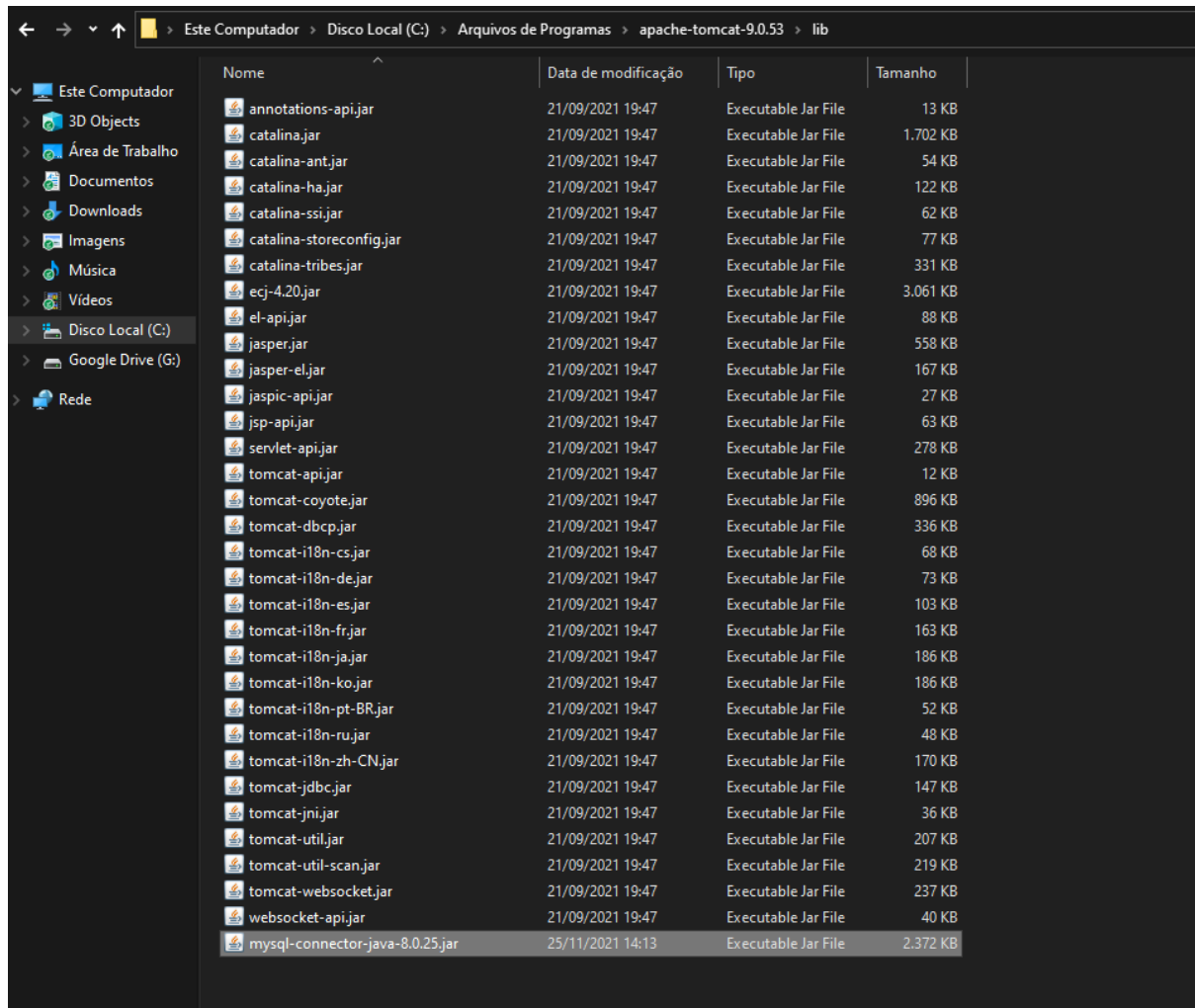


Como uso o JDK 11, assim como nas atividades anteriores, usei o tomcat como servidor e usando maven também, porque o Glassfish não é compatível com o JDK 11, e uso Windows.

Para configurar um DataSource no Tomcat primeiro é necessário colocar o **jar** do driver de conexão do banco na pasta **lib** do Tomcat.



Depois é necessário configurar o Context. esta configuração pode ser feita no contexto da aplicação **/META-INF/context.xml** sendo esta configuração disponível somente para a aplicação, ou ser feita no **CATALINA_HOME/conf/context.xml** que torna a configuração disponível para todas as aplicações do servidor.

