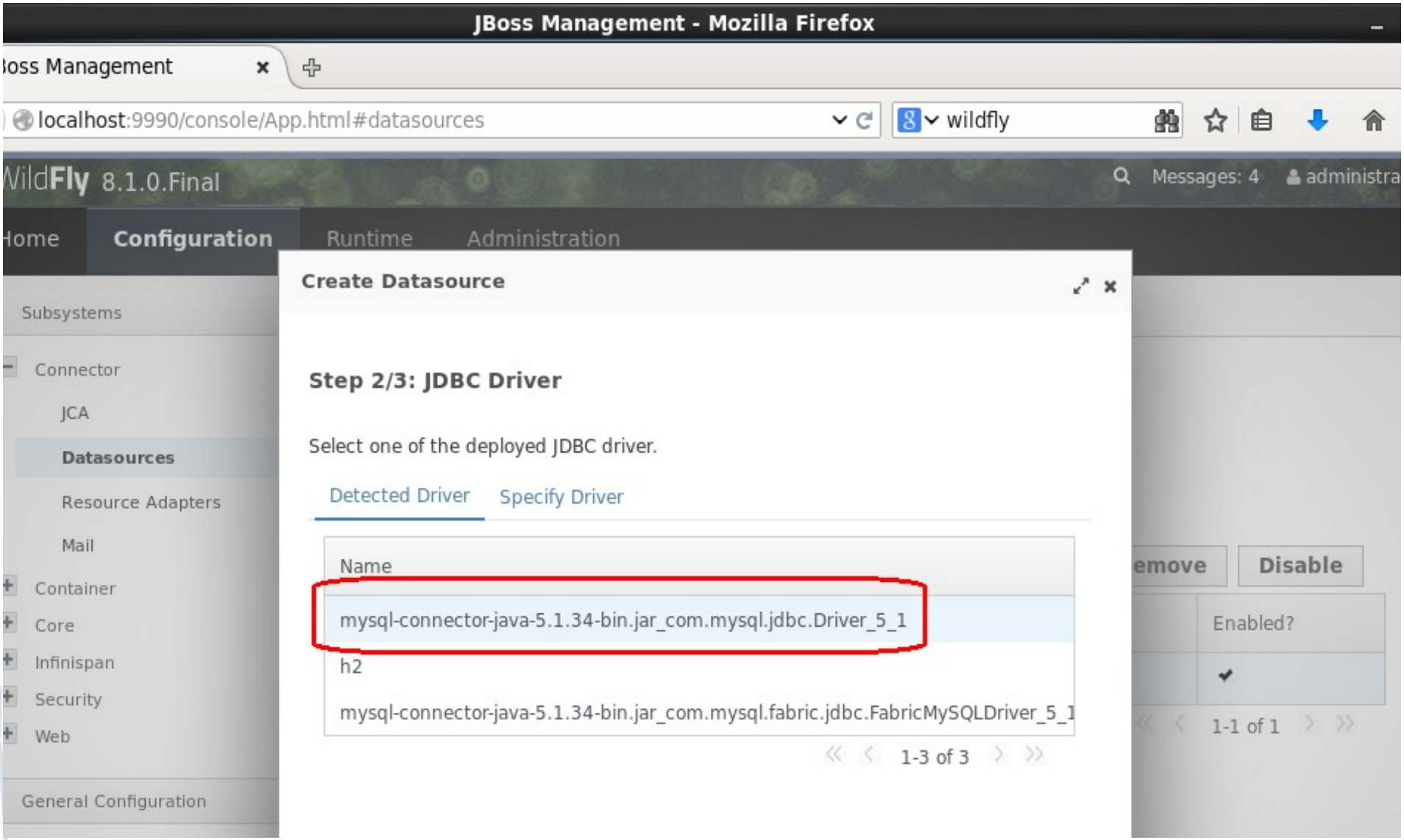
Crear un DataSource (I)



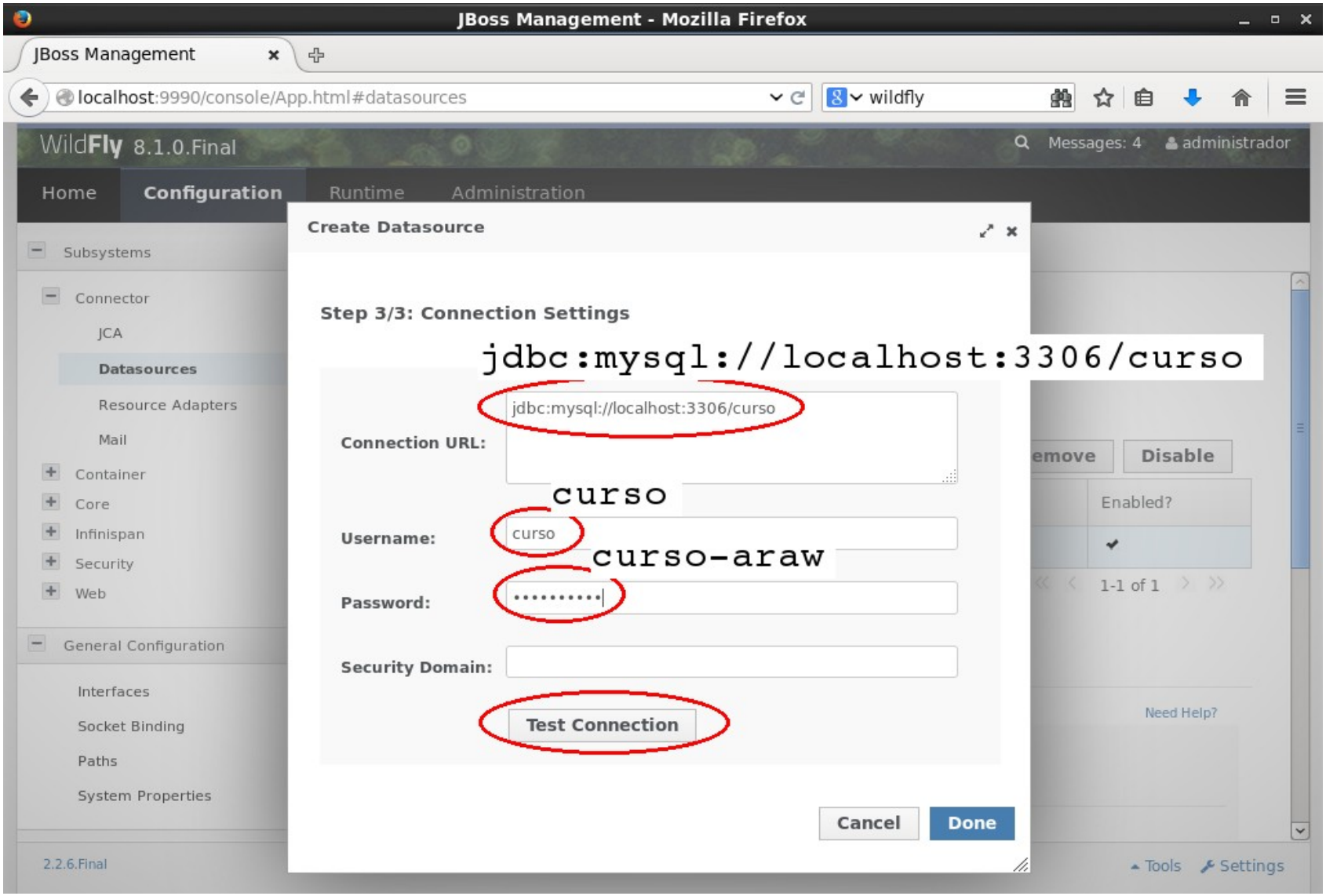
Crear un DataSource (II)



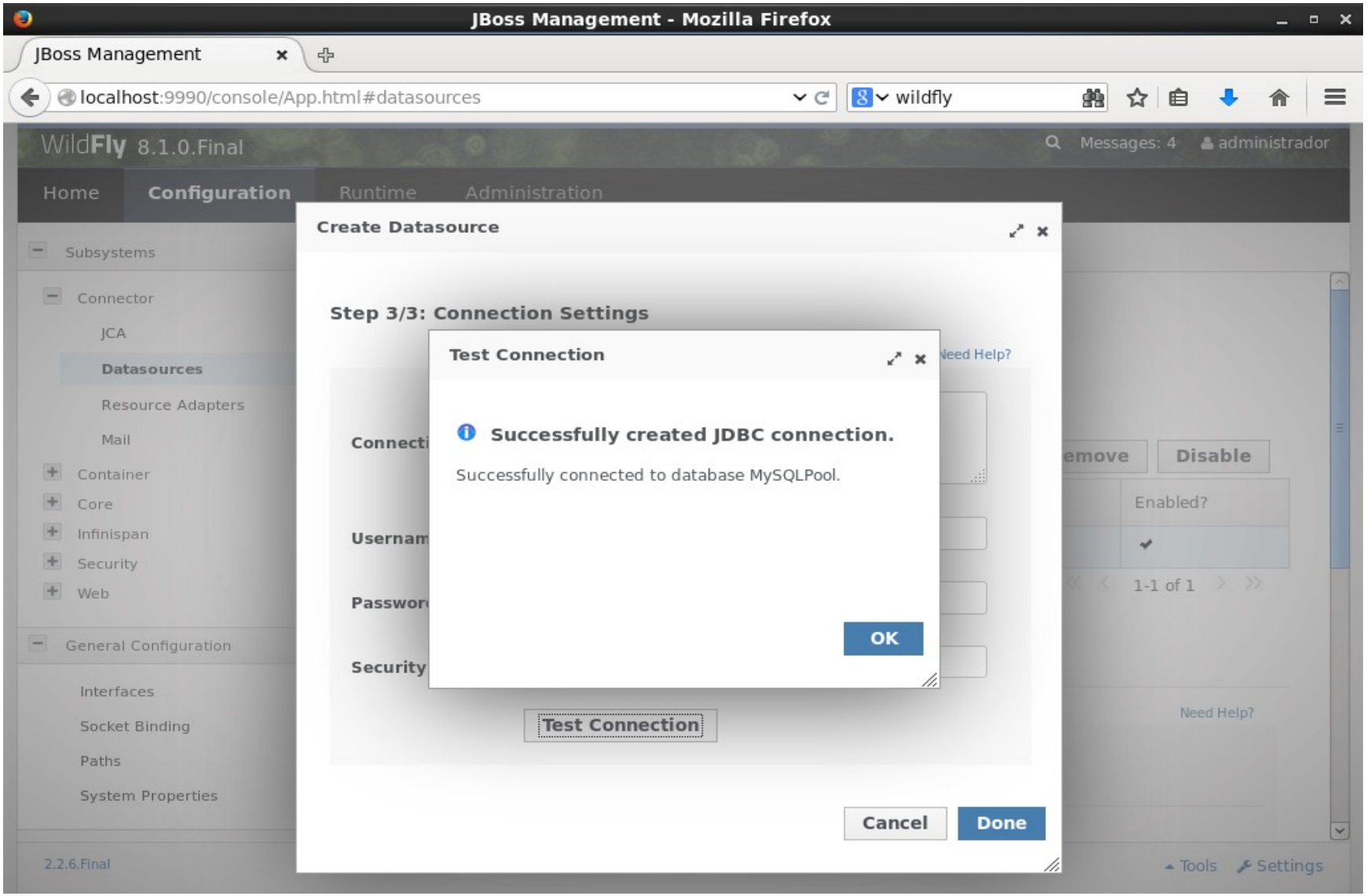
Crear un DataSource (III)



