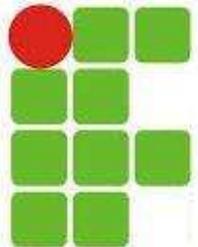

Prof. Diego Oliveira

BD



Conexão com BD

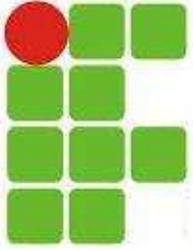




Conexões

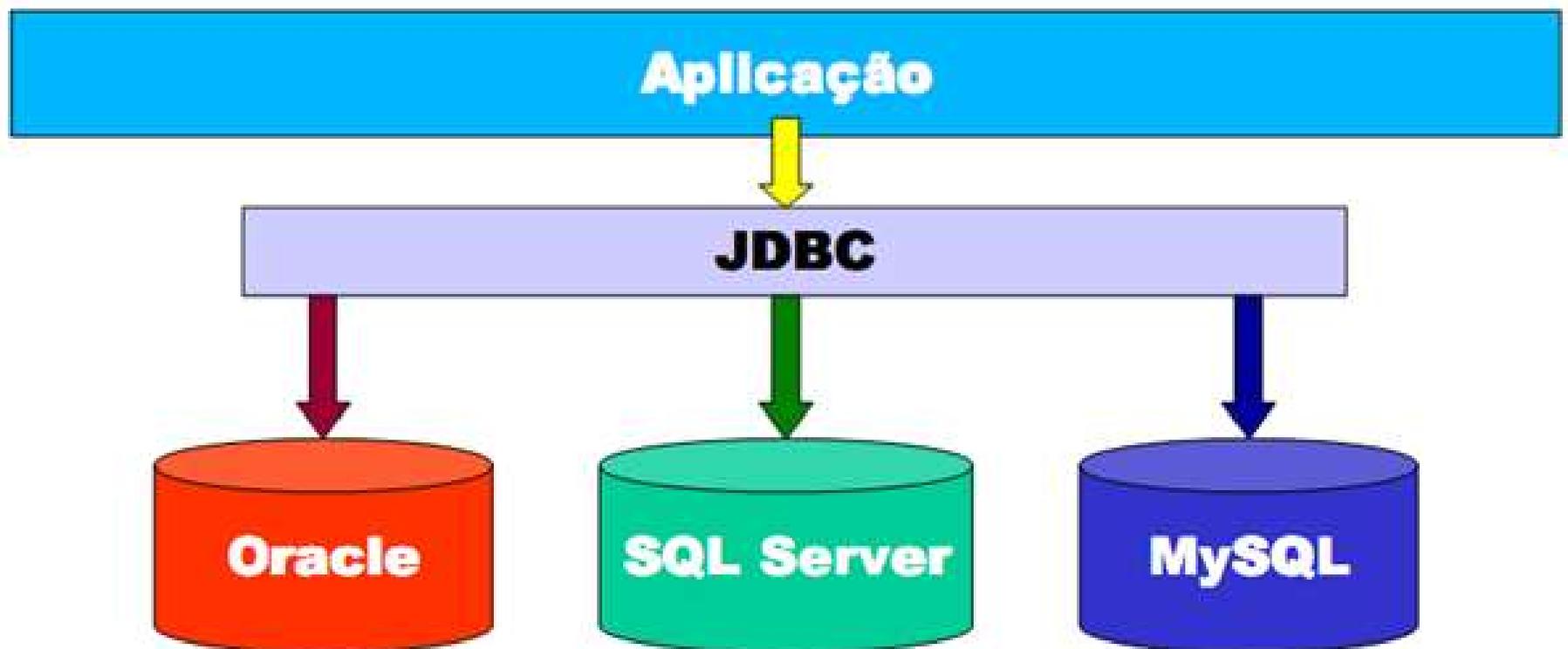
- Nas aulas anteriores aprendemos o que é um banco de dados, quais são suas representações, formas normais e a linguagem SQL: SELECT, INSERT, UPDATE e DELETE; além de junções, funções de agregação (avg, sum, min, max, count) e subconsultas
- Agora que aprendemos a teoria precisamos partir para a prática!





