

Java EE

TEMA POOL DE CONEXIONES CON MYSQL, GLASSFISH Y NETBEANS

DOCENTE Ing. Eric Gustavo Coronel Castillo

BLOG <http://gcoronelc.blogspot.com>

1 SOFTWARE A UTILIZAR

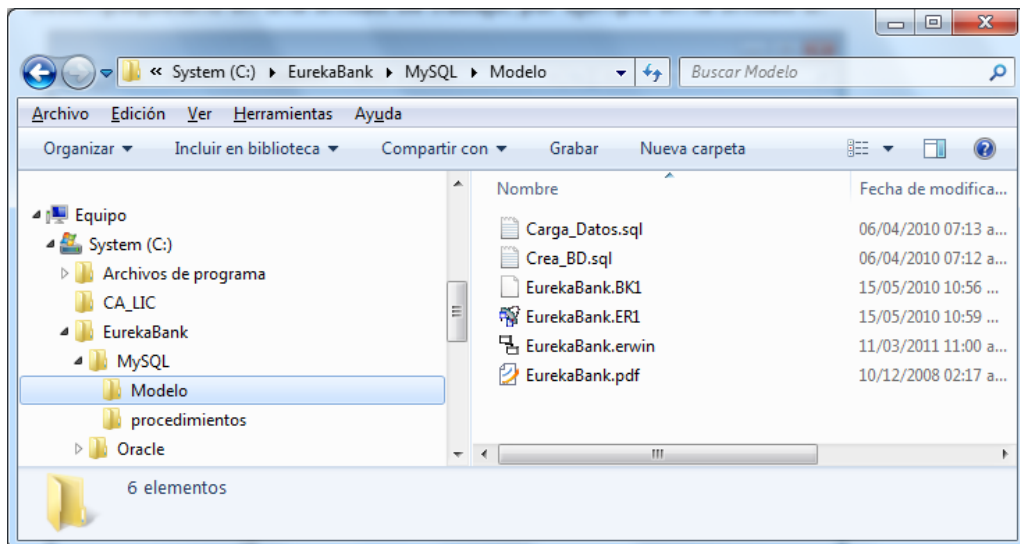
Para esta práctica de laboratorio asumiremos que tenemos instalado el siguiente software:

- Java JDK
- NetBeans
- GlassFish
- MySQL

2 CREACIÓN DE LA BASE DE DATOS DE PRUEBA

Es necesario tener una base de datos de prueba, en esta ocasión utilizaremos la base de datos EUREKABANK, que se puede obtener desde mi blog: <http://gcoronelc.blogspot.com>, los pasos a seguir para crear la base de datos son los siguientes:

1. Lo primero que se debe hacer es descargar el archivo EurekaBank.RAR y desempaquetarlo en una unidad de trabajo, por ejemplo en la unidad C.



En el directorio [EurekaBank\MySQLModelo](#) encuentra el script que debemos ejecutar para crear la base de datos [EurekaBank](#).