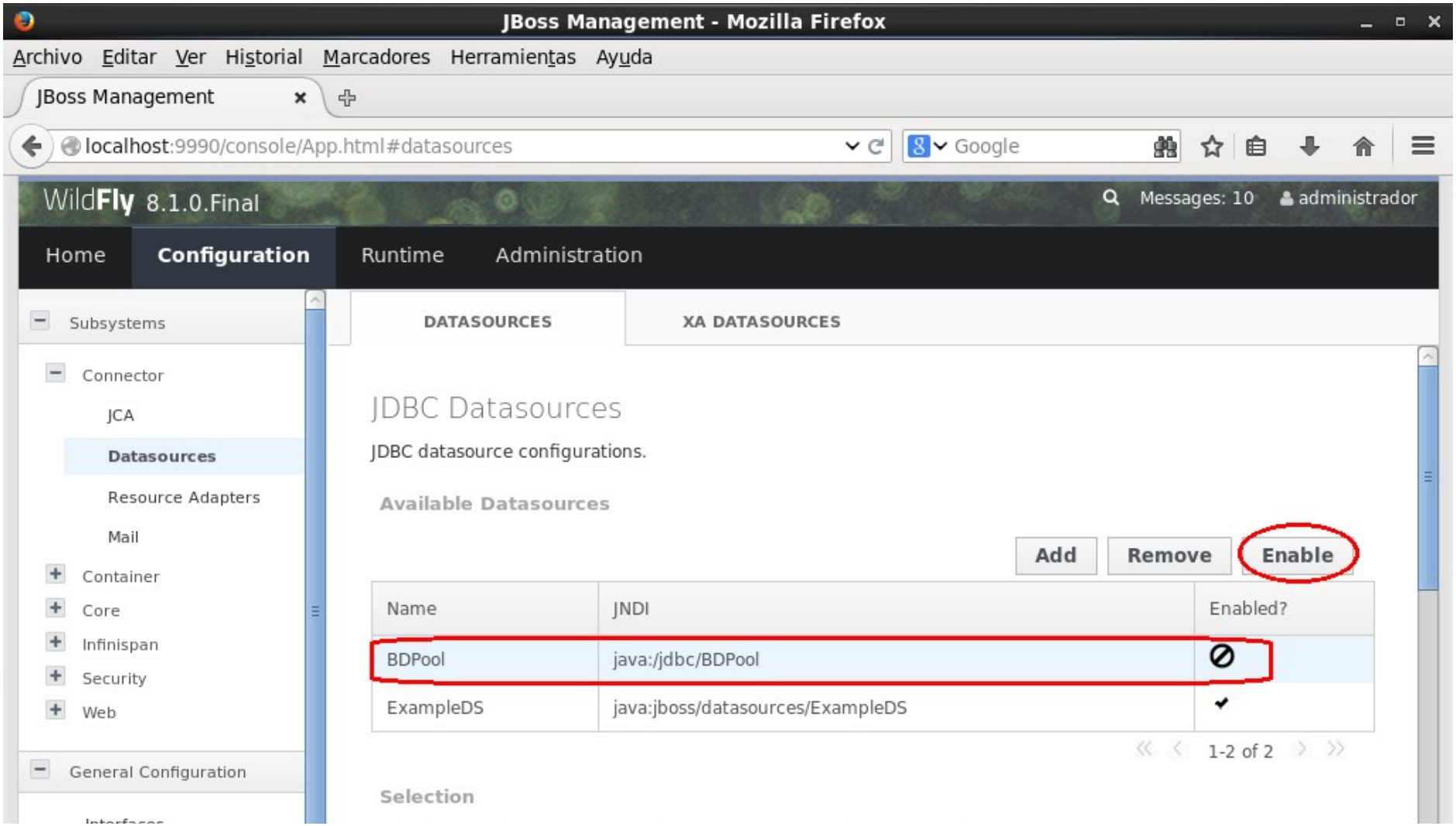
Crear un DataSource (IV)



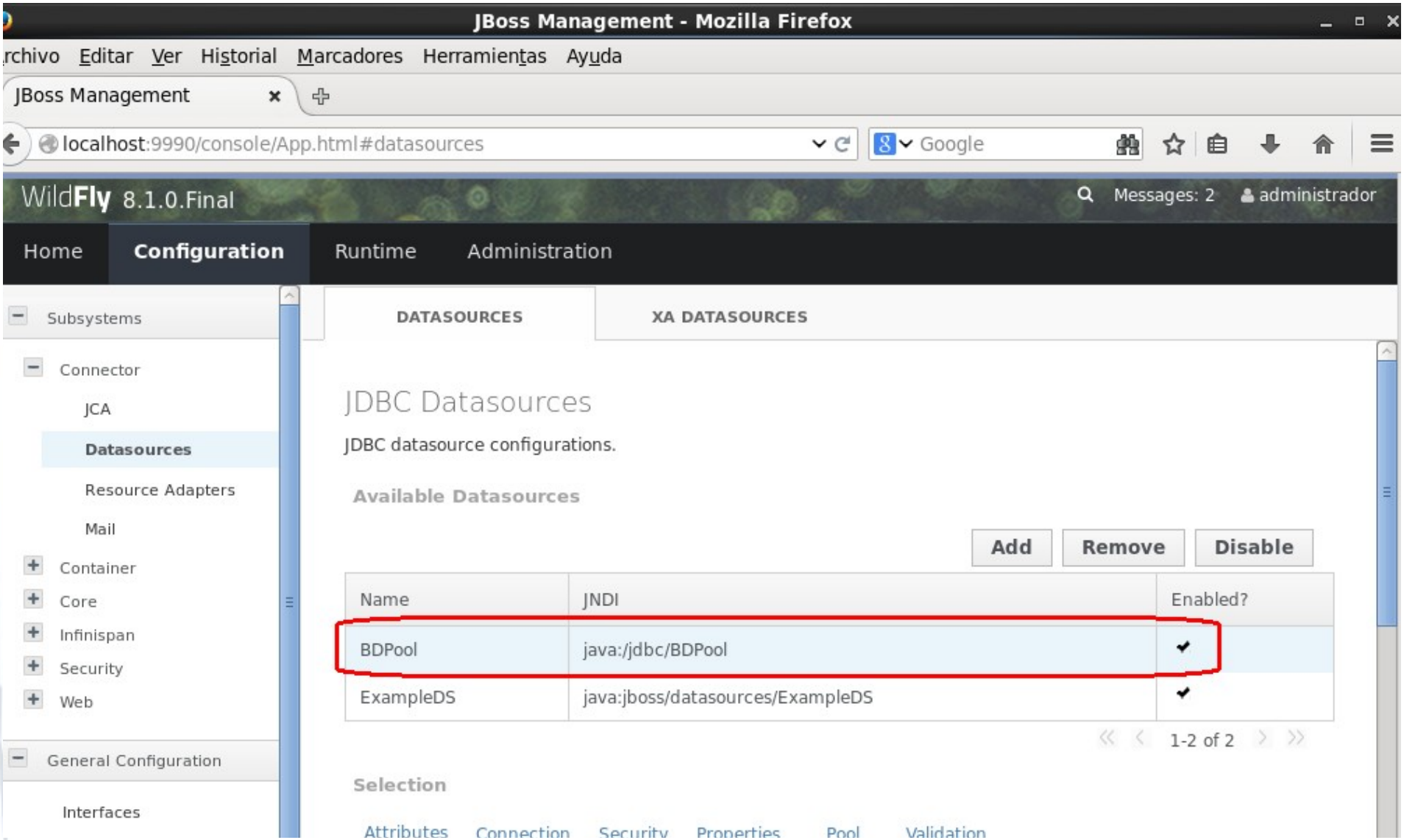
Crear un DataSource (V)



Crear un DataSource (VI)



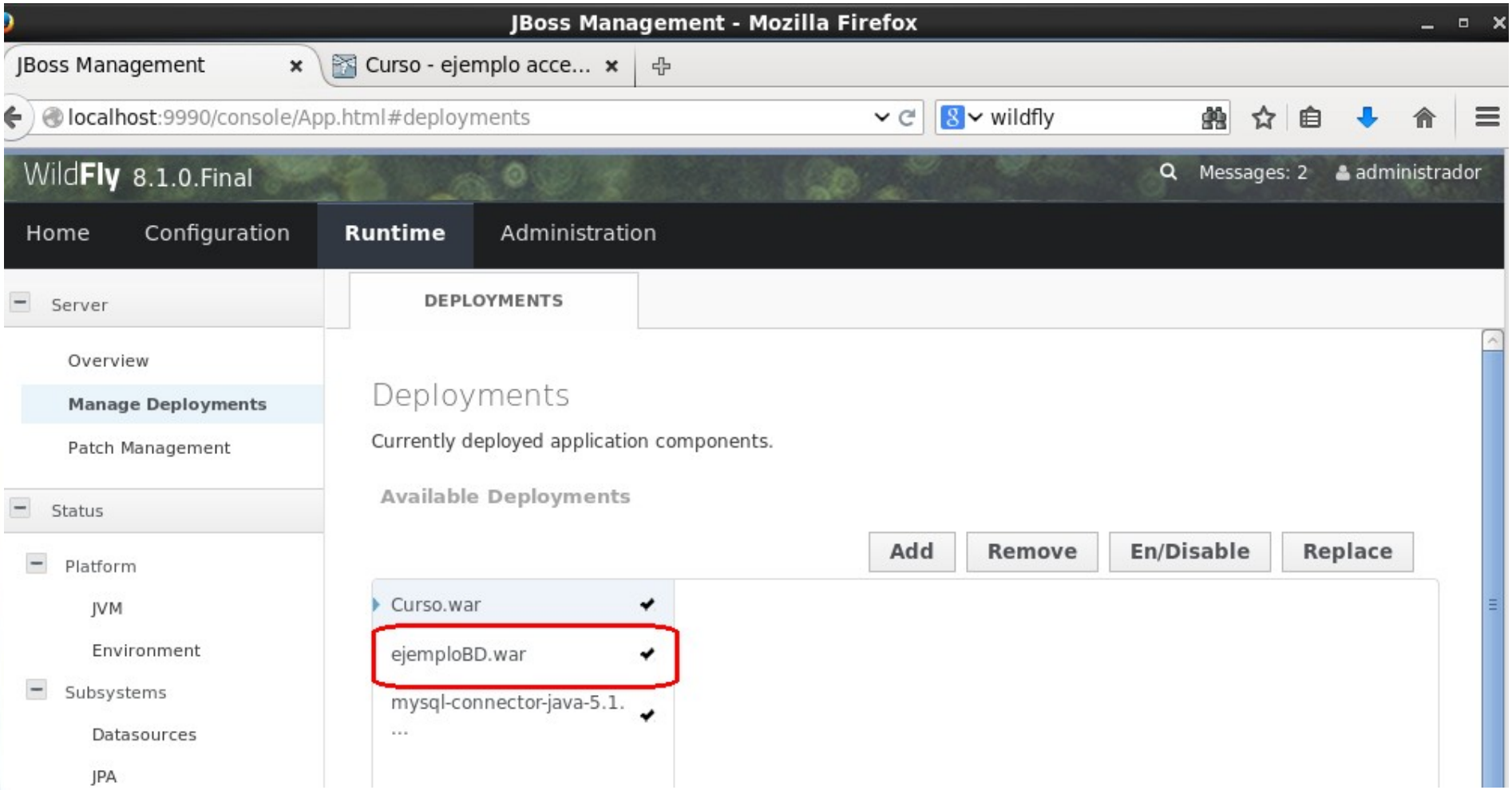
Crear un DataSource (VII)



