

**UNIP**  
UNIVERSIDADE PAULISTA

# Aplicação Linguagem de Programação Orientada a Objeto

## ALPOO 03 - Conexão com banco de dados - JDBC

Prof. Msc Wanderley Gonçalves Freitas

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 1 de 30

**UNIP**  
UNIVERSIDADE PAULISTA

## Agenda


- SQL (Structured Query Language)
- Fundamentos do JDBC - Java Database Connectivity

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 2 de 30

**UNIP**  
UNIVERSIDADE PAULISTA

## Funcionalidade

- registrar o aluno com as seguintes funcionalidades: inclusão, exclusão, alteração e consulta



powered by Astah

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 3 de 30

**UNIP**  
UNIVERSIDADE PAULISTA

## SQL

O que é SQL (Structured Query Language) - É linguagem padrão para manipulação de dados em ambiente relacional.

- Linguagem definição de Dados (DDL - Data Definition Language)
  - Permite ao usuário a definição da estrutura e organização dos dados armazenados, a relações entre eles.

**HSQldb**

```

1 create table aluno
2 {
3     id          INTEGER GENERATED BY DEFAULT AS IDENTITY (START WITH 1),
4     matricula  int not null,
5     nome       varchar(254)not null,
6     primary key (id)
7 };

```

**MySQL**

```

1 create table aluno
2 {
3     id          int not null AUTO_INCREMENT,
4     matricula  int not null,
5     nome       varchar(254)not null,
6     primary key (id)
7 };

```

Prof: Msc Wanderley 4 de 30

**UNIP**  
UNIVERSIDADE PAULISTA

## SQL

- Linguagem manipulação de Dados (DML - Data Manipulation Language)
  - Permite ao usuário ou um programa, a execução de operações de inclusão, seleção, atualização e exclusão de dados armazenado na base de dados.

```

/* comando inserir */
INSERT INTO aluno (matricula, nome) VALUES (123, 'wanderley goncalves freitas');
INSERT INTO aluno (matricula, nome) VALUES (456, 'maria da silva');

/* comando selecao */
Select * from aluno

/* comando deletar */
Delete from aluno where id=1

/* comando atualizar */
update aluno set nome='jose da silva ' where id=2

```

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 5 de 30

**UNIP**  
UNIVERSIDADE PAULISTA

## JDBC

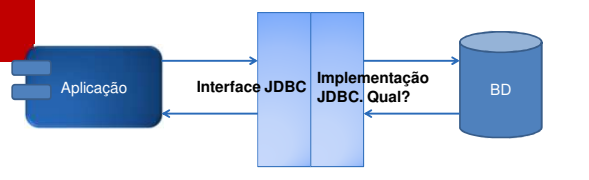
O que é JDBC?

- É uma API que reúne conjuntos de classes e interfaces escritas na linguagem Java na qual possibilita se conectar através de um **driver** específico do banco de dados desejado.
- Com esse driver pode-se executar instruções **SQL de qualquer tipo de banco de dados relacional**.

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 6 de 30

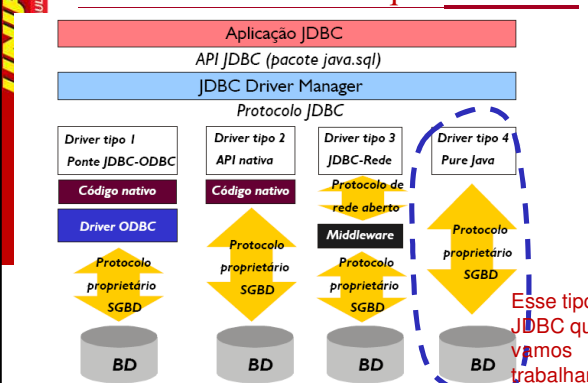
## A conexão em Java - JDBC

- Temos um único conjunto de interfaces muito bem definidas que devem ser implementadas.
- Esse conjunto de interfaces fica dentro do pacote java.sql e nos referimos a ele como JDBC.



Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 7 de 30

## Arquitetura JDBC

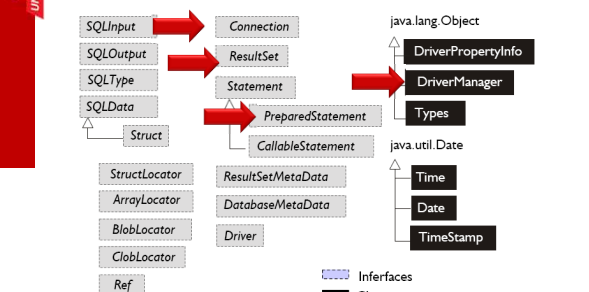


Esse tipo JDBC que vamos trabalhar

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 8 de 30

## Pacote java.sql

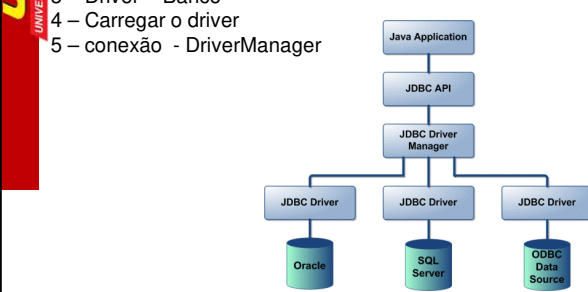
Esse pacote oferece a biblioteca Java o acesso e processamento de dados em uma fonte de dado. As classes e interfaces mais importantes são:



Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 9 de 30

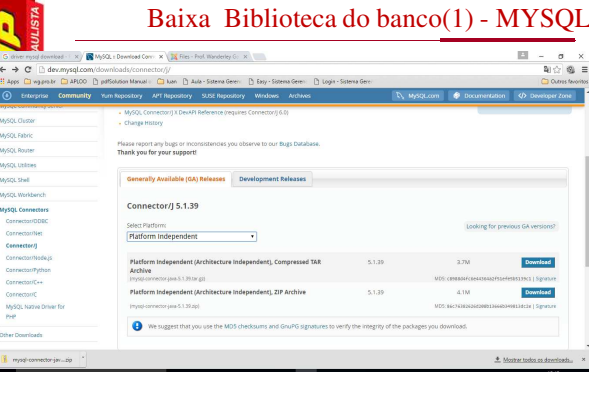
## Conexão – Processo de JDBC

- 1 – Baixa Biblioteca do banco
- 2 – URL - JDBC do banco
- 3 – Driver - Banco
- 4 – Carregar o driver
- 5 – conexão - DriverManager



Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 10 de 30

## Baixa Biblioteca do banco(1) - MYSQL



Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 11 de 30

## Baixa Biblioteca do banco(1) - HSQLDB



Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 12 de 30

## Baixa Biblioteca do banco(1) - SQLSERVER

Microsoft JDBC Drivers 6.0, 4.2, 4.1 e 4.0 para SQL Server

Selecione idioma: Português (Brasil) Baixar

Baixe o Microsoft JDBC Driver 6.0, 4.2, 4.1 ou 4.0 para SQL Server, um driver JDBC Tipo 4 que fornece conectividade a banco de dados por meio das APIs (interfaces de programação de aplicativo) padrão do JDBC disponíveis em Java Platform, Enterprise Editions.

Outlook  
Baixe o aplicativo de email grátis

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 13 de 30

## URL – JDBC(2)

- Uma aplicação JDBC pode carregar ao mesmo tempo diversos drivers.
- Para determinar qual driver será usado usa-se uma URL:
  - `jdbc:<subprotocolo>:<dsn>`
  - A aplicação usa o **subprotocolo** para identificar e selecionar o driver a ser instanciado.
  - O **dsn** é o nome que o subprotocolo utilizará para localizar um determinado servidor ou base de dados.
  - Sintaxe dependente do fabricante. Veja alguns exemplos:

```

jdbc:mysql://localhost:3306/loja
jdbc:hsqldb:hsqldb://localhost/faculdade;shutdown=true;
jdbc:oracle:thin:@localhost:1521:faculdade
jdbc:postgresql://localhost/faculdade
jdbc:sqlserver://localhost;databaseName=faculdade;
    
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 16 de 30

## Driver – Banco(3)

- A interface **Driver** é utilizada apenas pelas implementações de drivers JDBC
  - É preciso carregar a classe do driver na aplicação que irá utilizá-lo. Isto pode ser feito com `Class.forName()`:

```

Class.forName("com.mysql.jdbc.Driver"); // mysql
Class.forName("org.hsqldb.jdbcDriver"); //Hsqldb
Class.forName("oracle.jdbc.OracleDriver"); //Oracle
Class.forName("org.postgresql.Driver"); //Postgres
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");/
    
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 15 de 30

## A conexão em Java - DriverManager

A classe DriverManager é responsável por se comunicar com o driver.

```

String driver = "org.hsqldb.jdbcDriver";
String url = "jdbc:hsqldb:hsqldb://localhost/faculdade;shutdown=true;";
String usuario = "sa";
String senha = "";

Connection conexao = null;
try {
    Class.forName(driver);
    conexao = DriverManager.getConnection(url, usuario, senha);
}
    
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 16 de 30

## classe Conexão

