



**TECNOLOGIA E INOVAÇÃO
EM PROL DA INDÚSTRIA**



Curso de Desenvolvimento de Sistemas

Conexão com o Banco de Dados

Prof: Diego Corrêa

Banco de dados

- Os bancos de dados e os sistemas de bancos de dados se tornaram componentes essenciais no cotidiano da sociedade moderna
- No decorrer do dia, a maioria de nós se depara com atividades que envolvem alguma interação com os bancos de dados

Banco de dados

Por exemplo:

- Se formos ao banco para efetuarmos um depósito ou retirar dinheiro
- Se fizermos reservas em um hotel ou para a compra de passagens aéreas
- Se acessarmos o catálogo de uma biblioteca informatizada para consultar uma bibliografia
- Se comprarmos produtos — como livros, brinquedos ou computadores

Banco de dados



JDBC

- O Java possui uma biblioteca que ajuda o programador a ligar o software a um banco de dados
- Permitindo uma melhor gerência e recuperação da informação previamente salva

Conceito

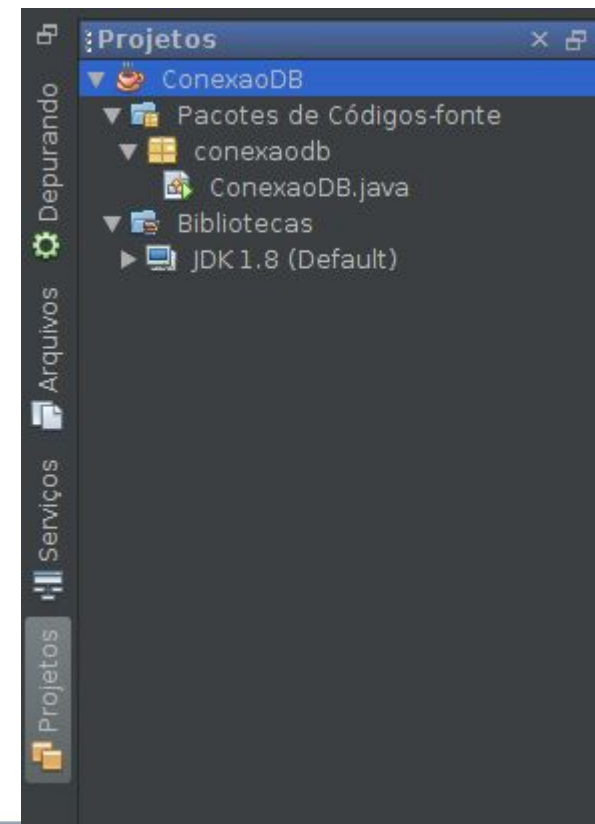
- Assim, por padrão o Java possui a **Java DataBase Connectivity (JDBC)**
- Biblioteca que já vem com o Java
- Para importar é necessário adicionar a linha:

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


Código

Para usar a biblioteca **JDBC** é necessário primeiro adicionar a biblioteca no projeto