Conexões

- As conexões com banco de dados em JAVA utilizam o JDBC (Java Database Connectivity)

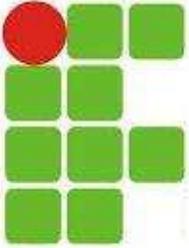




Conexões

- O JDBC é uma API, ou seja, um conjunto de bibliotecas que possibilita a conexão com diferentes bancos de dados
- O pacote principal é o `java.sql.*`
- Para cada banco de dados existe um DRIVER de conexão específico, porém a forma de conexão no código Java permanece a mesma
- Vamos ver dois exemplos:





Conexões

- Configuração da conexão para PostgreSQL:

```
9   public class Banco {
10      private Statement stmt;
11      private ResultSet rs;
12      public Connection conn;
13      private static Banco b = new Banco();
14      public Banco() {
15          String url = "jdbc:postgresql://localhost:5432/biblioteca";
16          String usr = "postgres";
17          String pas = "postgres";
18          try {
19              Class.forName("org.postgresql.Driver");
20              conn = DriverManager.getConnection(url, usr, pas);
21              stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
22                                          ResultSet.CONCUR_READ_ONLY);
23          } catch (Exception e) {
24              System.out.println("Erro" + e.getMessage());
25          }
26      }
}
```





Conexões

- Configuração da conexão para MySQL:

```
8 public class Banco {
9     private Statement stmt;
10    private ResultSet rs;
11    public Connection conn;
12    private static Banco b = new Banco();
13    public Banco() {
14        String url = "jdbc:mysql://localhost:3306/biblioteca";
15        String usr = "root";
16        String pas = "root";
17        try {
18            Class.forName("com.mysql.jdbc.Driver");
19            conn = DriverManager.getConnection(url, usr, pas);
20            stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
21                                     ResultSet.CONCUR_READ_ONLY);
22        } catch (Exception e) {
23            System.out.println("Erro" + e.getMessage());
24        }
25    }
}
```



Conexões

- Para os exemplos e trabalhos de sala de aula utilizaremos o MySQL, pois já está instalado nas máquinas do laboratório 3.
- O DRIVER recomendado para a versão 5.6 do MySQL é o
mysql-connector-java-5.1.32-bin.jar
- Também é possível clicar com o botão direito nas bibliotecas do projeto NetBeans e adicionar Driver do MySQL:





Conexões

Add Library [X]

Available Libraries:

- JSTL 1.2.2
- JUnit 3.8.2
- JUnit 4.10
- JWS Ant Tasks
- LWJGL
- METRO 2.0
- MySQL JDBC Driver**
- NetBeans DataBindingME
- NetBeans DataBindingME PIM
- NetBeans DataBindingME SVG
- Persistence (JPA 2.1)
- Persistence JPA2.1
- PostgreSQL JDBC Driver
- PrimeFaces 5.0
- Spring Framework 3.2.7

Contents of MySQL JDBC Driver - mysql-connector-java-5.1.23-bin.jar:

- META-INF
- META-INF.services
- com.mysql.jdbc
- com.mysql.jdbc.authentication
- com.mysql.jdbc.configs
- com.mysql.jdbc.exceptions
- com.mysql.jdbc.exceptions.jdbc4
- com.mysql.jdbc.integration.c3p0
- com.mysql.jdbc.integration.jboss
- com.mysql.jdbc.interceptors
- com.mysql.jdbc.jdbc2.optional
- com.mysql.jdbc.jmx
- com.mysql.jdbc.log
- com.mysql.jdbc.profiler
- com.mysql.jdbc.util
- org.gjt.mm.mysql

[Add Library] [Cancel]



Conexões

- A partir deste ponto, basta escrever os métodos com as consultas SQL específicas:

```
public void devolverLivro(int id, int usuario) throws SQLException {  
    stmt.executeUpdate("UPDATE livro set disponivel = 1, id_usuario=5 where id = " + id);  
}  
  
public ResultSet consultarMeusLivrosEmprestados(int usuario) throws SQLException {  
    return stmt.executeQuery("SELECT * from livro where id_usuario = " + usuario);  
}
```

- Também precisamos tratar a exceções, mas isso é assunto para a próxima aula 😊



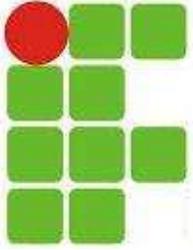


Atividade

- Crie um projeto no NetBeans que tenha uma classe Aluno com int id, String nome, int idade, double média.
- Crie a classe Banco.java para MySQL com os seguintes métodos:
 - inserirAluno(Aluno a)
 - consultarAluno(int id)
 - atualizarAluno(Aluno a)
 - removerAluno(int id)



- Crie um menu com as mesmas opções



Perguntas?