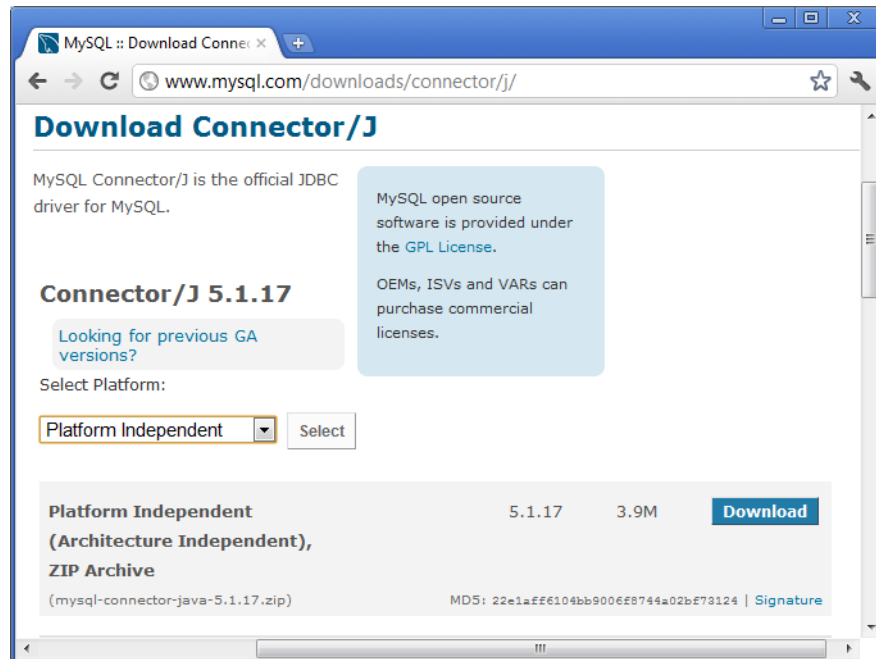
- [Crea_BD.sql](#) crea la base de datos y sus respectivas tablas.
 - [Carga_Datos.sql](#) se encarga de cargar datos de prueba en sus respectivas tablas.
2. Cargar la consola de [MySQL](#) y conectarse con usuario [root](#).
 3. Utilizando el comando [source](#) proceda a ejecutar los dos scripts anteriores.

```
mysql> source C:\EurekaBank\MySQL\Modelo\Crea_BD.sql [Enter]
mysql> source C:\EurekaBank\MySQL\Modelo\Carga_Datos.sql [Enter]
```

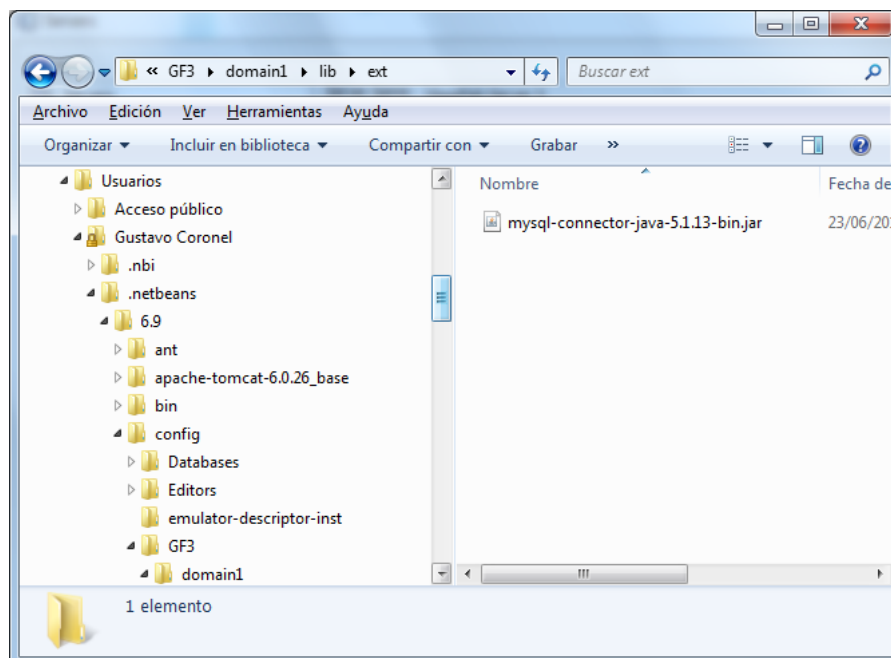
3 DRIVER DE BASE DE DATOS

Es necesario que el servidor (GlassFish) tenga el driver JDBC para MySQL, por lo tanto primero que se debe realizar es descargar el driver JDBC del sitio oficial de MySQL:

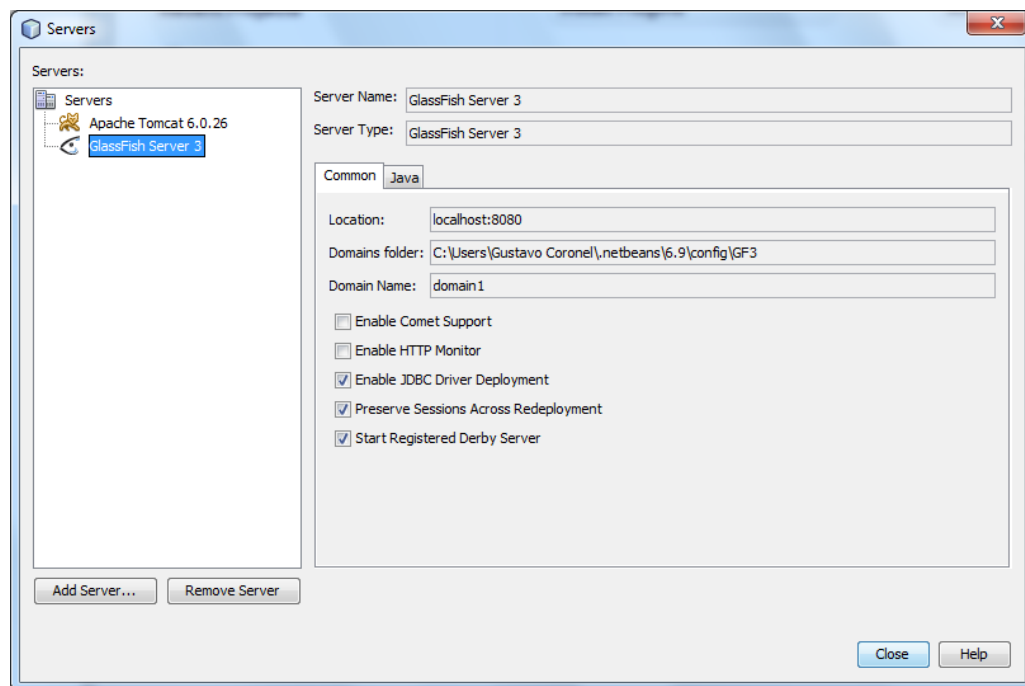
<http://www.mysql.com/downloads/connector/j/>



Después de haber descargado el driver, se debe copiar en el directorio **lib\ext** de GlasFish.

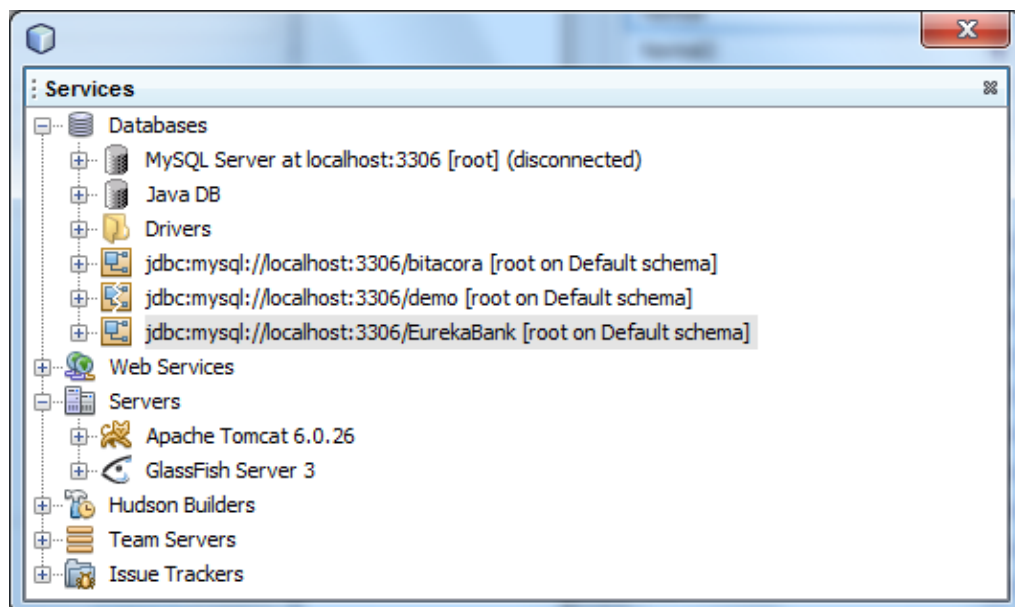


Para averiguar la dirección donde está instalado GlassFish se debe cargar la venta de servidores (Tools\Servers), tal como se ilustra a continuación:



4 CONEXIÓN DE NETBEANS CON EUREKABANK

Registrar una conexión con la base de datos **EUREKABANK** desde la carpeta **Database** en la ventana de servicios, tal como se ilustra en la siguiente figura:



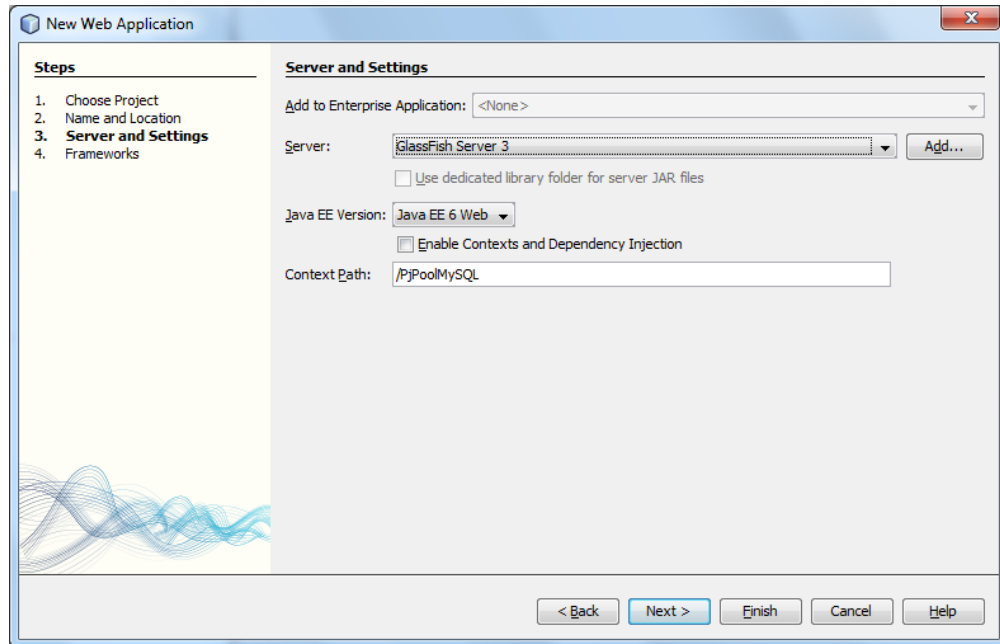
5 PROYECTO CON NETBEANS

El siguiente procedimiento es válido para las versiones 2 y 3 de GlassFish.

5.1 Creación de un nuevo proyecto

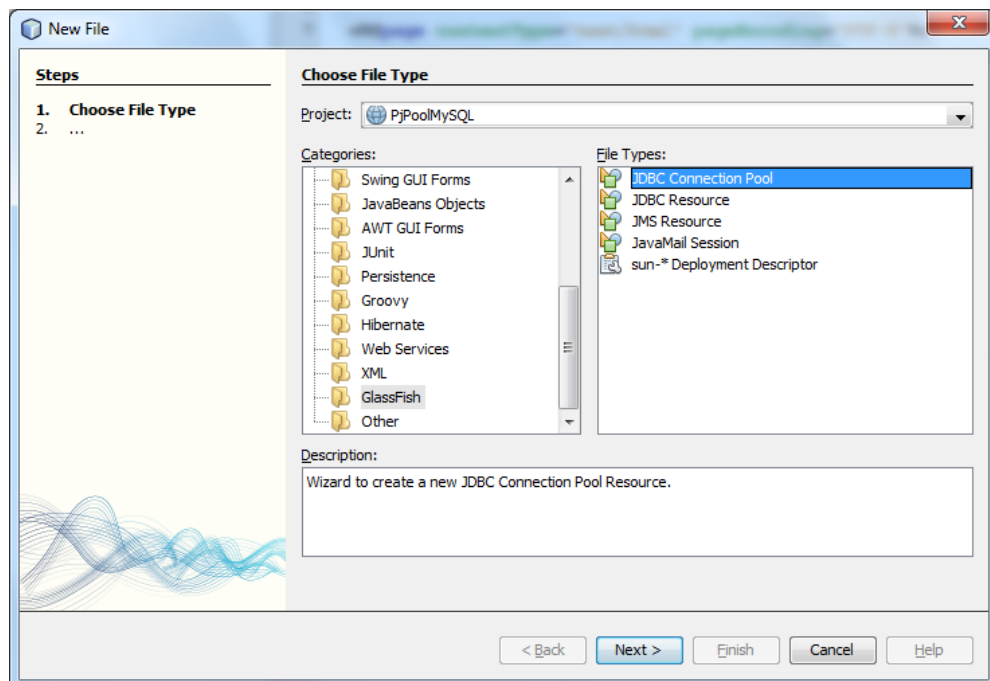
Se debe crear un nuevo proyecto con NetBeans utilizando las siguientes especificaciones:

- Tipo de proyecto Java Web
- Nombre del proyecto PjPoolMySQL
- Servidor GlassFish
- Plataforma Java EE 6 Web



5.2 Configurar el pool de conexiones

El siguiente grafico muestra el tipo de archivo que debe seleccionar para configurar el pool de conexiones.



El siguiente grafico muestra la opción donde se le asigna el nombre al pool y se selecciona la conexión con la base de datos EurekaBank.

The screenshot shows the 'New JDBC Connection Pool' dialog box, specifically the 'Choose Database Connection' step. The 'Steps' pane on the left lists: 1. Choose ..., 2. Choose Database Connection (highlighted), 3. Add Connection Pool Properties, and 4. Add Connection Pool Optional Properties. The main area contains instructions: 'Provide configuration information for the JDBC Connection Pool. Either choose an existing database connection to extract information, or enter the configuration information. Fields with an * mark are required.' Below this, the 'JDBC Connection Pool Name:' field is set to 'eurekaPool'. The 'Extract from Existing Connection:' radio button is selected, and the dropdown menu shows 'jdbc:mysql://localhost:3306/EurekaBank [root on Default schema]'. The 'New Configuration using Database:' radio button is unselected, with a dropdown showing '< Select from the list >'. The 'XA (Global Transaction)' checkbox is unchecked. At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Steps

1. Choose ...
2. **Choose Database Connection**
3. Add Connection Pool Properties
4. Add Connection Pool Optional Properties

Choose Database Connection

Provide configuration information for the JDBC Connection Pool.
Either choose an existing database connection to extract information, or enter the configuration information.
Fields with an * mark are required.

JDBC Connection Pool Name: * eurekaPool

☒ Extract from Existing Connection:

jdbc:mysql://localhost:3306/EurekaBank [root on Default schema]

☐ New Configuration using Database:

< Select from the list >

☐ XA (Global Transaction)

< Back Next > Finish Cancel Help

El siguiente grafico muestra los parámetros de conexión con la base de datos EurekaBank obtenidos de la conexión seleccionada en el paso anterior.

The screenshot shows the 'New JDBC Connection Pool' dialog box, specifically the 'Add Connection Pool Properties' step. The 'Steps' pane on the left lists: 1. Choose ..., 2. Choose Database Connection, 3. Add Connection Pool Properties (highlighted), and 4. Add Connection Pool Optional Properties. The main area contains instructions: 'Enter the Datasource Classname, URL, and User to continue. Hit the Enter key to save values in the Properties table.' Below this, the 'Datasource Classname:' field is set to 'com.mysql.jdbc.jdbc2.optional.MysqlDataSource'. The 'Resource Type:' dropdown menu is set to 'javax.sql.DataSource'. The 'Description:' field is empty. Below the 'Properties:' label is a table with two columns: 'Name' and 'Value'. The table contains three rows: 'URL' with value 'jdbc:mysql://localhost:3306/EurekaBank', 'User' with value 'root', and 'Password' with value 'mysql'. To the right of the table are 'Add' and 'Remove' buttons. At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Steps

1. Choose ...
2. Choose Database Connection
3. **Add Connection Pool Properties**
4. Add Connection Pool Optional Properties

Add Connection Pool Properties

Enter the Datasource Classname, URL, and User to continue.
Hit the Enter key to save values in the Properties table.