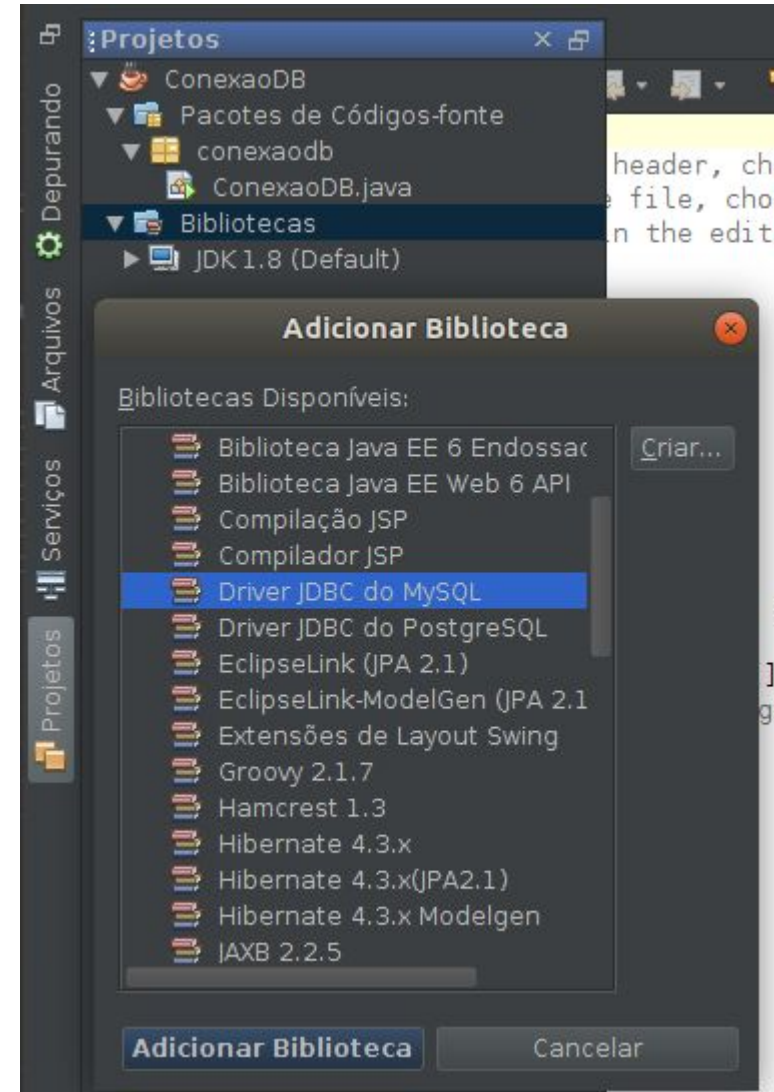
No NetBeans vá em projetos
Aperte com o botão mouse direito
Encima de Bibliotecas
E depois em Adicionar Bibliotecas



Código

Procure por:
Driver JDBC do MySQL

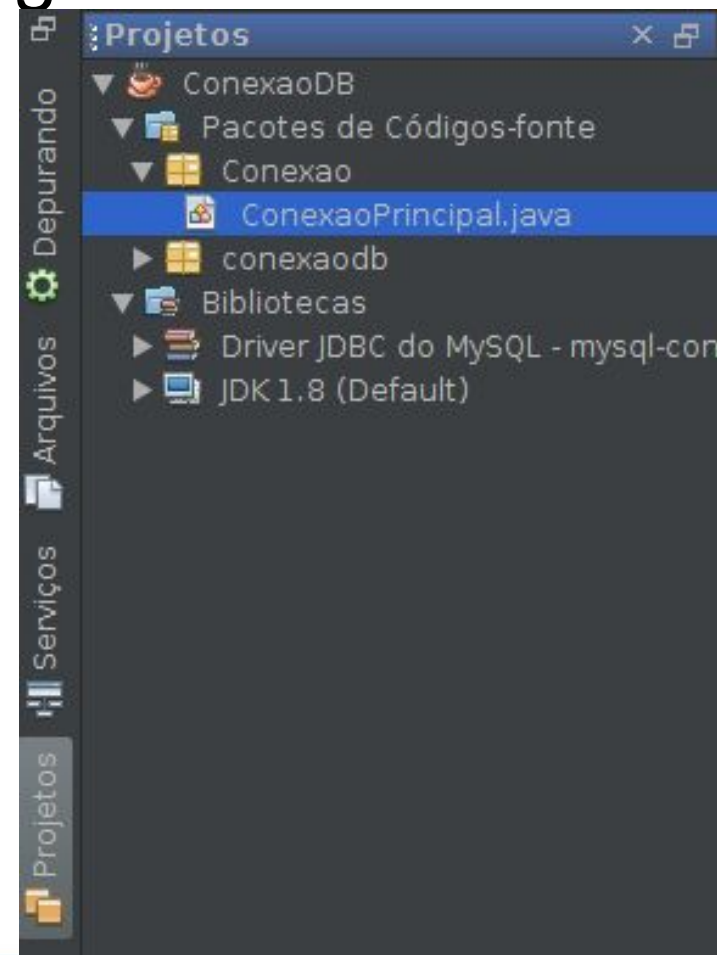
Depois aperte:
“Adicionar Biblioteca”



Código

Crie um Pacote dentro do programa chamado
“Conexao”

Depois crie uma classe:
“ConexaoPrincipal”



Código

- A classe “Conexão Principal” irá fornecer a base para a ligação com o banco de dados
- Nela é necessário adicionar atributos que são básicos em todas as conexões com qualquer banco de dados
- Assim com os imports:
 - `import java.sql.Connection;`
 - `import java.sql.DriverManager;`
 - `import java.sql.SQLException;`