```

public final class FabricaConexao {
    private static final String HSQLDB_USUARIO = "sa";
    private static final String HSQLDB_SENHA = "";
    private static final String HSQLDB_DRIVER = "org.hsqldb.jdbcDriver";
    private static final String HSQLDB_URL = "jdbc:hsqldb:hsqldb://localhost/bibliotecaaula";

    public static Connection getConnection() {
        Connection conexao = null;
        try {
            Class.forName(HSQLDB_DRIVER);
            conexao = DriverManager.getConnection(HSQLDB_URL, HSQLDB_USUARIO, HSQLDB_SENHA);
        } catch (SQLException | ClassNotFoundException e) {
            throw new RuntimeException(e);
        }
        return conexao;
    }

    public static void fechaConexao(Connection pConexao) {
        try {
            if ((pConexao != null) && (!pConexao.isClosed())) {
                pConexao.close();
            }
            pConexao = null;
        }
    }
}
    
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 17 de 30

## Manipulando bancos de dados com o JDBC

Delete from aluno where id=1

INSERT INTO aluno (matricula, nome) VALUES (123, 'wanderley goncalves freitas')

update aluno set nome='jose da silva ' where id=1

Select \* from aluno

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 18 de 30

## Manipulando bancos de dados com o JDBC

- Interfaces que contém métodos implementados em todos os drivers JDBC.
- Connection
  - Representa uma conexão ao banco de dados, que é retornada pelo DriverManager na forma de um objeto.
- Statement
  - Oferece meios de passar instruções SQL para o sistema de bancos de dados.
- ResultSet
  - É um cursor para os dados recebidos.

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 19 de 30

## Manipulando bancos de dados com o JDBC

### Statement

- Obtendo-se um objeto Connection, chama-se sobre ele o método createStatement() para obter um objeto do tipo Statement:
 

```
Statement stmt = con.createStatement();
```

 que poderá usar métodos como execute(), executeQuery(), executeBatch() e executeUpdate() para enviar instruções SQL ao BD.
- Subinterfaces:
  - PreparedStatement e CallableStatement
 

```
PreparedStatement pstmt = con.prepareStatement(...);
CallableStatement cstmt = con.prepareCall(...);
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 20 de 30

## Manipulando bancos de dados com o JDBC

### PreparedStatement

- Statement *pré-compilado* que é mais eficiente quando várias queries similares são enviadas com parâmetros diferentes
- String com instrução SQL é preparado previamente, deixando-se "?" no lugar dos parâmetros
- Parâmetros são inseridos em ordem, com setXXX() onde XXX é um tipo igual aos retornados pelos métodos de ResultSet

```
String sql = "INSERT INTO Livros VALUES(?, ?, ?)";
PreparedStatement pstmt = con.prepareStatement(sql);
pstmt.setInt(1, 18943);
pstmt.setString(2, "Lima Barreto");
pstmt.setString(3, "O Homem que Sabia Javanês");
pstmt.executeUpdate();
...
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 21 de 30

## Manipulando bancos de dados com o JDBC

### ResultSet

- O método executeQuery(), da interface Statement, retorna um objeto ResultSet.
  - Cursor para as linhas de uma tabela.
  - Pode-se navegar pelas linhas da tabela recuperar as informações armazenadas nas colunas
- Os métodos de navegação são
  - next(), previous(), absolute(), first() e last()
- Métodos para obtenção de dados:
  - getInt()
  - getString()
  - getDate()
  - getXXX(), ...

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 22 de 30

## Manipulando bancos de dados com o JDBC

### Tipos JDBC e métodos getXXX()

Método de ResultSet	Tipo de dados SQL92
getInt()	INTEGER
getLong()	BIG INT
getFloat()	REAL
getDouble()	FLOAT
getBigDecimal()	DECIMAL
getBoolean()	BIT
getString()	CHAR, VARCHAR
getDate()	DATE
getTime()	TIME
getTimestamp()	TIME STAMP
getObject()	Qualquer tipo (Blob)

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 23 de 30

## Manipulando bancos de dados com o JDBC

### ResultSet

- Exemplo de uso de ResultSet
 

```
ResultSet rs = stmt.executeQuery("SELECT Numero, Texto, "
+ " Data FROM Anuncios");

while (rs.next()) {
    int x = rs.getInt("Numero");
    String s = rs.getString("Texto");
    java.sql.Date d = rs.getDate("Data");
    // faça algo com os valores obtidos...
}
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 24 de 30

**UNIP** UNIVERSIDADE PAULISTA

## Manipulando bancos de dados com o JDBC

### Transações

- Permite a execução atômica de comandos enviados ao banco. Implementada através dos métodos de *Connection*
  - `commit()`
  - `rollback()`
  - `setAutoCommit(boolean autoCommit)`: default é true.
- Por default, as informações são processadas a medida em que são recebidas. Para mudar:
 