Datasource Classname: com.mysql.jdbc.jdbc2.optional.MysqlDataSource

Resource Type: javax.sql.DataSource

Description:

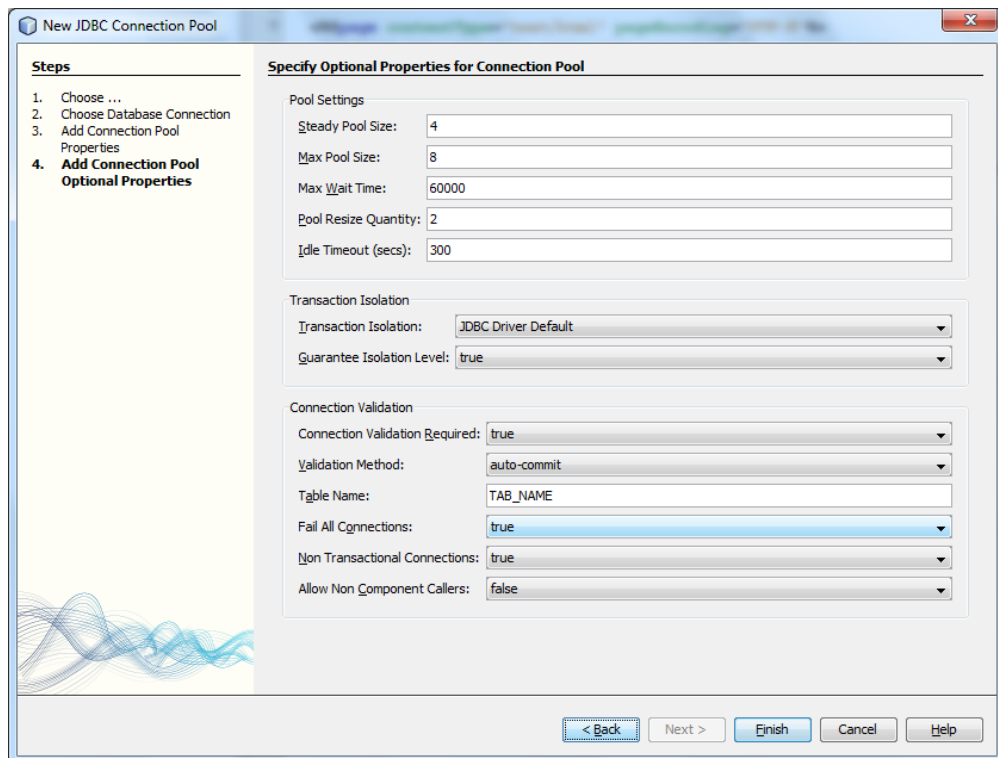
Properties:

Name	Value
URL	jdbc:mysql://localhost:3306/EurekaBank
User	root
Password	mysql

Add Remove

< Back Next > Finish Cancel Help

El siguiente grafico muestra datos adicionales del pool.