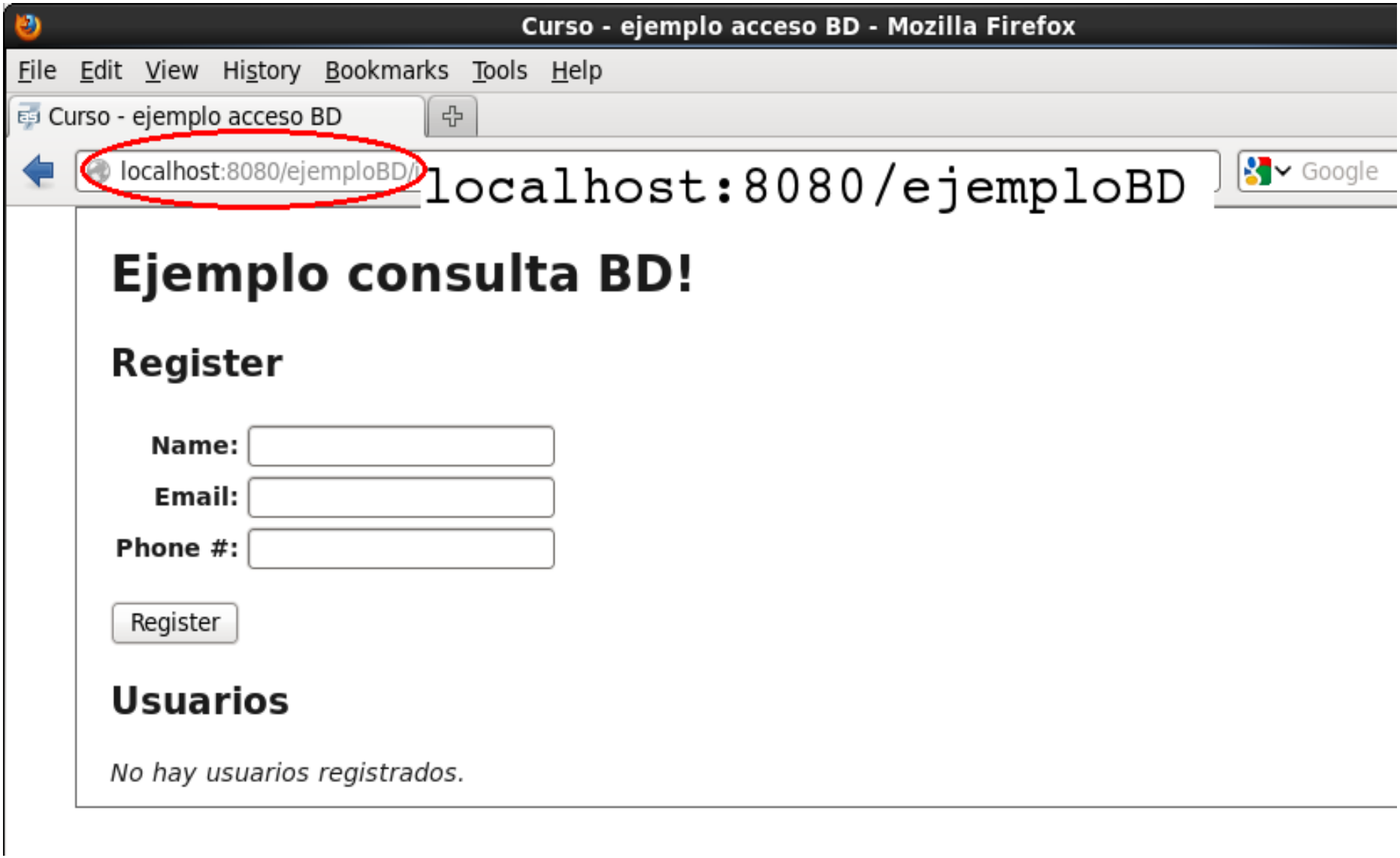
Acceso a MySQL desde WildFly (I)

- Llegados a este punto ya está configurada la conexión del WildFly con MySQL.
  - Es necesario reiniciar el servidor de WildFly para utilizar el DataSource.
- Para comprobar que funciona vamos a desplegar la misma aplicación utilizada para el Glassfish (EjemploBD.war).
- Esta aplicación utiliza el DataSource creado anteriormente (BDPool).
  - No es necesario definir los datos de usuario, contraseña y nombre de la base de datos en la aplicación, puesto que ya están definidos en el propio DataSource.

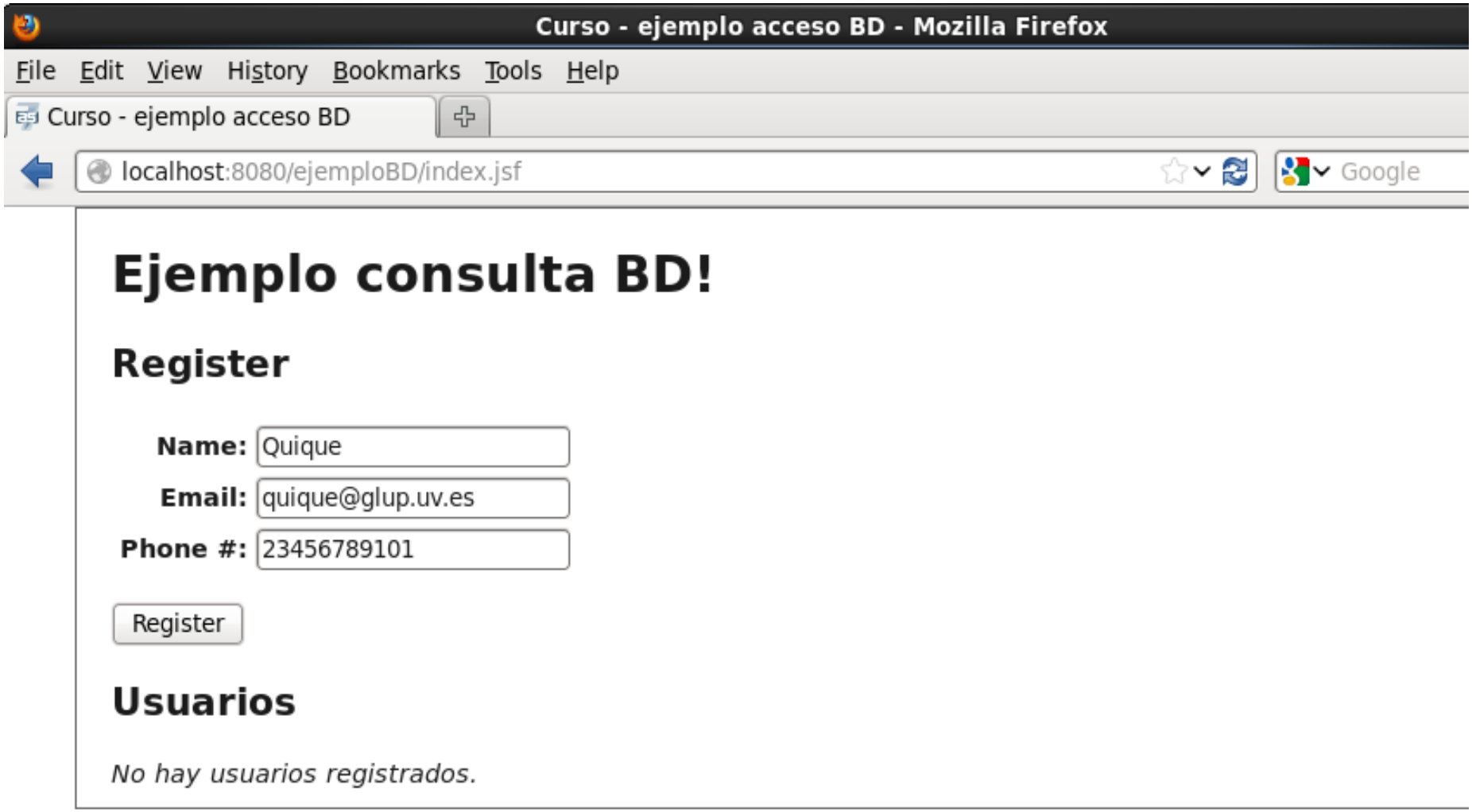
Acceso a MySQL desde WildFly (II)



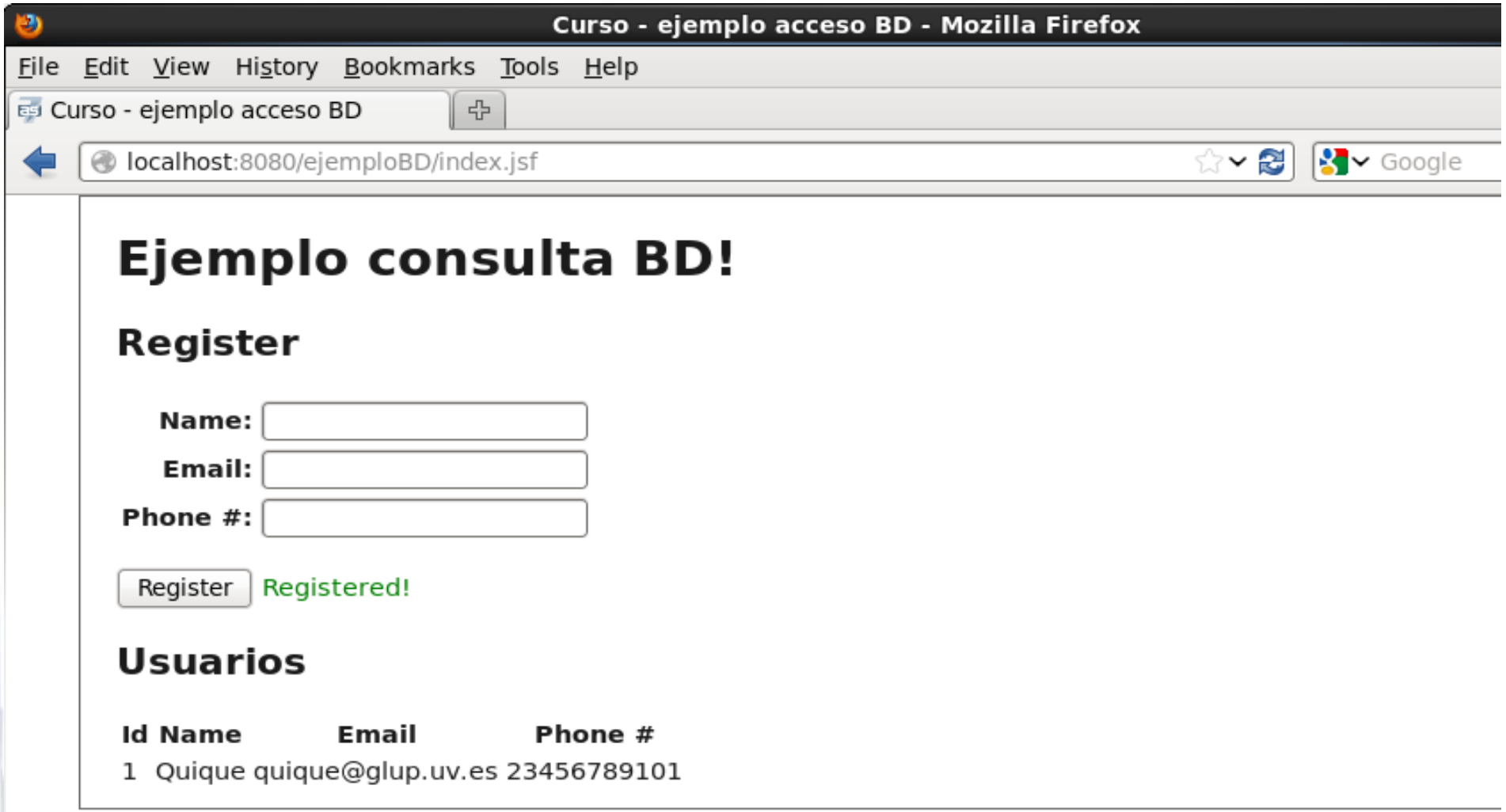
Acceso a MySQL desde WildFly (III)