Abertura e Fechamento de Conexão

Código

Importe todas as seguintes classes

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
import java.util.logging.Level;  
import java.util.logging.Logger;
```


Código

O Construtor irá inicializar todas as variáveis de conexão

Para isso um banco de dados deve estar previamente criado com os mesmos dados

```
public class ConexaoPrincipal {  
    private static final String DRIVER = "com.mysql.jdbc.Driver";  
    private static final String DB_NAME = "aula_conexao";  
    private static final String DB_PORT = "3306";  
    private static final String DB_PREFIX = "jdbc:mysql://";  
    private static final String HOST_NAME = "localhost";  
    private static final String DB_USER = "diego_";  
    private static final String DB_PW = "123456";  
    private static final String URL = DB_PREFIX + HOST_NAME + ":" + DB_PORT + "/" + DB_NAME + "/";  
}
```

Código

Dois métodos básicos devem ser implementados na classe:

- **getConnection()**
- **closeConnection()**

Código

```
public static Connection getConnection() {  
    try {  
        System.out.println("Iniciando Conexão!");  
        Class.forName(DRIVER);  
        return DriverManager.getConnection(URL, DB_USER, DB_PW);  
    } catch (ClassNotFoundException | SQLException e) {  
        throw new RuntimeException("Erro de conexão: " + e);  
    }  
}
```

Código

```
public static void closeConnection(Connection CON) {
    try {
        if (CON != null) {
            CON.close();
            System.out.println("Conexão Fechan!");
        }
    } catch (SQLException e) {
        Logger.getLogger(ConexaoPrincipal.class.getName()).log(Level.SEVERE, null, e);
    }
}

public static void closeConnection(Connection CON, PreparedStatement stmt) {
    try {
        if (CON != null && stmt != null) {
            closeConnection(CON);
            stmt.close();
        }
    } catch (SQLException e) {
        Logger.getLogger(ConexaoPrincipal.class.getName()).log(Level.SEVERE, null, e);
    }
}
```

Código

Vá no NetBeans no menu “Janelas” e procure por “Serviços”

Em Banco de Dados aperte com o botão direito do mouse e selecione “Nova Conexão”



Código

Assistente de Nova Conexão

Localizar Driver

Driver: MySQL (Connector/J driver)

Arquivo(s) de driver: /home/lowe/netbeans-8.2/ide/modules/ext/mysql-conne

Adicionar...

Remover

< Voltar

Próximo >

Assistente de Nova Conexão

Personalizar Conexão

Nome do Driver: MySQL (Driver Connector/J) em MySQL (Connector/J driver)

Host: localhost

Porta: 3306

Banco de dados: aula_conexao

Nome do Usuário: diego_

Senha:

☐ Lembrar senha

Propriedades da Conexão

Testar Conexão

JDBC URL:

jdbc://localhost:3306/aula_conexao?zeroDateTimeBehavior=convertToNull

< Voltar

Próximo >

Finalizar

Cancelar

Ajuda

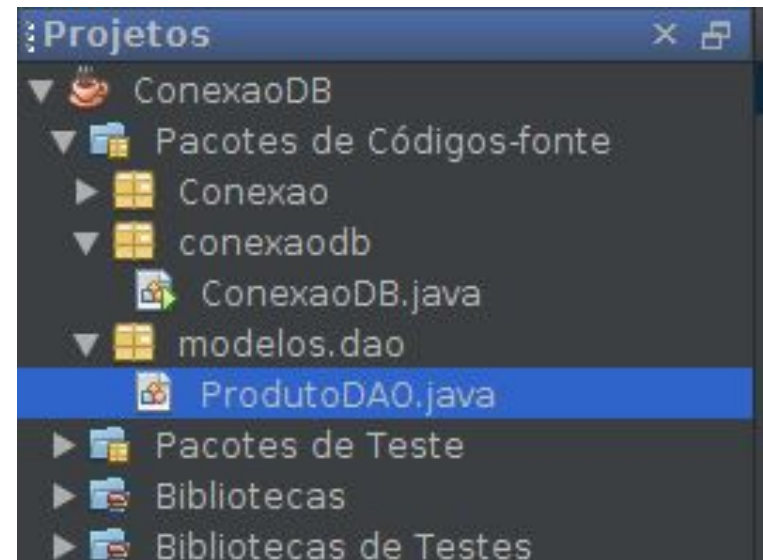
Inserindo Dados no Banco

Código

Crie um pacote chamado “modelos.dao” e dentro do pacote uma classe chamada “ProdutoDAO”

DAO significa:

Data Access Object



Código

Criar um pacote chamado “modelos.bean” e dentro adicionar uma classe chamada “Produto.java”



Código

Na classe “Produto” adicione os atributos e métodos que serão usados pela instância na memória

