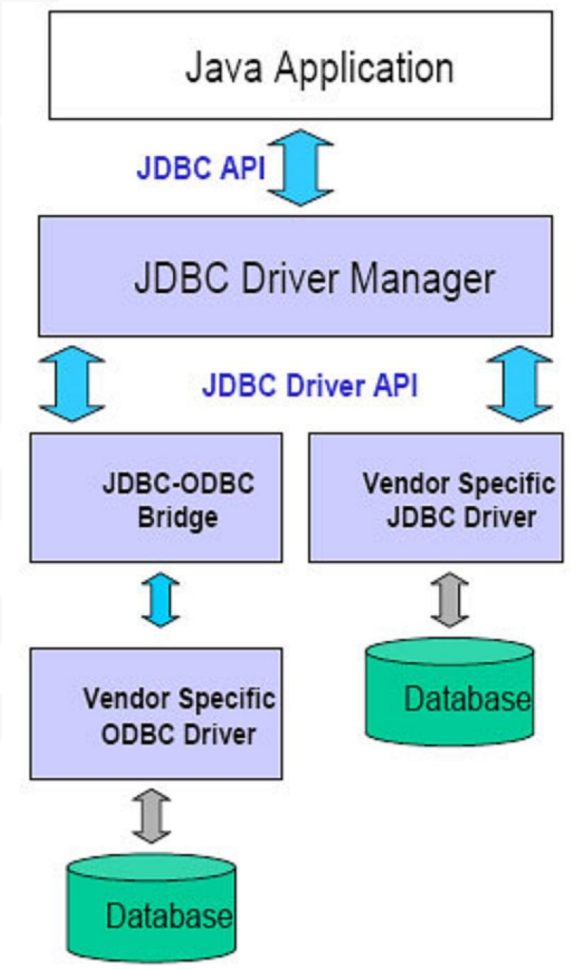
Databases Laboratory
Lesson 8

Connection PostgreSQL - Java

JDBC

- JDBC (Java DataBase Connectivity) is a Java API that allows:
  - Connect to databases and
  - Perform operations on them using SQL instructions from a Java application.
- JDBC offers an interface to connect to a database regardless of the type of database.
- What does JDBC do?
  - Establish a connection with a database
  - Send SQL statements
  - Process the results

JDBC

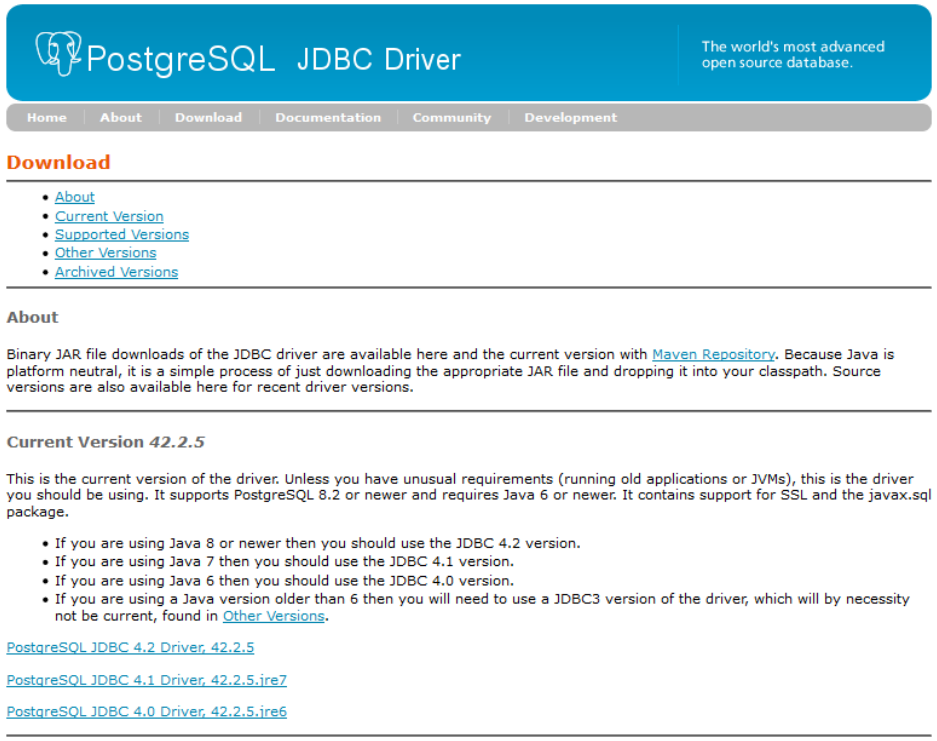

Java Application
JDBC API
JDBC Driver Manager
JDBC Driver API
JDBC-ODBC Bridge
Vendor Specific JDBC Driver
Vendor Specific ODBC Driver
Database
Database