Acceso a MySQL desde WildFly (IV)



Acceso a MySQL desde WildFly (V)

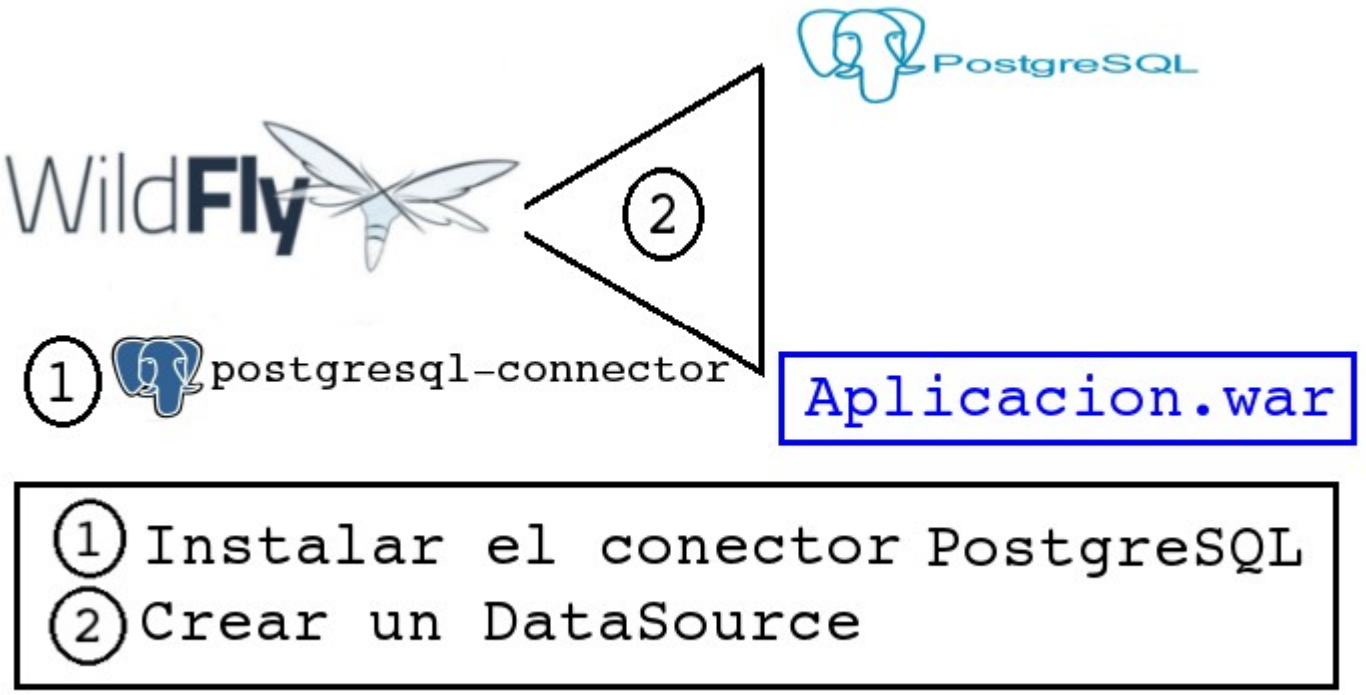


Acceso a MySQL desde WildFly (VI)

- Antes de realizar el acceso a PostgreSQL, eliminamos lo que hemos creado en el servidor de WildFly:
  - Eliminar la aplicación ejemploBD
  - Eliminar el Datasource BDPool

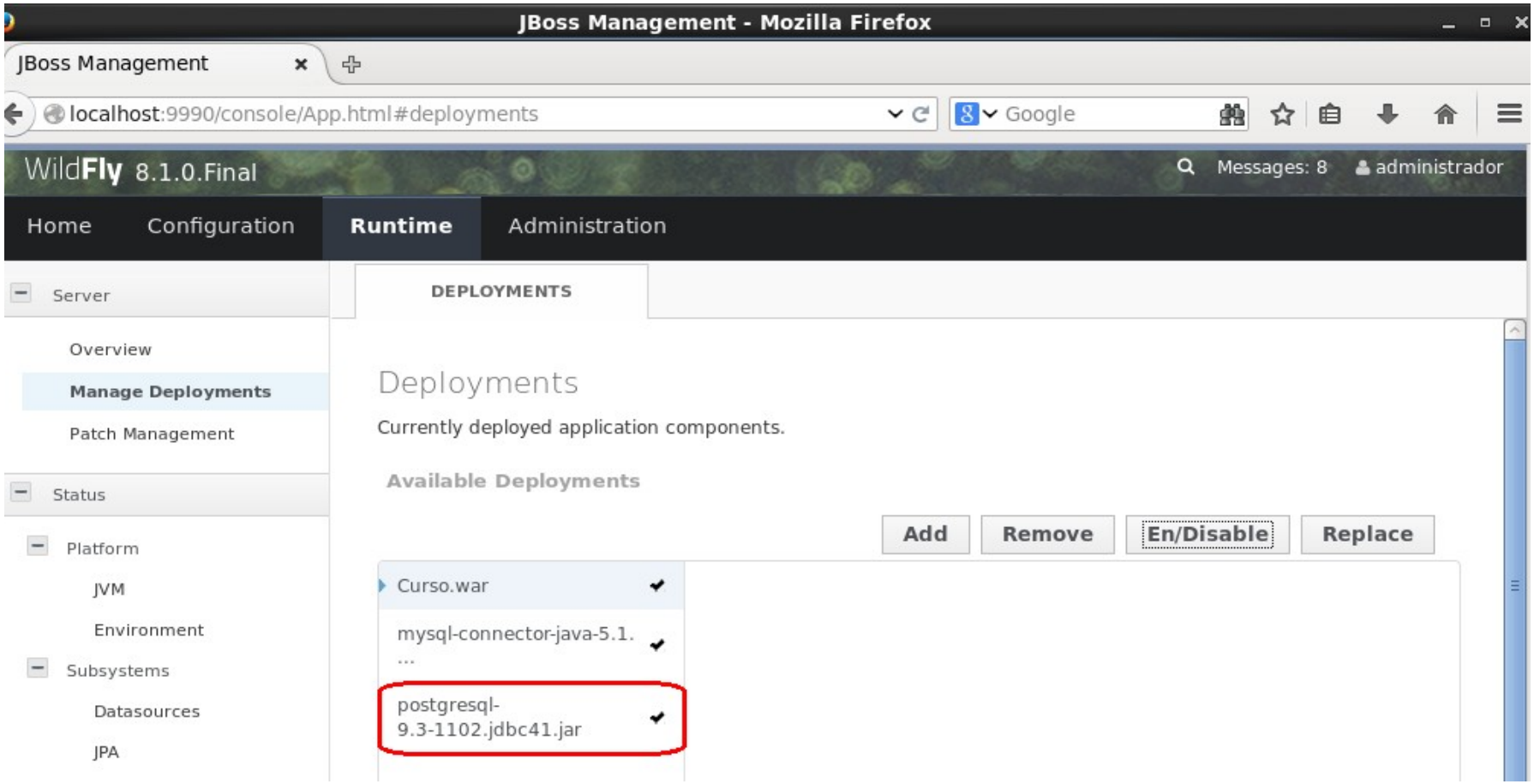
Conexión PostgreSQL - WildFly

- Para que la aplicación desplegada en WildFly acceda a una base de datos PostgreSQL, se deben seguir los mismos pasos que para MySQL:

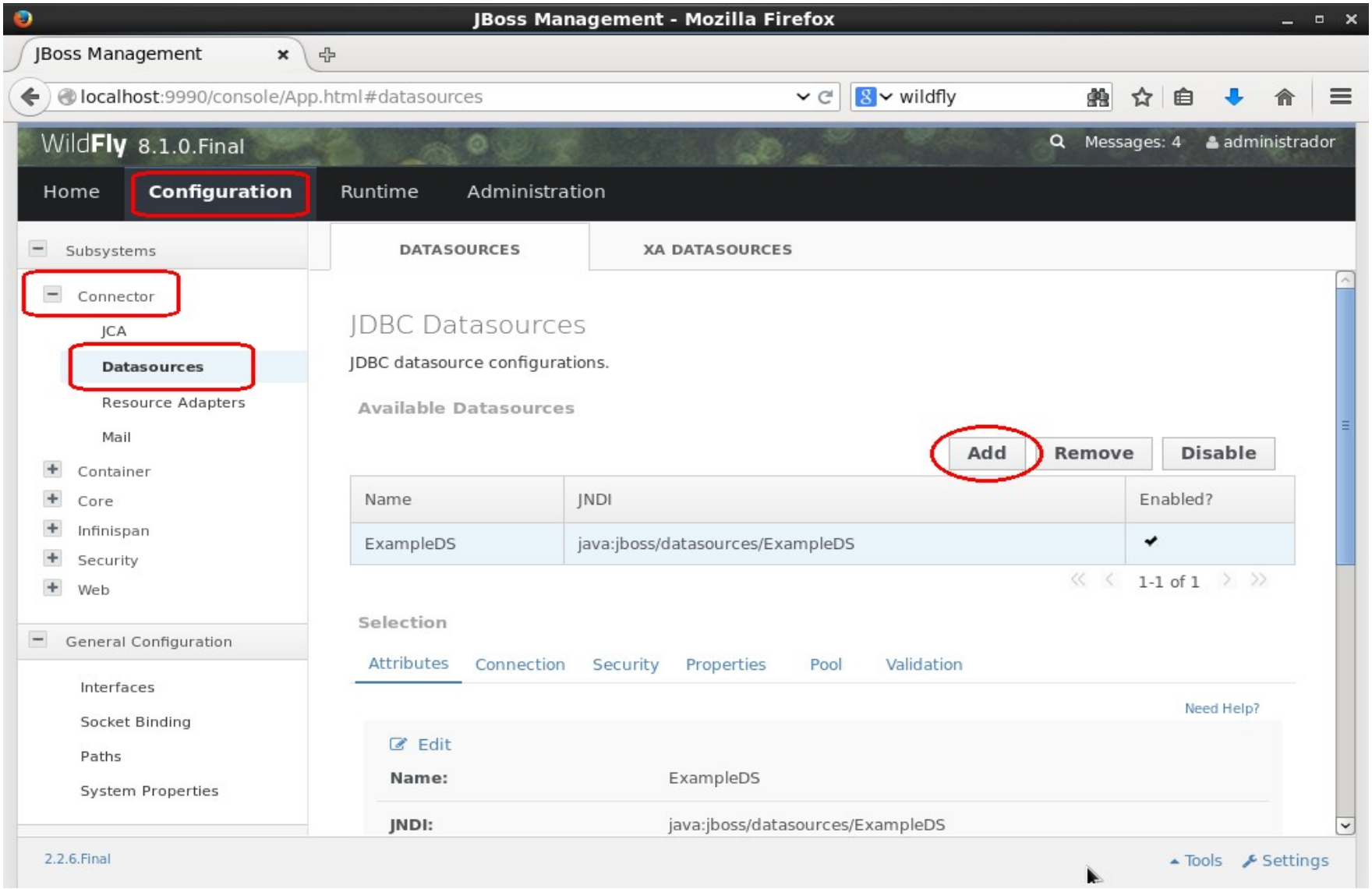


Instalar el conector de PostgreSQL

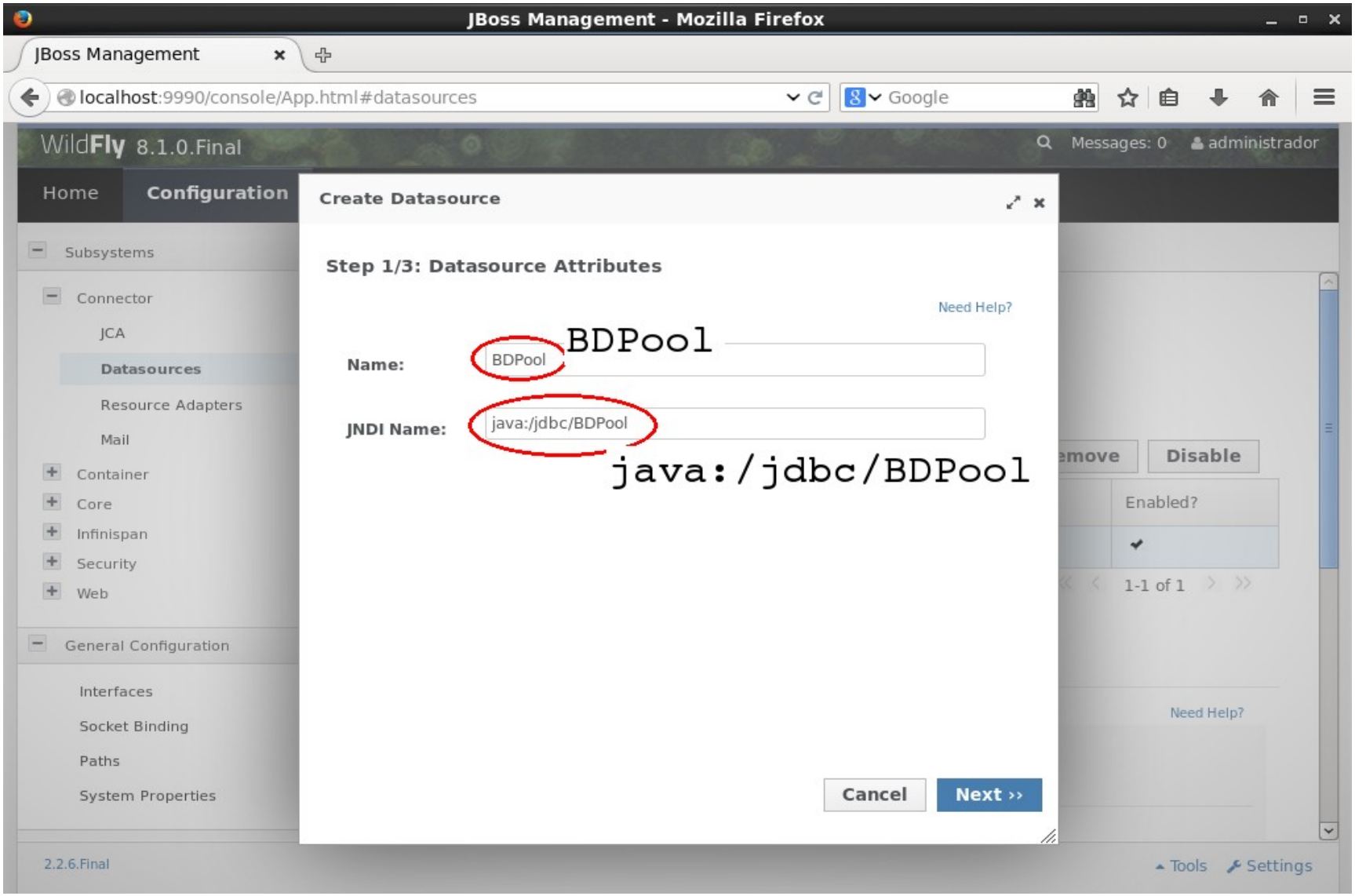
- Desplegar el **postgresql-<version>.jdbc41.jar** como cualquier aplicación de WildFly.



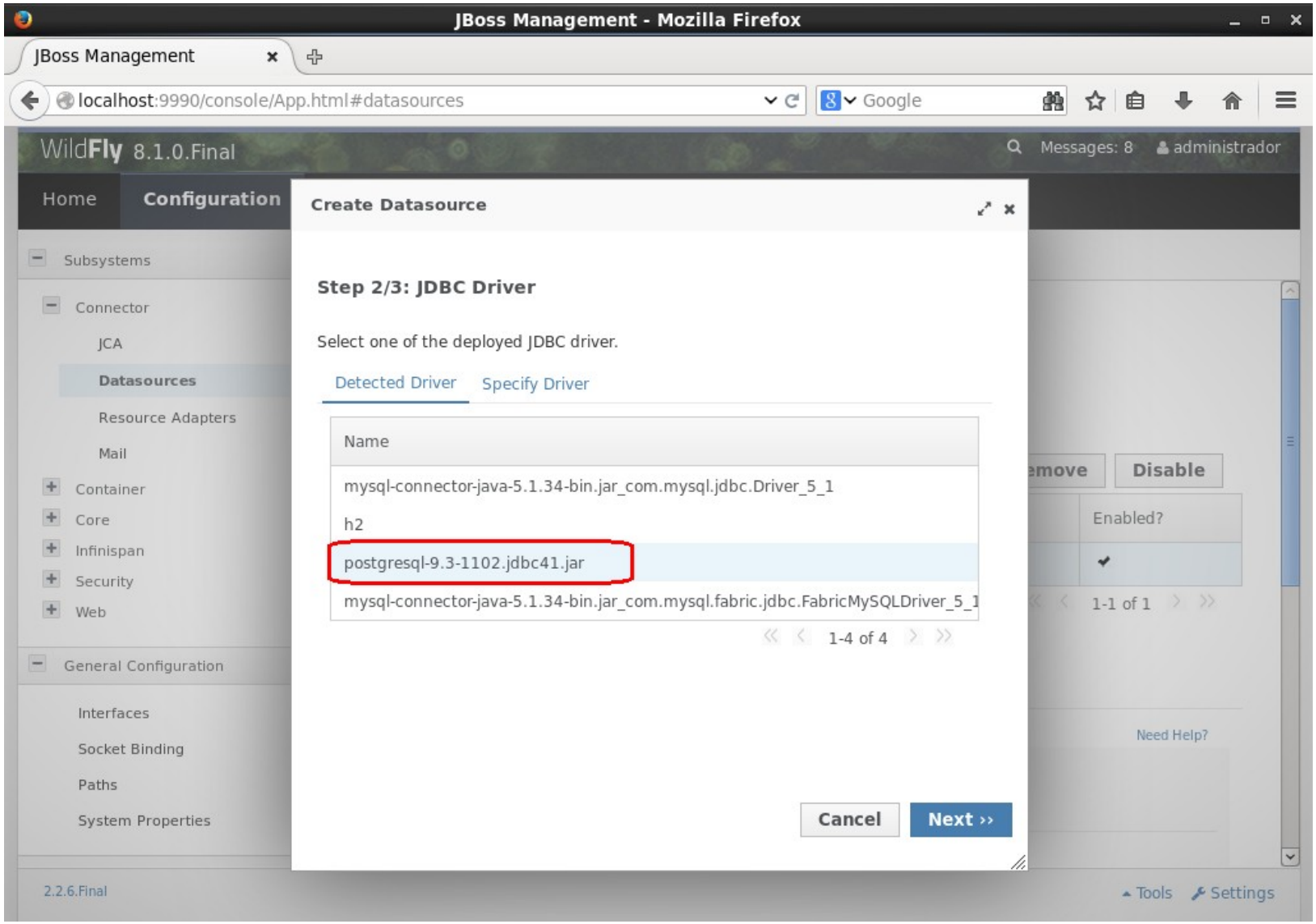
Crear un DataSource (I)



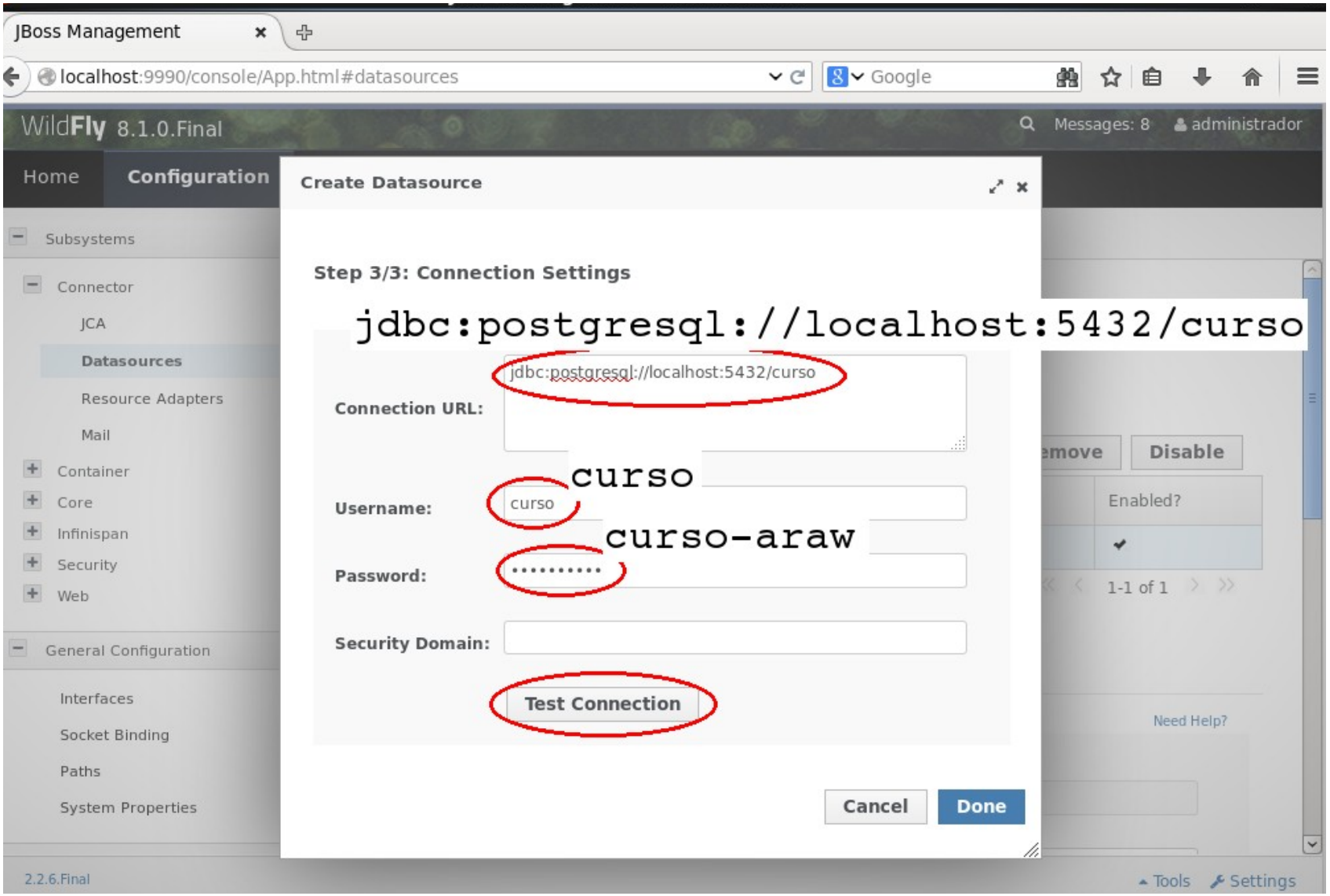
Crear un DataSource (II)