```
package modelos.bean;
```

```
public class Produto {  
    private int id;  
    private String descricao;  
    private double preco;  
    private int qtd;
```

Código

```

public Produto(String descricao, double preco, int qtd) {
    this.descricao = descricao;
    this.preco = preco;
    this.qtd = qtd;
}

public String getDescricao() {
    return descricao;
}

public double getPreco() {
    return preco;
}

public int getQtd() {
    return qtd;
}
  
```

Código

```
public void setDescricao(String descricao) {  
    this.descricao = descricao;  
}  
  
public void setPreco(double preco) {  
    this.preco = preco;  
}  
  
public void setQtd(int qtd) {  
    this.qtd = qtd;  
}  
  
public void imprimirDados() {  
    System.out.println("Descricao: " + this.getDescricao());  
    System.out.println("preco: " + this.getPreco());  
    System.out.println("Quantidade: " + this.getQtd());  
}
```


Código

Na classe “ProdutoDAO” importe as bibliotecas que serão usadas

```
package modelos.dao;
```

```
import Conexao.ConexaoPrincipal;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.SQLException;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
import modelos.bean.Produto;
```

Código

```
public class ProdutoDAO {  
    public void criar(Produto produ) {  
        Connection con = ConexaoPrincipal.getConnection();  
        PreparedStatement stmt = null;  
  
        try {  
            stmt = con.prepareStatement("INSERT INTO produto (descricao, qtd, preco) VALUES (?, ?, ?)");  
  
            stmt.setString(1, produ.getDescricao());  
            stmt.setInt(2, produ.getQtd());  
            stmt.setDouble(3, produ.getPreco());  
  
            stmt.executeUpdate();  
  
            System.out.println("Dado salvo com sucesso!");  
        } catch (SQLException ex) {  
            System.err.println("Não foi possível executar!");  
            Logger.getLogger(ProdutoDAO.class.getName()).log(Level.SEVERE, null, ex);  
        } finally {  
            ConexaoPrincipal.closeConnection(con, stmt);  
        }  
    }  
}
```

Código

Na classe principal do programa teste:

```
package conexaodb;

import modelos.bean.Produto;
import modelos.dao.ProdutoDAO;

public class ConexaoDB {

    public static void main(String[] args) {
        Produto p = new Produto("descrição da descrita", 200.5, 10);
        ProdutoDAO pDAO = new ProdutoDAO();

        pDAO.criar(p);
    }
}
```

Lendo Dados do Banco

Código

Na Classe “ProdutoDAO” adicione as novas chamadas

```
package modelos.dao;  
  
import Conexao.ConexaoPrincipal;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import modelos.bean.Produto;
```


Código

- Para ler os dados do banco é necessário criar um método que acesse esses dados
- E trate-os de forma que esses fiquem de fácil manipulação do programa

Código

```
public ArrayList<Produto> ler() {
    Connection con = ConexaoPrincipal.getConnection();
    PreparedStatement stmt = null;
    ResultSet resultado = null;

    ArrayList<Produto> listaProdutos = new ArrayList<>();

    try {
        stmt = con.prepareStatement("SELECT * FROM produto");
        resultado = stmt.executeQuery();

        while(resultado.next()) {
            listaProdutos.add(new Produto(
                resultado.getString("descricao"),
                resultado.getDouble("preco"),
                resultado.getInt("qtd")
            ));
        }
    } catch (SQLException ex) {
        Logger.getLogger(ProdutoDAO.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        ConexaoPrincipal.closeConnection(con, stmt, resultado);
    }
    return listaProdutos;
}
```

Código

```
package conexaodb;

import java.util.ArrayList;
import modelos.bean.Produto;
import modelos.dao.ProdutoDAO;

public class ConexaoDB {

    public static void main(String[] args) {
        Produto p = new Produto("descrição da descrita", 200.5, 10);
        ProdutoDAO pDAO = new ProdutoDAO();

        pDAO.criar(p);
        ArrayList<Produto> listProduto = pDAO.ler();
        for(Produto prod: listProduto) {
            prod.imprimirDados();
        }
    }
}
```

Código