Connection Java-PostgreSQL

Steps to connect a JAVA application with a PostgreSQL server

1- Download JDBC Driver

- Go to https://jdbc.postgresql.org/download.html, and download the JAR corresponding to our version of the installed JDK.



PostgreSQL JDBC Driver

The world's most advanced open source database.

Home | About | Download | Documentation | Community | Development

Download

- About
- Current Version
- Supported Versions
- Other Versions
- Archived Versions

About

Binary JAR file downloads of the JDBC driver are available here and the current version with Maven Repository. Because Java is platform neutral, it is a simple process of just downloading the appropriate JAR file and dropping it into your classpath. Source versions are also available here for recent driver versions.

Current Version *42.2.5*

This is the current version of the driver. Unless you have unusual requirements (running old applications or JVMs), this is the driver you should be using. It supports PostgreSQL 8.2 or newer and requires Java 6 or newer. It contains support for SSL and the javax.sql package.

- If you are using Java 8 or newer then you should use the JDBC 4.2 version.
- If you are using Java 7 then you should use the JDBC 4.1 version.
- If you are using Java 6 then you should use the JDBC 4.0 version.
- If you are using a Java version older than 6 then you will need to use a JDBC3 version of the driver, which will by necessity not be current, found in Other Versions.

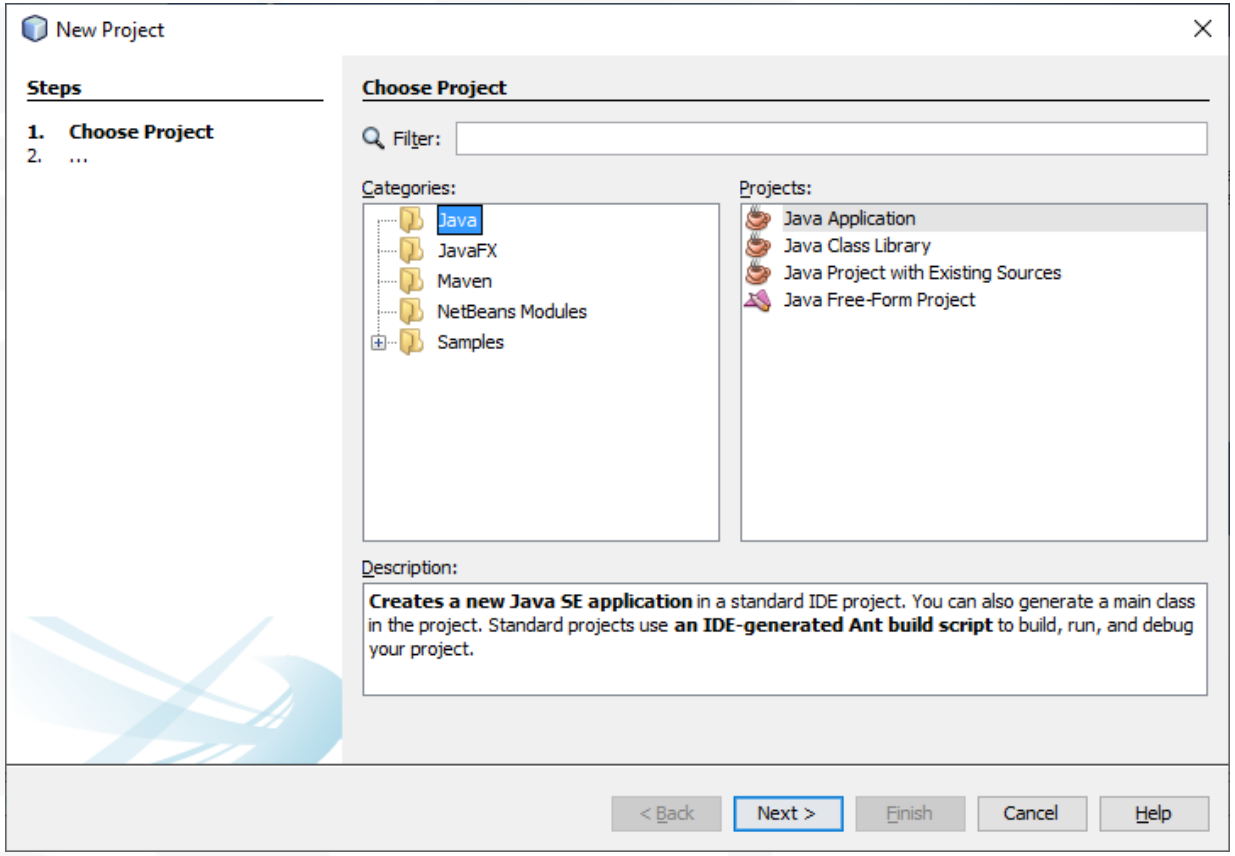
PostgreSQL JDBC 4.2 Driver, 42.2.5

PostgreSQL JDBC 4.1 Driver, 42.2.5.jre7

PostgreSQL JDBC 4.0 Driver, 42.2.5.jre6

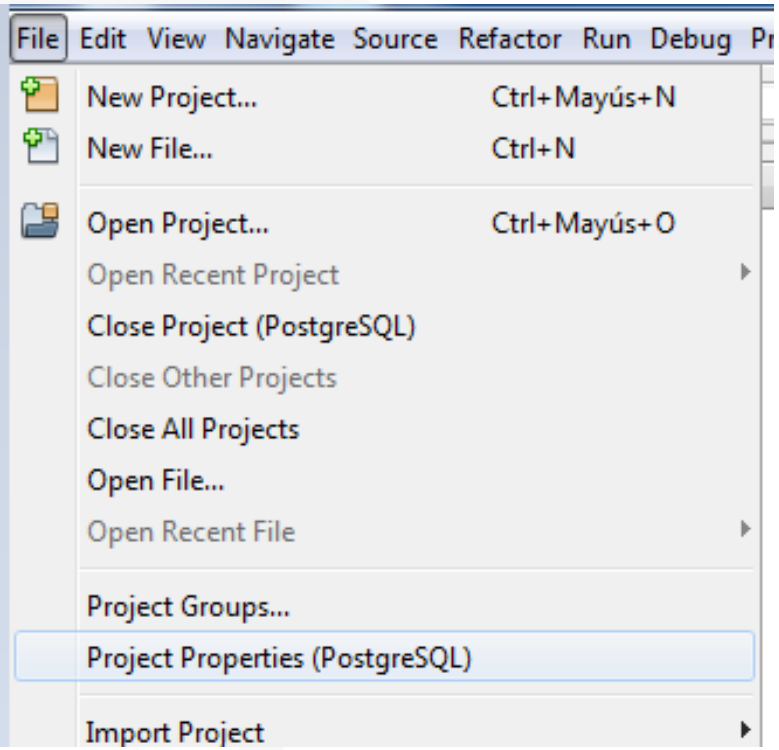
2- Create a Netbeans project

- Create a new Project, type "*Java Application*"