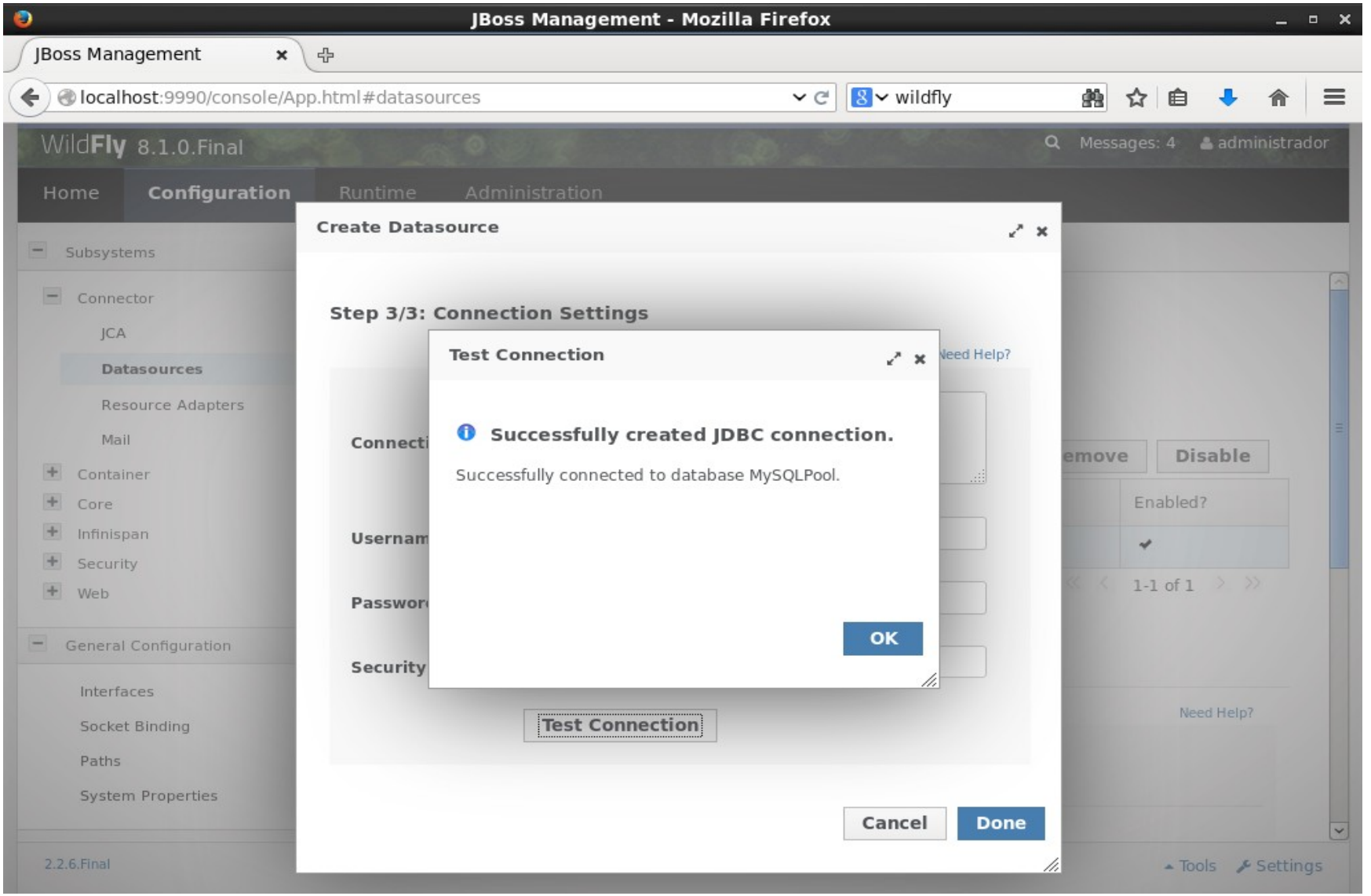
Crear un DataSource (III)



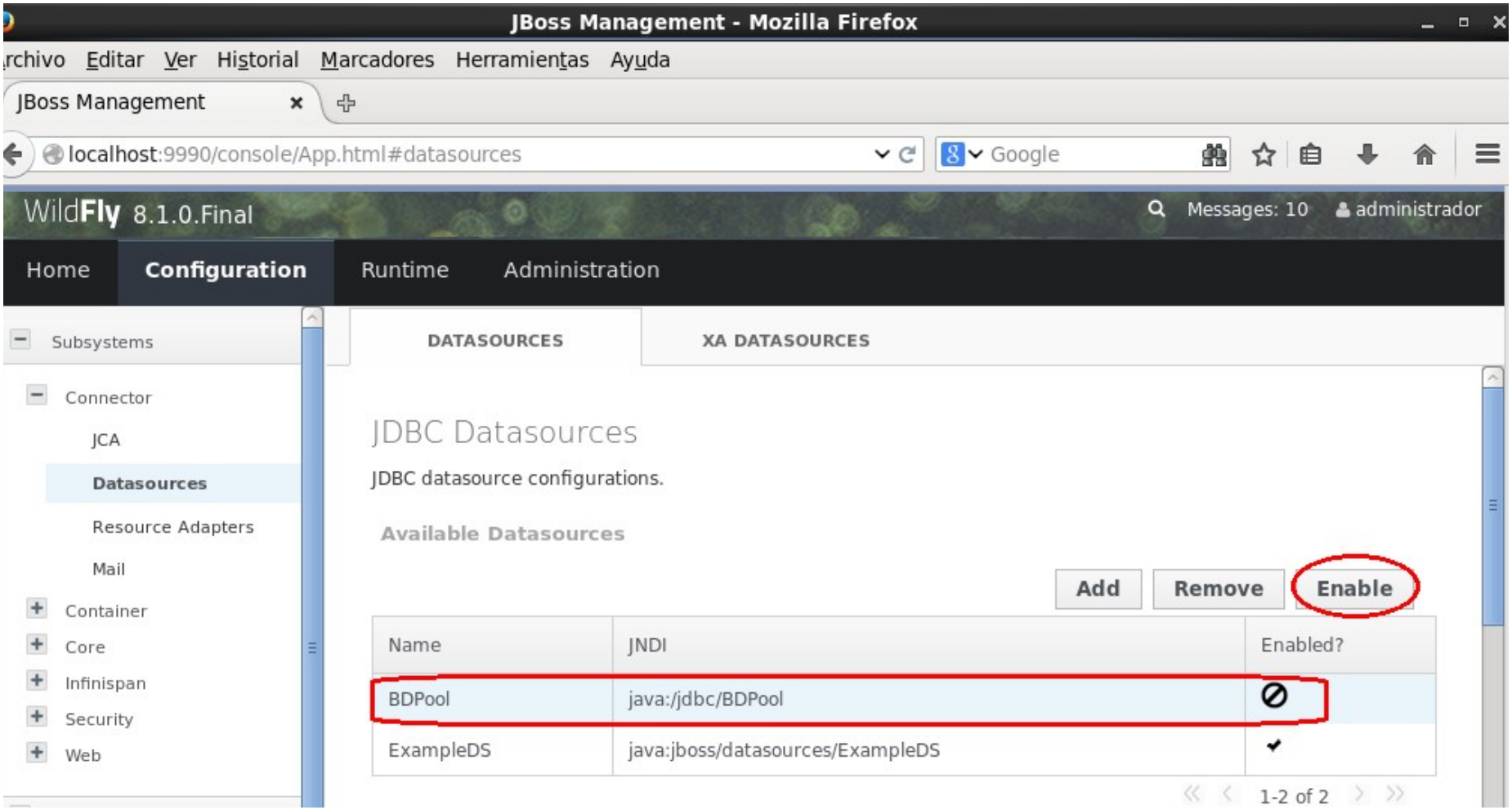
Crear un DataSource (IV)