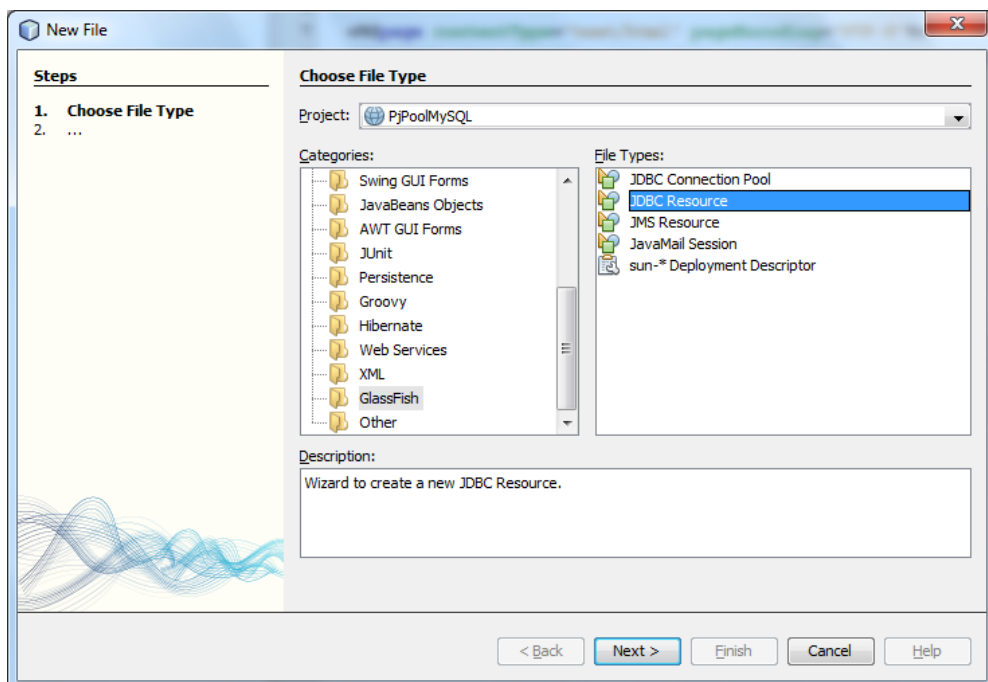


Finalmente se debe hacer ejecutar el botón **Finish** para terminar con la configuración del pool.

5.3 Configuración del recurso JDBC

Para poder utilizar el pool es necesario configurar el recurso JDBC.

El siguiente grafico el tipo de archivo que debemos crear para configurar el recurso JDBC.



El siguiente grafico muestra la configuración del recurso JDBC.

Steps

1. Choose ...
2. **General Attributes - JDBC Resource**
3. Properties

General Attributes

Provide configuration information for the JDBC Resource.
Either choose an existing JDBC Connection Pool, or create a new JDBC Connection Pool.
Fields with an * mark are required.

☒ Use Existing JDBC Connection Pool

eurekaPool

☐ Create New JDBC Connection Pool

JNDI Name:* jdbc/eureka

Object Type: user

Enabled: true

Description:

< Back Next > Finish Cancel Help

Para finalizar se debe ejecutar el botón **Finish**.

5.4 Creación de un servlet de prueba

Crear un servlet de nombre Demo01 y programarlo de la siguiente manera:

```
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

@WebServlet(name = "Demo01", urlPatterns = {"/Demo01"})
public class Demo01 extends HttpServlet {

    @Override
    protected void service(HttpServletRequest req,
        HttpServletResponse resp) throws ServletException, IOException {
        resp.setContentType("text/html;charset=UTF-8");
        PrintWriter out = resp.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>PRUEBA DEL POOL</title>");
            out.println("</head>");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        out.println("<body>");
        out.println("<h1>PRUEBA DEL POOL</h1>");
        // Uso del pool
        Context ctx = new InitialContext();
        DataSource ds = (DataSource) ctx.lookup("jdbc/eureka");
        //Obtener la conexión del Pool
        Connection cn = ds.getConnection();
        out.println("<p style='color:blue'>" + "Conexión ok." + "</p>");
        //Devolver la conexión al Pool
        cn.close();
    } catch (Exception e) {
        out.println("<p style='color:red'>Error: " + e.getMessage() + "</p>");
    } finally {
        out.println("</body>");
        out.println("</html>");
        out.close();
    }
}
}

```

5.5 Ejecución de la prueba

A continuación se muestra el resultado de la ejecución del servlet.



6 Otras pruebas a desarrollar

Programar los siguientes procesos:

1. Ingreso al sistema.
2. Consultar los movimientos de una cuenta.
3. Registrar un deposito