Eu optei por configurar a do Tomcat e as minhas configurações foram as seguintes:

```
<Resource
    maxTotal="10"
    maxIdle="4"
    maxWaitMillis="15000"
    name="jdbc/banco_testes"
```

```
auth="Container"
type="javax.sql.DataSource"
username="root"
password="1998"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/banco_testes?serverTimezone=UTC" />
```

```
context.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 Licensed to the Apache Software Foundation (ASF) under one or more
4 contributor license agreements. See the NOTICE file distributed with
5 this work for additional information regarding copyright ownership.
6 The ASF licenses this file to You under the Apache License, Version 2.0
7 (the "License"); you may not use this file except in compliance with
8 the License. You may obtain a copy of the License at
9
10 http://www.apache.org/licenses/LICENSE-2.0
11
12 Unless required by applicable law or agreed to in writing, software
13 distributed under the License is distributed on an "AS IS" BASIS,
14 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15 See the License for the specific language governing permissions and
16 limitations under the License.
17 -->
18 <!-- The contents of this file will be loaded for each web application -->
19 <Context>
20
21 <!-- Default set of monitored resources. If one of these changes, the -->
22 <!-- web application will be reloaded. -->
23 <WatchedResource>WEB-INF/web.xml</WatchedResource>
24 <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
25 <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
26
27 <!-- Uncomment this to disable session persistence across Tomcat restarts -->
28 <!--
29 <Manager pathname="" />
30 -->
31 <Resource
32     maxTotal="10"
33     maxIdle="4"
34     maxWaitMillis="15000"
35     name="jdbc/banco_testes"
36     auth="Container"
37     type="javax.sql.DataSource"
38     username="root"
39     password="1998"
40     driverClassName="com.mysql.jdbc.Driver"
41     url="jdbc:mysql://localhost:3306/banco_testes?serverTimezone=UTC" />
42 </Context>
43
```

Os detalhes da configuração são os seguintes:

- **maxTotal:** O número máximo de conexões do banco de dados no pool ao mesmo tempo. Tenha certeza que sua configuração do `mysqld(max_connections)` é capaz de lidar com todas estas conexões. Para conexões ilimitadas limite utilize -1.
- **maxIdle:** O número máximo de conexões ociosas retidas no pool. Para conexões ilimitadas limite utilize -1.
- **maxWaitMillis:** Máximo de tempo de espera para uma conexão ao banco de dados se tornar disponível(em ms), no exemplo acima 15 segundos. Uma exceção é lançada se o tempo for esgotado. Para não utilizar limite utilize -1.
- **name:** O nome JNDI do recurso a ser criado.
- **type:** O nome canônico da classe esperada pela aplicação web quando esta realizar o lookup para este recurso.
- **auth:** Especifica se a aplicação irá lidar com o recurso ou o Container. Os valores possíveis são Application ou Container.
- **username e password:** O nome de usuário e senha para a conexão ao banco de dados.
- **driverClassName:** Nome da classe do drive jdbc de conexão a ser utilizado. Neste caso o driver mysql.
- **url:** A url de conexão para seu banco de dados.

Depois é necessário configurar o ***/WEB-INF/web.xml*** da aplicação

```
<resource-ref>
```

```
    <res-ref-name>jdbc/banco_testes</res-ref-name>
```

```
    <res-type>javax.sql.DataSource</res-type>
```

```
    <res-auth>Container</res-auth>
```

```
</resource-ref>
```

```
FabricaConexao.java x web.xml x PessoaDaoClasseBanco.java x
Source General Servlets Filters Pages References Security History
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3
3 <servlet>
4     <servlet-name>Controller</servlet-name>
5     <servlet-class>com.jhsantiago.datasourcepool.controller.Controller</servlet-class>
6 </servlet>
7 <servlet-mapping>
8     <servlet-name>Controller</servlet-name>
9     <url-pattern>/controller</url-pattern>
10 </servlet-mapping>
11 <session-config>
12     <session-timeout>
13         30
14     </session-timeout>
15 </session-config>
16 <resource-ref>
17     <res-ref-name>jdbc/banco_testes</res-ref-name>
18     <res-type>javax.sql.DataSource</res-type>
19     <res-auth>Container</res-auth>
20 </resource-ref>
21 </web-app>
```

Agora para o código da aplicação para recuperar um conexão do *pool*:

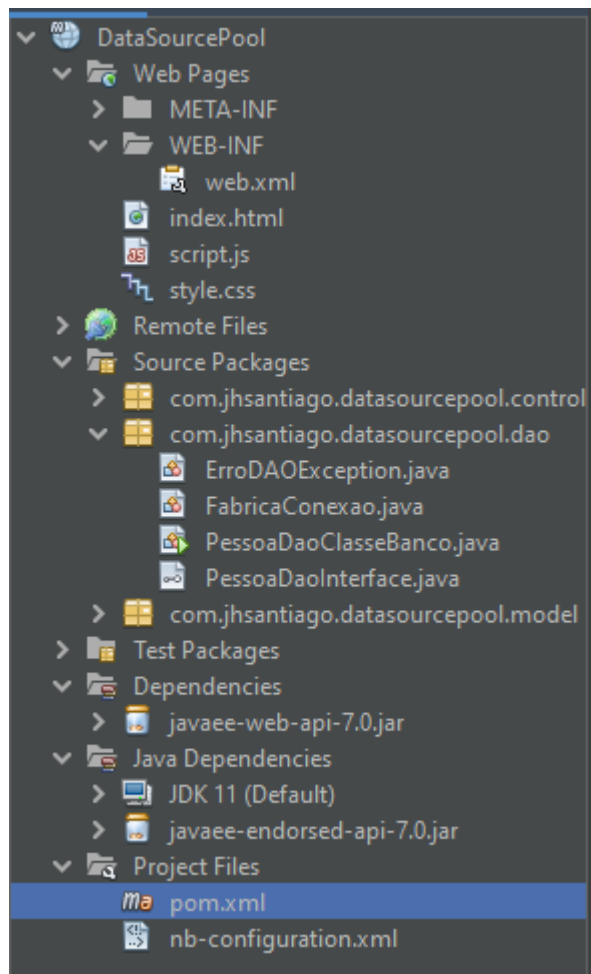
```
20 public class FabricaConexao {
21
22     public Connection pegaConexao() throws ErroDAOException {
23         try {
24             Connection con;
25             InitialContext ic = new InitialContext();
26             DataSource ds = (DataSource) ic.lookup("java:comp/env/jdbc/banco_testes");
27             con = ds.getConnection();
28             return con;
29         } catch (NamingException | SQLException ex) {
30             throw new ErroDAOException(ex);
31         }
32     }
33
34 }
```

Como pode ver no arquivo **pom.xml** não tem a dependência do mysql

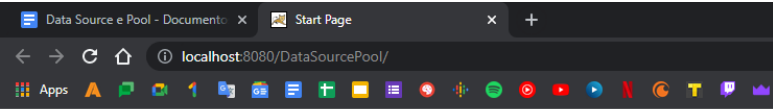


```
1  S<?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3      <modelVersion>4.0.0</modelVersion>
4
5      <groupId>com.jhsantiago</groupId>
6      <artifactId>DataSourcePool</artifactId>
7      <version>1.0-SNAPSHOT</version>
8      <packaging>war</packaging>
9
10     <name>DataSourcePool</name>
11
12     <properties>
13         <endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>
14         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15     </properties>
16
17     <dependencies>
18         <dependency>
19             <groupId>javax</groupId>
20             <artifactId>javaee-web-api</artifactId>
21             <version>7.0</version>
22             <scope>provided</scope>
23         </dependency>
24     </dependencies>
25
26     <build>
27         <plugins>
28             <plugin>
29                 <groupId>org.apache.maven.plugins</groupId>
30                 <artifactId>maven-compiler-plugin</artifactId>
31                 <version>3.1</version>
32                 <configuration>
33                     <source>1.7</source>
34                     <target>1.7</target>
35                     <compilerArguments>
36                         <endorseddirs>${endorsed.dir}</endorseddirs>
37                     </compilerArguments>
38                 </configuration>
39             </plugin>
40             <plugin>
41                 <groupId>org.apache.maven.plugins</groupId>
42                 <artifactId>maven-war-plugin</artifactId>
43                 <version>2.3</version>
44                 <configuration>
45                     <failOnMissingWebXml>>false</failOnMissingWebXml>
```

Nem na pasta do projeto



Porém conseguir acessar os dados da tabela:



Código	Nome	Idade
1	jhonatan	23
4	Juliana	25



Navigator: Query 1

SCHEMAS

Filter objects

- banco_testes
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sakila
- sys
- world

1 • select * from pessoa

2

Result Grid

	codigo	nome	idade
▶	1	jhonatan	23
	4	Juliana	25
*	NULL	NULL	NULL