



**UNIVERSIDADE PAULISTA**

Professor: Me. Pablo I. Gandulfo

Data: 01/08/2020

Versão: 1.1

# Aplicações de Linguagem de Programação Orientada a Objetos

---

MÓDULO 6



**UNIVERSIDADE PAULISTA**

Professor: Me. Pablo I. Gandulfo

Data: 01/08/2020

Versão: 1.1

## Manipulação de dados com linguagem SQL

---

- Passos Envolvidos
- Comandos SQL:
  - Statement
  - PreparedStatement
  - CallableStatement
- Exemplo Completo de um CRUD
  - Classe Banco de Dados
- Metadados e DDL
- Exercícios

## Exemplo Completo de um CRUD (Create, Read, Update e Delete)

---

- Classe Funcionário:
  - Propriedades:
    - Conexão (conexão de Banco de Dados)
    - Matrícula (inteiro)
    - Nome (texto)
    - Cargo (texto)
  - Métodos:
    - Construtor com Conexão: guarda a conexão passada
    - Construtor com Conexão e Matrícula: guarda a conexão e recupera os dados correspondentes (SELECT) do funcionário
    - Gravar: inclui (INSERT) ou atualiza (UPDATE) o funcionário
    - Excluir: remove o funcionário (DELETE)
    - Pesquisa (estático): pesquisa vários funcionários por parte do nome

## Exemplo Completo de um CRUD – Classe Banco de Dados

---

```
public class BancoDados {  
    private Connection connection;  
  
    public BancoDados() throws ClassNotFoundException, SQLException {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        Connection conn = DriverManager.getConnection(  
            "jdbc:mysql://localhost:3306/test?useSSL=false&serverTimezone=UTC", "root", "admin");  
    }  
  
    public Connection getConnection() {  
        return connection;  
    }  
  
    public void finalizar() throws SQLException {  
        connection.close();  
    }  
}
```

## Exemplo Completo de um CRUD – Parte Básica da Classe Funcionário

---

```
public class Funcionario {  
    private Connection connection;  
    private int matricula;  
    private String nome;  
    private String cargo;  
  
    public String toString() {  
        return " [" + matricula + "] " + nome + " - " + cargo;  
    }  
  
    // getters e setters
```

## Exemplo Completo de um CRUD – Instanciação de Funcionário

---

```
public Funcionario(Connection connection) {  
    this.connection = connection;  
}  
  
public Funcionario(Connection connection, int matricula) throws SQLException {  
    this(connection);  
    this.matricula = matricula;  
    PreparedStatement stmt = connection.prepareStatement(  
        "SELECT * FROM funcionario WHERE matricula = ?");  
    stmt.setInt(1, matricula);  
    ResultSet rs = stmt.executeQuery();  
    if (rs.next()) {  
        nome = rs.getString("nome");  
        cargo = rs.getString("cargo");  
    }  
    rs.close();  
    stmt.close();  
}
```

## Exemplo Completo de um CRUD – Método gravar() de Funcionário

---

```
public void gravar() throws SQLException {  
    PreparedStatement stmt = connection.prepareStatement(  
        "SELECT COUNT(*) AS cnt FROM funcionario WHERE matricula = ?");  
    stmt.setInt(1, matricula);  
    ResultSet rs = stmt.executeQuery();  
    rs.next();  
    if (rs.getInt("cnt") == 0) {  
        PreparedStatement stmtUpd = connection.prepareStatement(  
            "INSERT INTO funcionario (matricula, nome, cargo) VALUES (?, ?, ?)");  
        stmtUpd.setInt(1, matricula);  
        stmtUpd.setString(2, nome);  
        stmtUpd.setString(3, cargo);  
        stmtUpd.executeUpdate();  
        stmtUpd.close();  
    }  
}
```

## Exemplo Completo de um CRUD – Método gravar() de Funcionário (cont.)

---

```
else {  
    PreparedStatement stmtUpd = connection.prepareStatement(  
        "UPDATE funcionario SET nome = ?, cargo = ? WHERE matricula = ?");  
    stmtUpd.setString(1, nome);  
    stmtUpd.setString(2, cargo);  
    stmtUpd.setInt(3, matricula);  
    stmtUpd.executeUpdate();  
    stmtUpd.close();  
}  
rs.close();  
stmt.close();  
}
```