```
con.setAutoCommit(false);
```
- Agora várias instruções podem ser acumuladas. Para processar:
 

```
con.commit();
```
- Se houver algum erro e todo o processo necessitar ser desfeito, pode-se emitir um ROLLBACK usando:
 

```
con.rollback();
```

Prof: Msc Wanderley Gonçalves Freitas 25 de 30

**UNIP** UNIVERSIDADE PAULISTA

## Manipulando bancos de dados com o JDBC

### Fechar conexão e exceções

- Após o uso, os objetos *Connection*, *Statement* e *ResultSet* devem ser **fechados**. Isto pode ser feito com o método `close()`:
  - `con.close();`
  - `stmt.close();`
  - `rs.close();`
- A exceção ***SQLException*** é a principal exceção a ser observada em aplicações JDBC

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 26 de 30

**UNIP** UNIVERSIDADE PAULISTA

## Referências bibliográficas

FAEMAN, Julio. Design Patterns Aplicados. *Java Magazine*, Rio de Janeiro, v. 3, n.20, p. 52 a 58, 2005.

FOWLER, Martin. **UML essencial : um breve guia para a linguagem** – padrão de modelagem de objetos. Tradução Vera Pezerico e Christian Thomas Prices. 3. ed. Porto Alegre : Bookman, 2005.

GRADY, Booch; RUMBAUGH, James; JACOBSON, Ivar. **UML, guia do usuário** . tradução de Fábio Freitas da Silva. Rio de Janeiro: Editora Campus. 2000.

JOHN, Melsker, Steven. **Padrões de projeto em java**. Porto Alegre: Editora Bookman. 2004.

KAZUO, Jorge. Catálogo de Soluções : Padrões de projeto na arquitetura JE22. *Java Magazine*, Rio de Janeiro, v. 3, n.20, p. 12 a 13, 2005.

LOZANO, Fernando. Padrões de Projeto e Classes de Negócio. *Java Magazine*, Rio de Janeiro, v. 3, n.20, p. 10 a 11, 2005. Cateína.

MENDES, Antonio. **Arquitetura de software : desenvolvimento orientado para arquitetura**. Rio de Janeiro: Editora Campus. 2002.

QUATRANI, Terry. **Modelagem visual rational rose 2000 e UML**. Rio de Janeiro: Editora Ciência Moderna Ltda. 2001.

Scott W. Ambler. **The Elements of Java Style**. Disponível em: <http://www.amblysoft.com/books/elementsJavaStyle.html> />. Acesso em : 01/12/2008

Writing Robust Java Code  
<http://www.amblysoft.com/downloads/javaCodingStandards.pdf/>> Acesso em : 01/12/2008

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 27 de 30

**UNIP** UNIVERSIDADE PAULISTA

## Referências Bibliográficas

BRUCE, Eckel. **Thinking in Java**. 4 ed. Massachusetts: Editora Prentice hall: 2006.

FOWLER, Martin. **UML essencial : um breve guia para a linguagem** – padrão de modelagem de objetos. Tradução Vera Pezerico e Christian Thomas Prices. 3. ed. Porto Alegre : Bookman, 2005.

FURLAN, Jose Davi. **Modelagem de Objetos Através da UML** São Paulo: Editora Makron Books, 2000, ISBN 8534609241,

GRADY, Booch; RUMBAUGH, James; JACOBSON, Ivar. **UML, guia do usuário** . tradução de Fábio Freitas da Silva. Rio de Janeiro: Editora Campus. 2000.

SIERRA, kathy; BATES, Bert. **Certificação Sun para programadores e desenvolvedor Java 2**. Rio de Janeiro: Editora Altas Books. 2003.

SIERRA, kathy; BATES, Bert. **Head first java 5. 2 ed**. Sebastopol: Editora O' Reilly Media. Books. 2005.

TOM, Pender. **UML, a bíblia**. Rio de Janeiro: Elsevier, 2004.

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 28 de 30

**UNIP** UNIVERSIDADE PAULISTA

## Perguntas



Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 29 de 30

**UNIP** UNIVERSIDADE PAULISTA

## Obrigado

Wanderley

[Wanderley.unip@gmail.com](mailto:Wanderley.unip@gmail.com)

[www.wg.pro.br](http://www.wg.pro.br)

Prof: Msc Wanderley Gonçalves Freitas ALPOO 03 – Conexão com Banco - JDBC 30 de 30