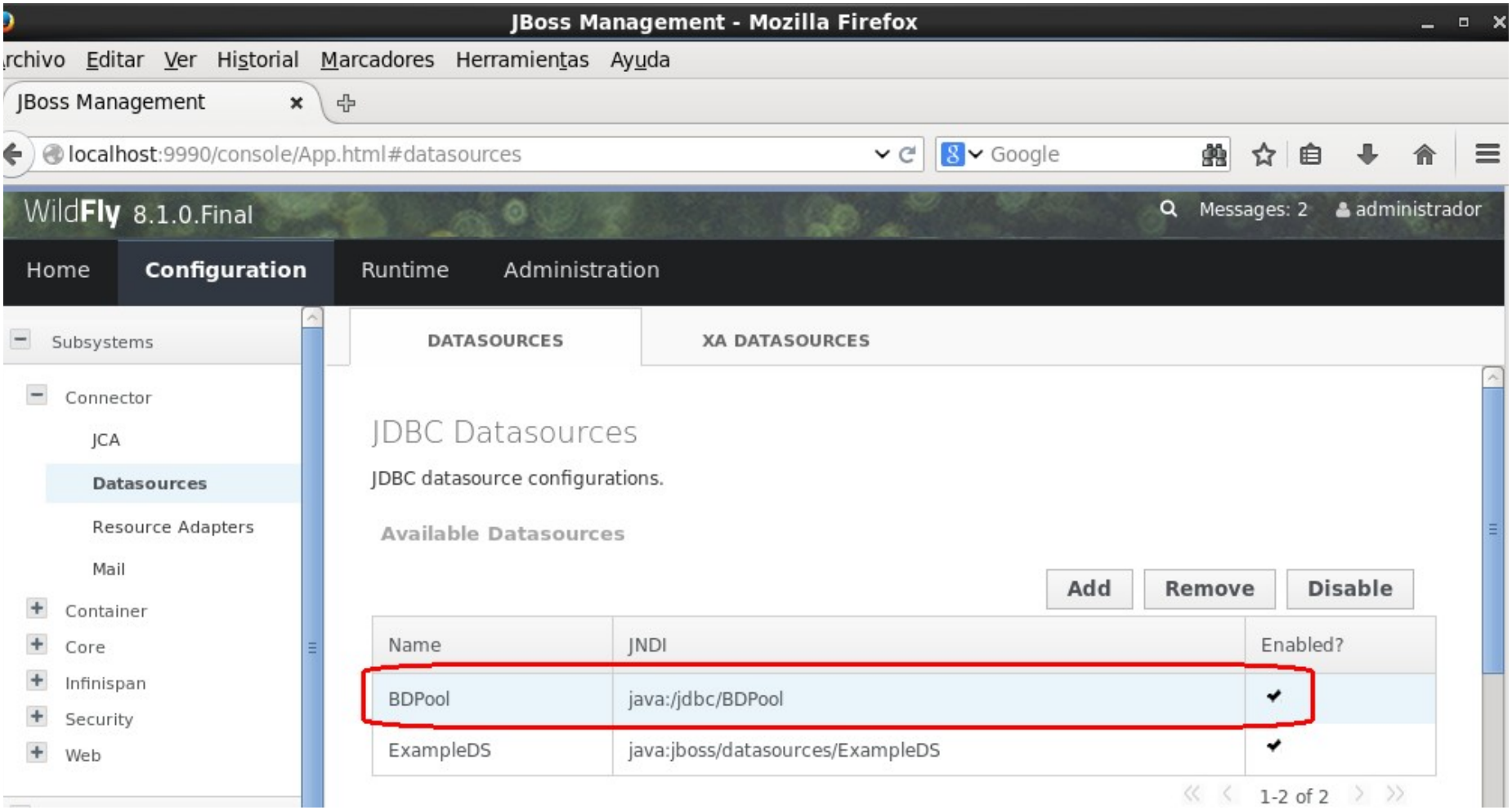
Crear un DataSource (V)



Crear un DataSource (VI)



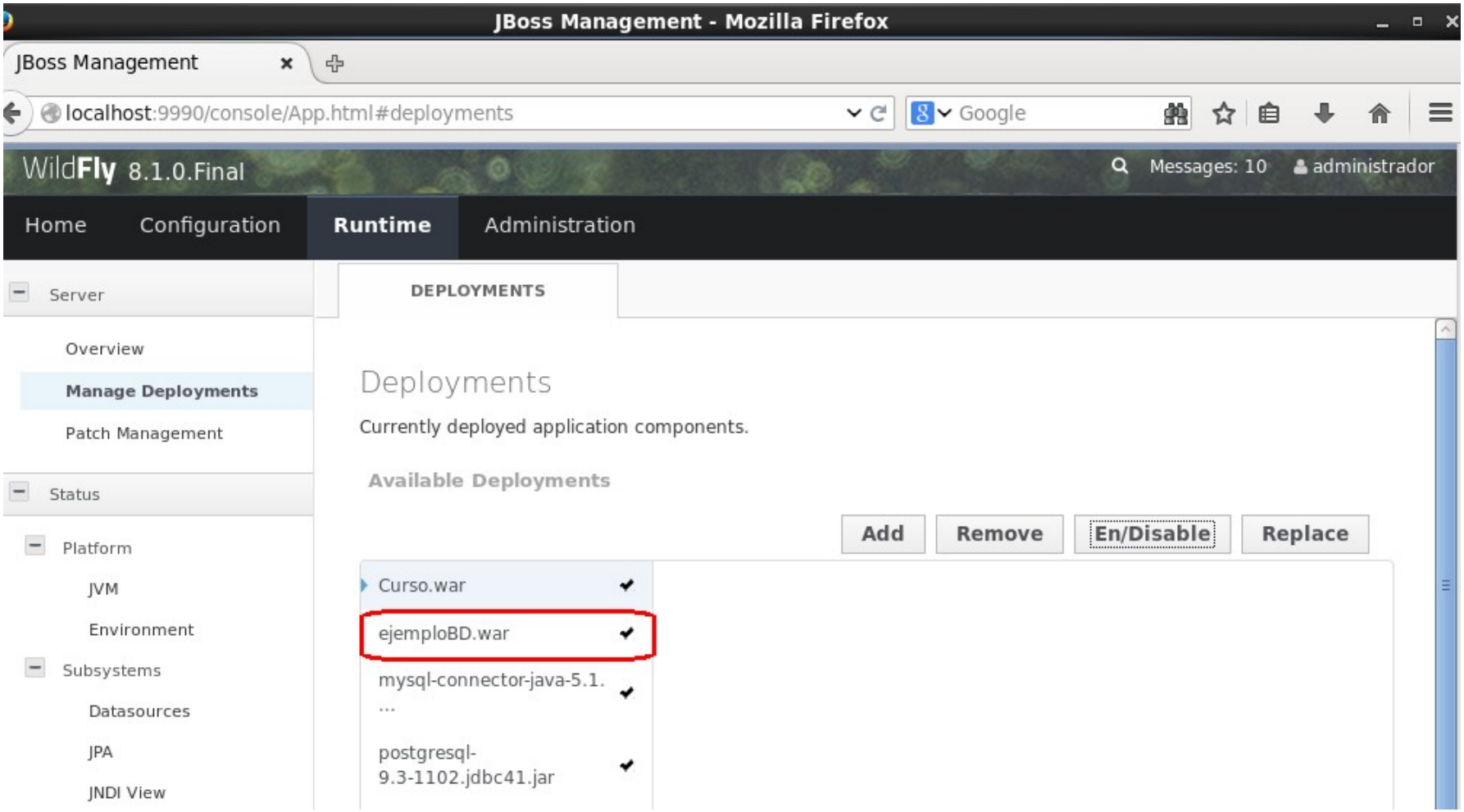
Crear un DataSource (VII)



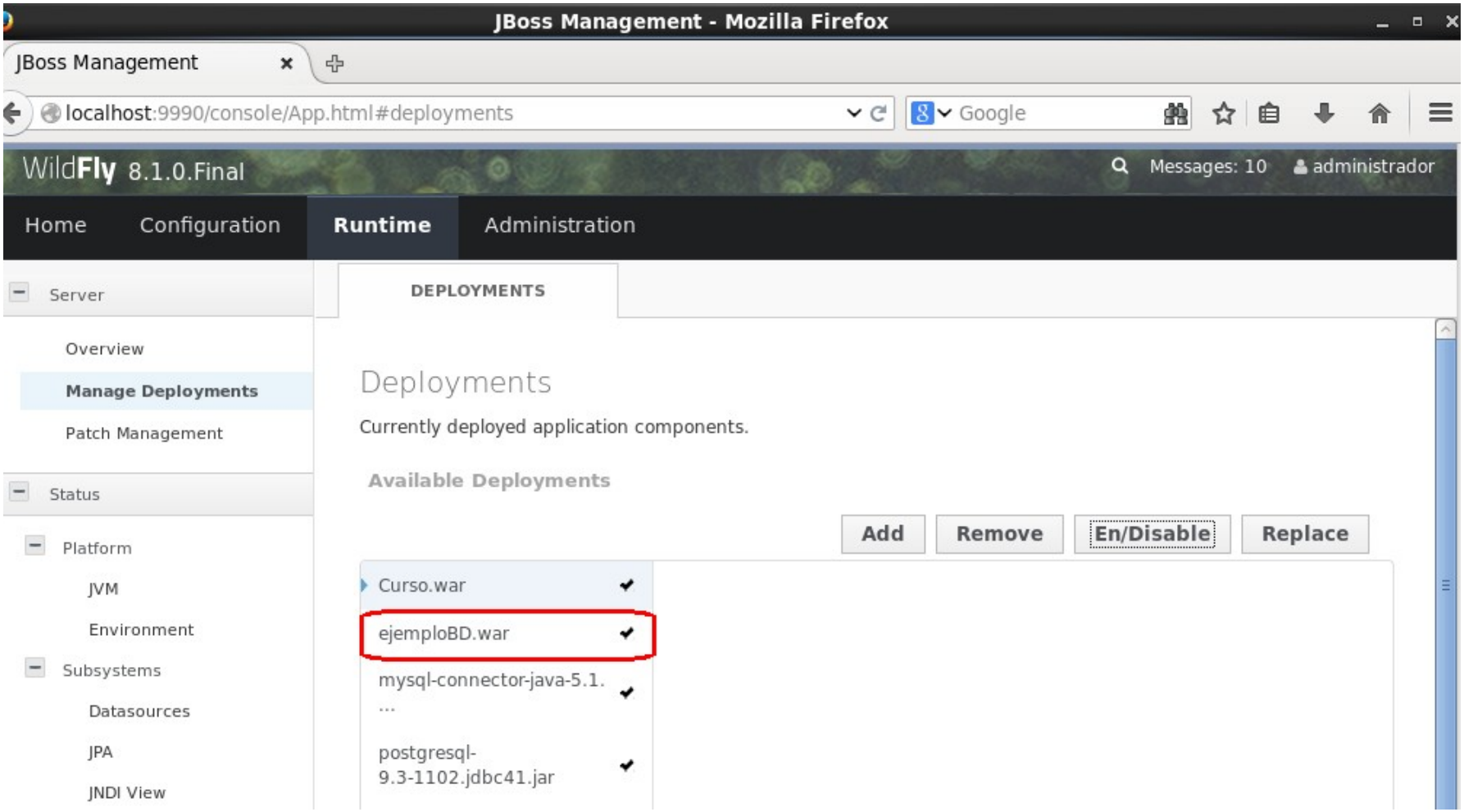
Acceso a PostgreSQL desde WildFly (I)

- Ya tenemos configurada la conexión del WildFly con PostgreSQL.
  - Es necesario reiniciar el servidor de WildFly para utilizar el DataSource.
- Para comprobar que funciona vamos a desplegar la aplicación EjemploBD.war.
- Esta aplicación utiliza el DataSource creado anteriormente (BDPool).
  - No es necesario definir los datos de usuario, contraseña y nombre de la base de datos en la aplicación, puesto que ya están definidos en el propio DataSource.