## Exemplo Completo de um CRUD – Método excluir() de Funcionário

---

```
public void excluir() throws SQLException {  
    PreparedStatement stmt = connection.prepareStatement(  
        "SELECT COUNT(*) AS cnt FROM funcionario WHERE matricula = ?");  
    stmt.setInt(1, matricula);  
    ResultSet rs = stmt.executeQuery();  
    rs.next();  
    if (rs.getInt("cnt") > 0) {  
        PreparedStatement stmtUpd = connection.prepareStatement(  
            "DELETE FROM funcionario WHERE matricula = ?");  
        stmtUpd.setInt(1, matricula);  
        stmtUpd.executeUpdate();  
        stmtUpd.close();  
    }  
    rs.close();  
    stmt.close();  
}
```

## Exemplo Completo de um CRUD – Método consultar() de Funcionário

---

```
public static ArrayList<Funcionario> consultar(Connection connection, String parteDoNome) throws
SQLException {
    PreparedStatement stmt = connection.prepareStatement(
        "SELECT matricula FROM funcionario WHERE nome LIKE ?");
    stmt.setString(1, "%" + parteDoNome + "%");
    ResultSet rs = stmt.executeQuery();
    ArrayList<Funcionario> resultado = new ArrayList<Funcionario>();
    while (rs.next()) {
        resultado.add(new Funcionario(connection, rs.getInt("matricula")));
    }
    rs.close();
    stmt.close();

    return resultado;
}
```

## Exemplo Completo de um CRUD – main() da Classe que Testa o CRUD

---

```
public static void main(String[] args) throws ClassNotFoundException, SQLException {  
    BancoDados bd = new BancoDados();  
    Funcionario f;  
  
    System.out.println("Listando os dados do funcionário 1 ...");  
    System.out.println(new Funcionario(bd.getConnection(), 1));  
  
    System.out.println("\nAlterando o cargo do funcionário 1 para 'Gerente Técnico' ...");  
    f = new Funcionario(bd.getConnection(), 1);  
    f.setCargo("Gerente Técnico");  
    f.gravar();  
  
    System.out.println("\nListando os dados do funcionário de matrícula 1 ...");  
    System.out.println(new Funcionario(bd.getConnection(), 1));  
}
```

## Exemplo Completo de um CRUD – main() da Classe que Testa o CRUD (cont.)

---

```
System.out.println("\nListando os dados de todos os funcionários ...");
ArrayList<Funcionario> funcionarios = Funcionario.consultar(bd.getConnection(), "a");
for (int i = 0; i < funcionarios.size(); i++) {
    System.out.println(funcionarios.get(i));
}

System.out.println("\nCadastrando o funcionário 3 ...");
f = new Funcionario(bd.getConnection(), 3);
f.setNome("Marlene Silva");
f.setCargo("Gerente de Gestão de Pessoas");
f.gravar();

System.out.println("\nListando os dados de todos os funcionários ...");
funcionarios = Funcionario.consultar(bd.getConnection(), "a");
for (int i = 0; i < funcionarios.size(); i++) {
    System.out.println(funcionarios.get(i));
}
```

## Exemplo Completo de um CRUD – main() da Classe que Testa o CRUD (cont.)

---

```
System.out.println("\nExcluindo o funcionário 3 ...");
f = new Funcionario(bd.getConnection(), 3);
f.excluir();

System.out.println("\nAlterando o cargo do funcionário 1 para 'Gerente de Operações' ...");
f = new Funcionario(bd.getConnection(), 1);
f.setCargo("Gerente de Operações");
f.gravar();

System.out.println("\nListando os dados de todos os funcionários ...");
funcionarios = Funcionario.consultar(bd.getConnection(), "a");
for (int i = 0; i < funcionarios.size(); i++) {
    System.out.println(funcionarios.get(i));
}

bd.finalizar();
}
```

## Exercício 1

---

- Crie a tabela CELULAR no Banco de Dados contendo os seguintes campos:
  - ID: inteiro
  - Modelo: texto
  - Preço: decimal
- Crie uma classe Celular contendo:
  - Propriedades: Id, Modelo e Preço
  - Métodos:
    - boolean incluir(Connection conexao, Celular c);
    - boolean excluir(Connection conexao, Integer id);
    - ArrayList consultarPorModelo(Connection conexao, String modelo);
- Desenvolva um código que cadastra alguns celulares no Banco de Dados
  - **Desafio:** a partir do arquivo Celulares.csv, cadastre os celulares no BD
- Em seguida, exclua alguns desses celulares
- Finalmente, consulte e imprima esses dados (no terminal)