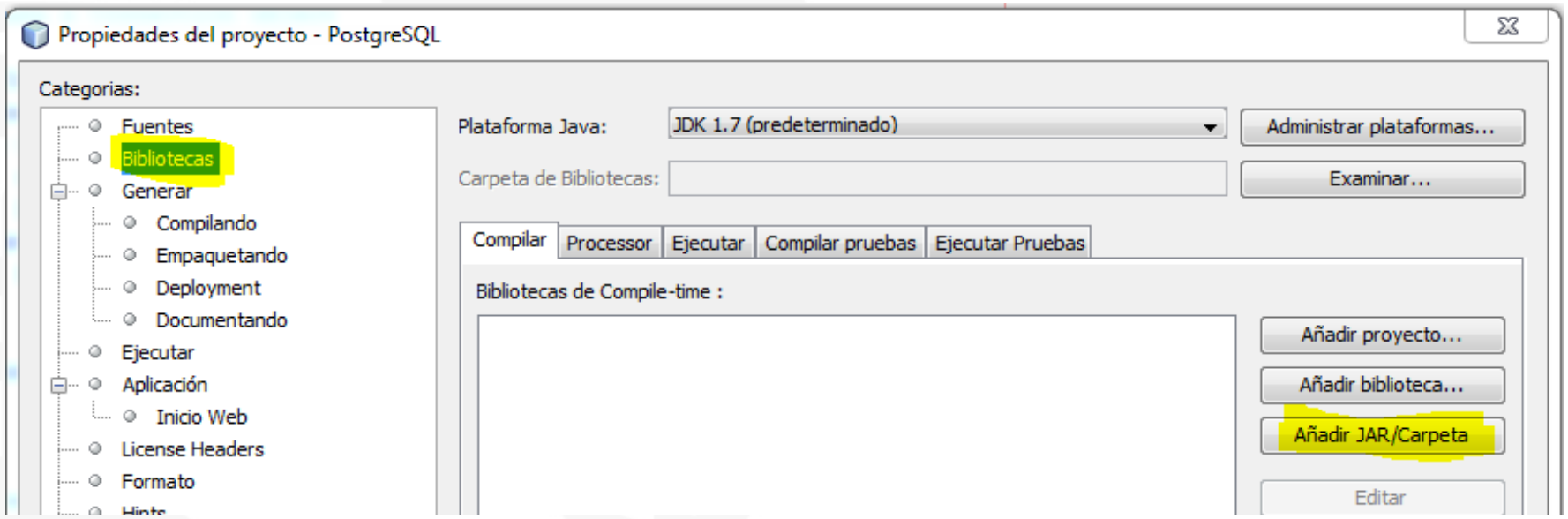
3- Associate the JAR file to the project (i)

- Access to the properties of the project: “File > Project Properties (ProjectName)”

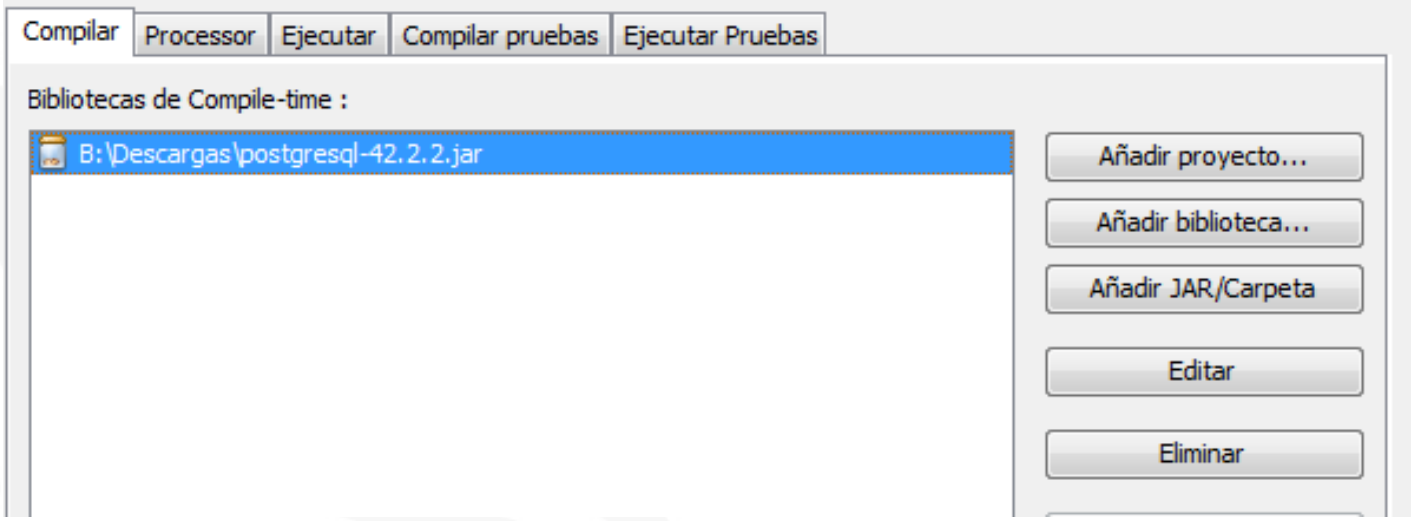


3- Associate the JAR file to the project (ii)

- Access the “Libraries” section and click on the “Add JAR / Folder” button, select the previously downloaded .jar file and accept



3- Associate the JAR file to the project (iii)



4- Add code (i)

- First, we must import the following classes :

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
```

4- Add code (ii)

- For convenience, we create some variables to store the server URL and access credentials:

```
private final String url = "jdbc:postgresql://<IP>:<puerto>/<DBname>";
private final String user = "<login>";
private final String password = "<contraseña>";
```

4- Add code (iii)

- Create a function to make the connection:

```
Connection conn = null;
public Connection connect() {
   try {
      Class.forName("org.postgresql.Driver").newInstance();
      conn = DriverManager.getConnection(url, user, password);
      System.out.println("Connected!!");
   } catch (SQLException e) {
      System.out.println(e.getMessage());
   }
   return conn;
}
```

4- Add code (iv)

- Create a function to close the connection :

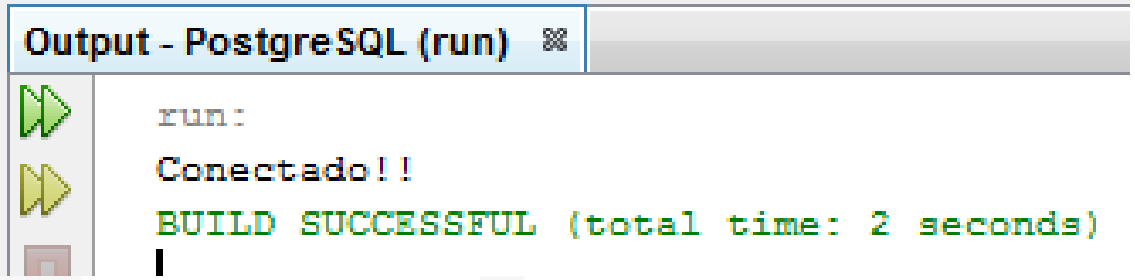
```
public void disconnect(){
    try {
        conn.close();
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    } catch (Exception ex){
        System.out.println(ex.getMessage());
    }
}
```

4- Add code (v)

- Program the main:

```
public static void main(String[] args) {
    PostgreSQL app = new PostgreSQL();
    app.connect();
}
```

- Run:



5- Common errors

- No suitable driver found for jdbc:postgresql://xx.xx.xx.xx:xx/yyyy
  - There is no connection to the server from the client
  - Solution: configure router or PC firewall

- FATAL: no hay una línea en pg_hba.conf para «192.168.1.3», usuario «postgres», base de datos «DBname»,
  - PostgreSQL does not allow incoming connections
  - Solution: configure the pg_hba.conf file (next slide)

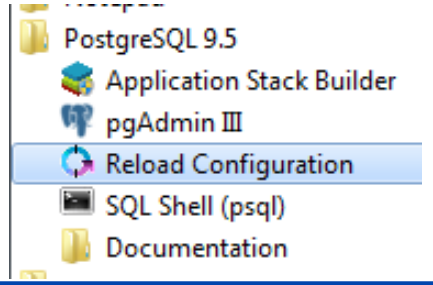
5- Common errors

- In the file *"pg_hba.conf"*, typically located in *"C:\Program Files\PostgreSQL\9.5\data\pg_hba.conf"* add at the end:

  #Accept incoming connections

  host all all 0.0.0.0/0 md5

- Save and restart PostgreSQL. Restart by means of commands or via the "Reload configuration" file (Start, All Programs) (note: run as administrator).)



SQL queries using Java

SQL queries using Java

- We use the Statement class to send the query to the database and we use an object of the ResultSet class to save or print the result.

- The result can have several rows, so it is advisable to use a while loop.

SQL queries using Java

- We create a function that performs a query and displays the result.

```
public void getAllClients(){
    try {
        Statement stmnt = conn.createStatement();
        ResultSet rs = stmnt.executeQuery("SELECT * FROM \"Client\";");

        System.out.println("List of clients");
        System.out.println("============================");
        while(rs.next()){
            System.out.println(""+rs.getString("name"));
        }
    } catch (SQLException ex) { System.out.println(ex.getMessage()); }
}
```

SQL queries using Java

- We program the *main()*

```
public static void main(String[] args) {
        PostgreSQL app = new PostgreSQL();
        app.connect();
        app.getAllClients();
        app.disconnect();
}
```

SQL queries using Java

```
Output - PostgreSQL (run)
run:
Conectado!!
Listado de clientes
============================
Ana
Berto
Camilo
Damian
Estefania
============================
Desconectado!!
BUILD SUCCESSFUL (total time: 0 seconds)
```