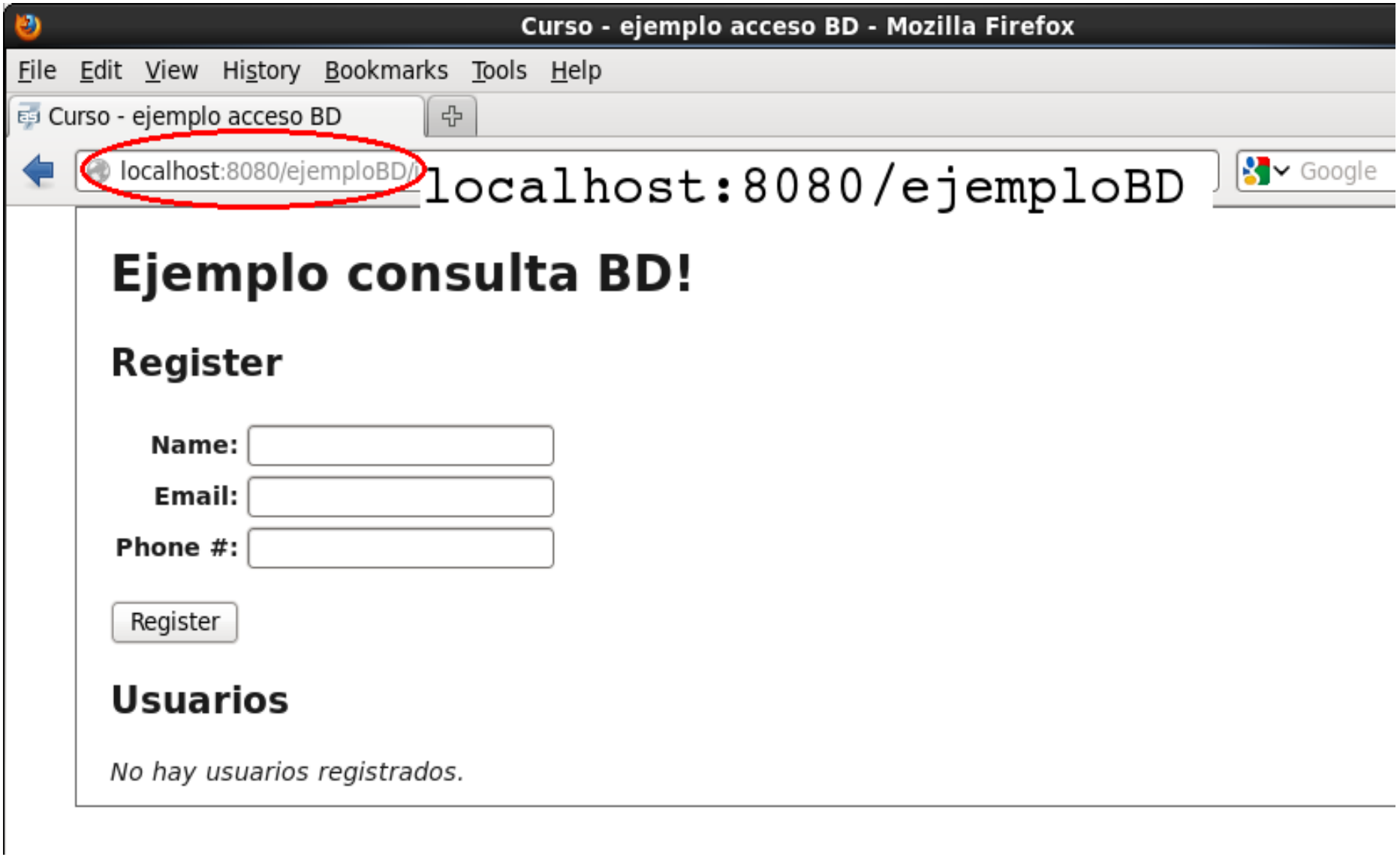
Acceso a PostgreSQL desde WildFly (II)



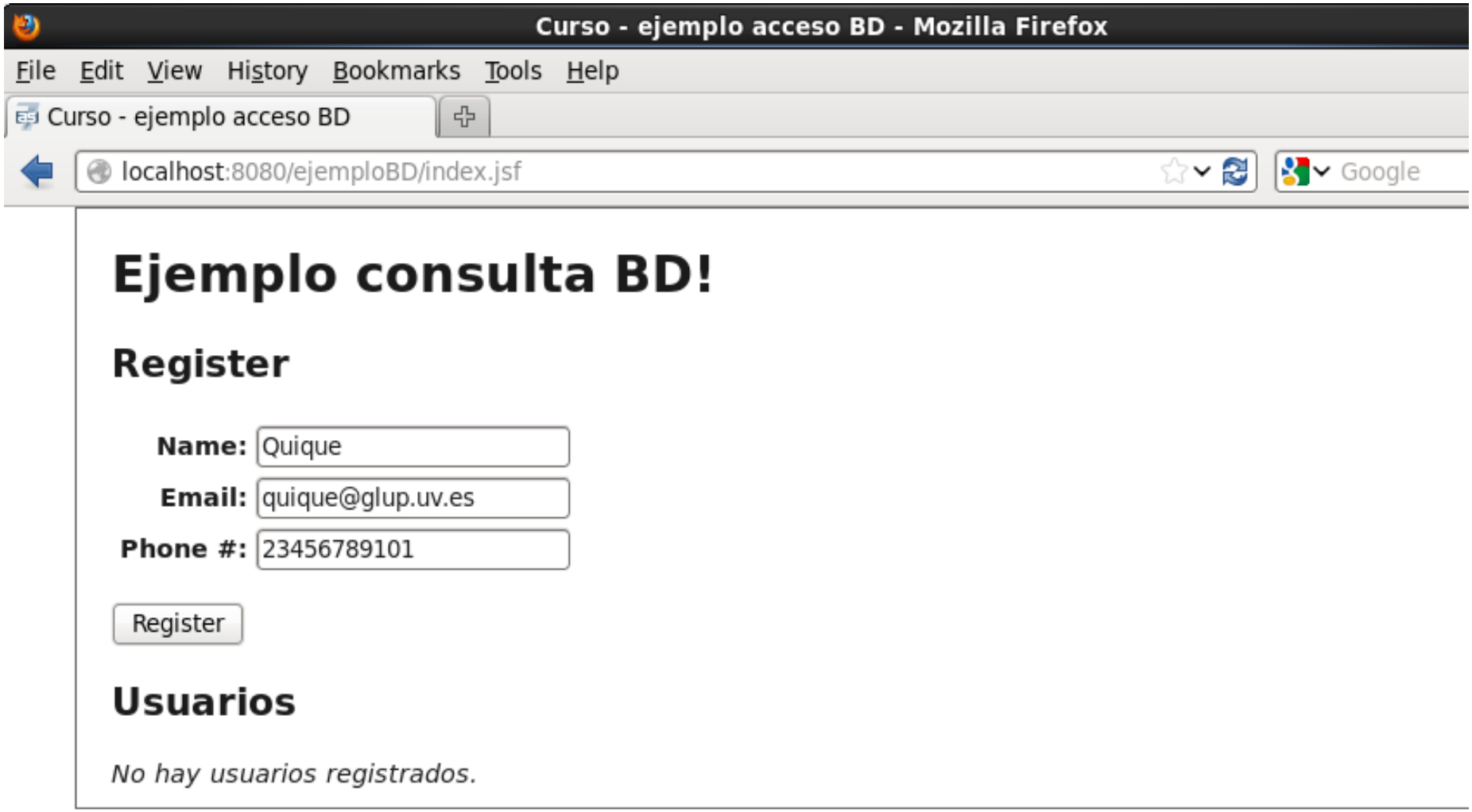
Acceso a PostgreSQL desde WildFly (II)



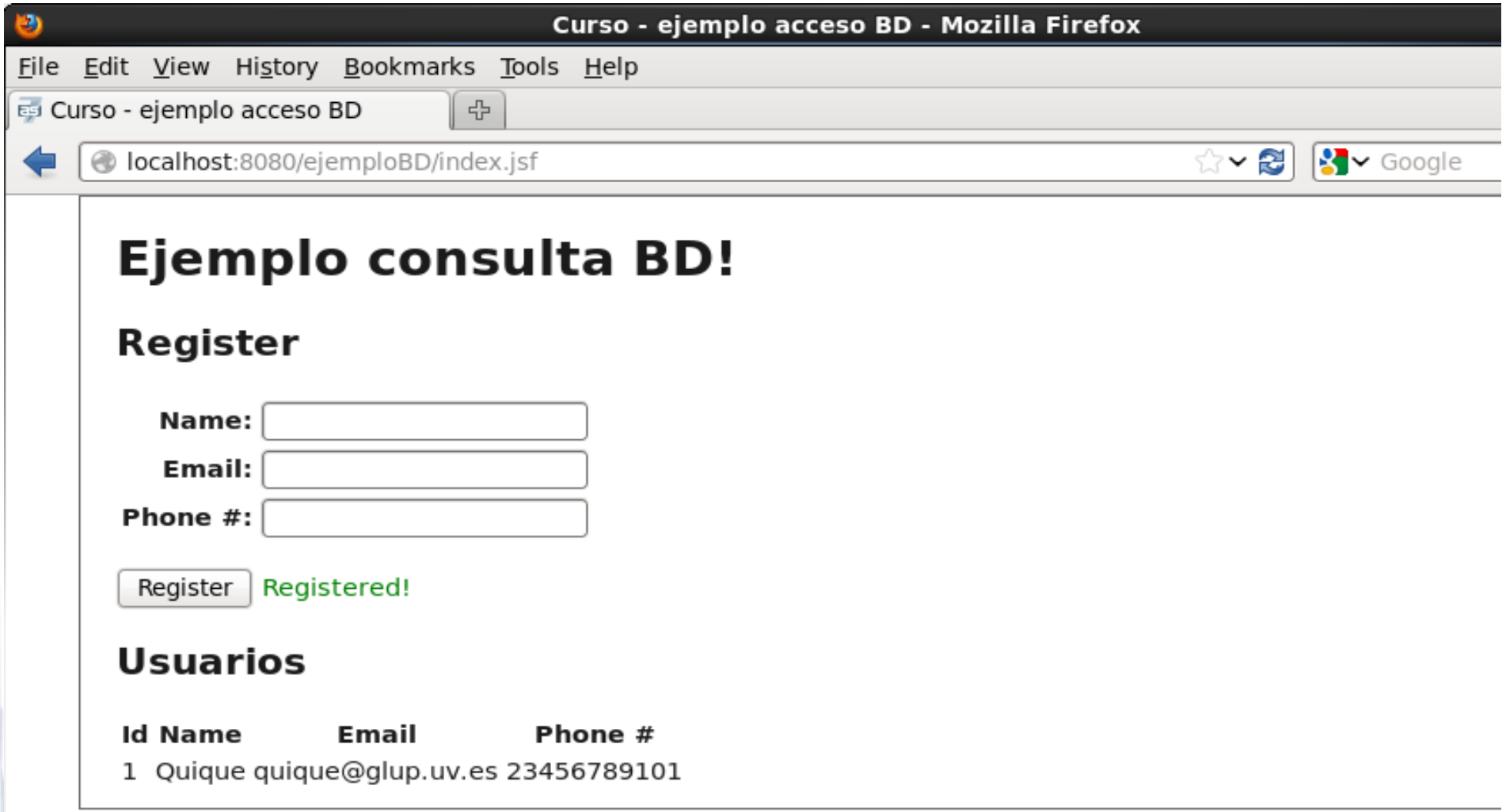
Acceso a PostgreSQL desde WildFly (III)



Acceso a PostgreSQL desde WildFly (IV)



Acceso a PostgreSQL desde WildFly (V)



Conclusión

- La aplicación que se debe conectar a la base de datos es independiente tanto del servidor de aplicaciones como de la base de datos a la que debe conectarse. Únicamente debe contener el nombre del “enlace” entre el servidor de aplicaciones y la base de datos.
  - En este caso, en la aplicación se indica que la conexión se realiza a través de BDPool, sin precisar si se accede a una base de datos MySQL o PostgreSQL y sin configurar los datos de acceso.