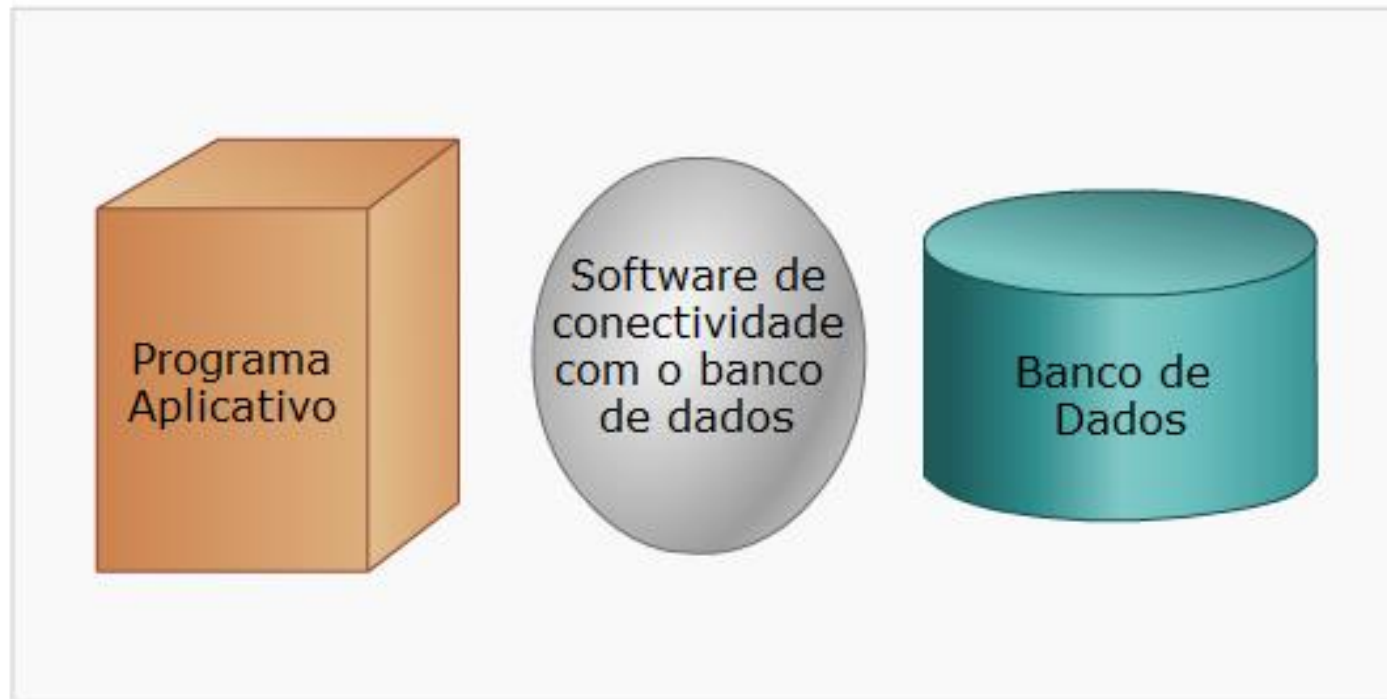
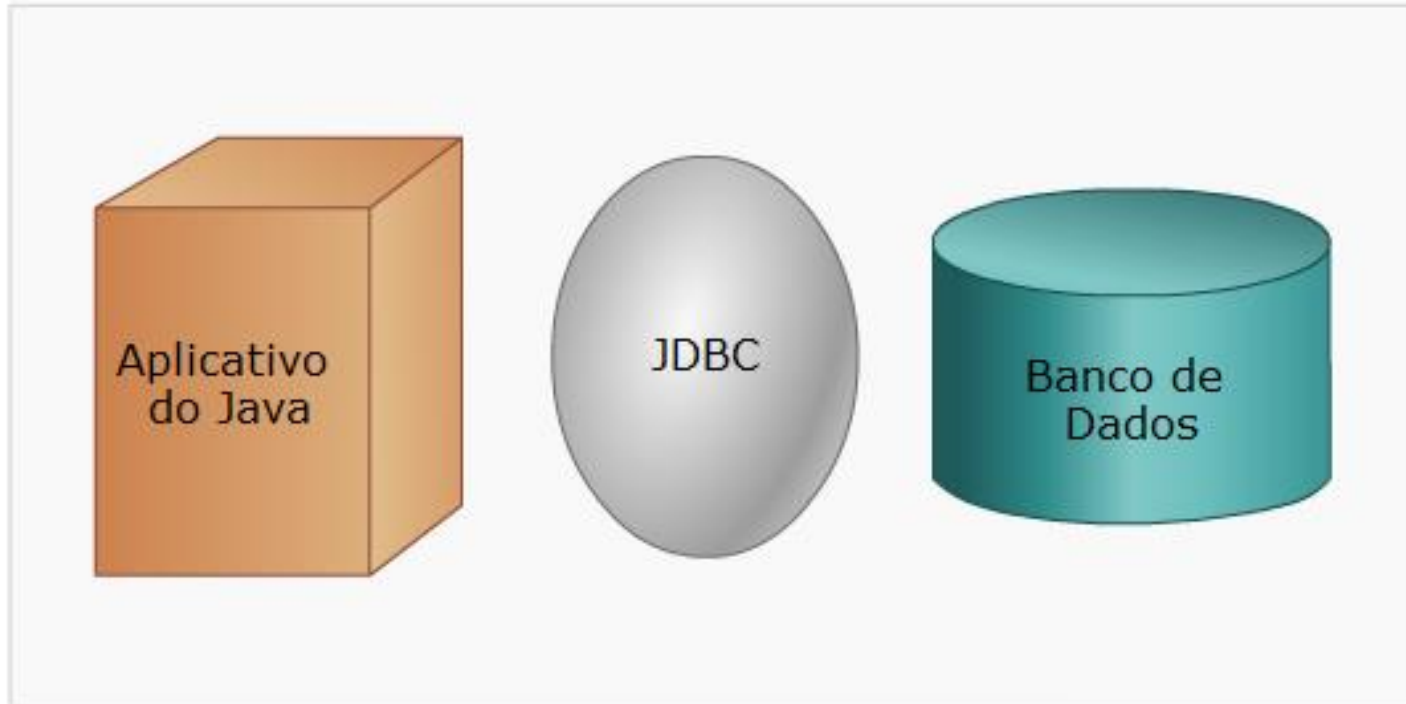


# Conexão com Banco de Dados usando Java

# ARQUITETURA



# ARQUITETURA

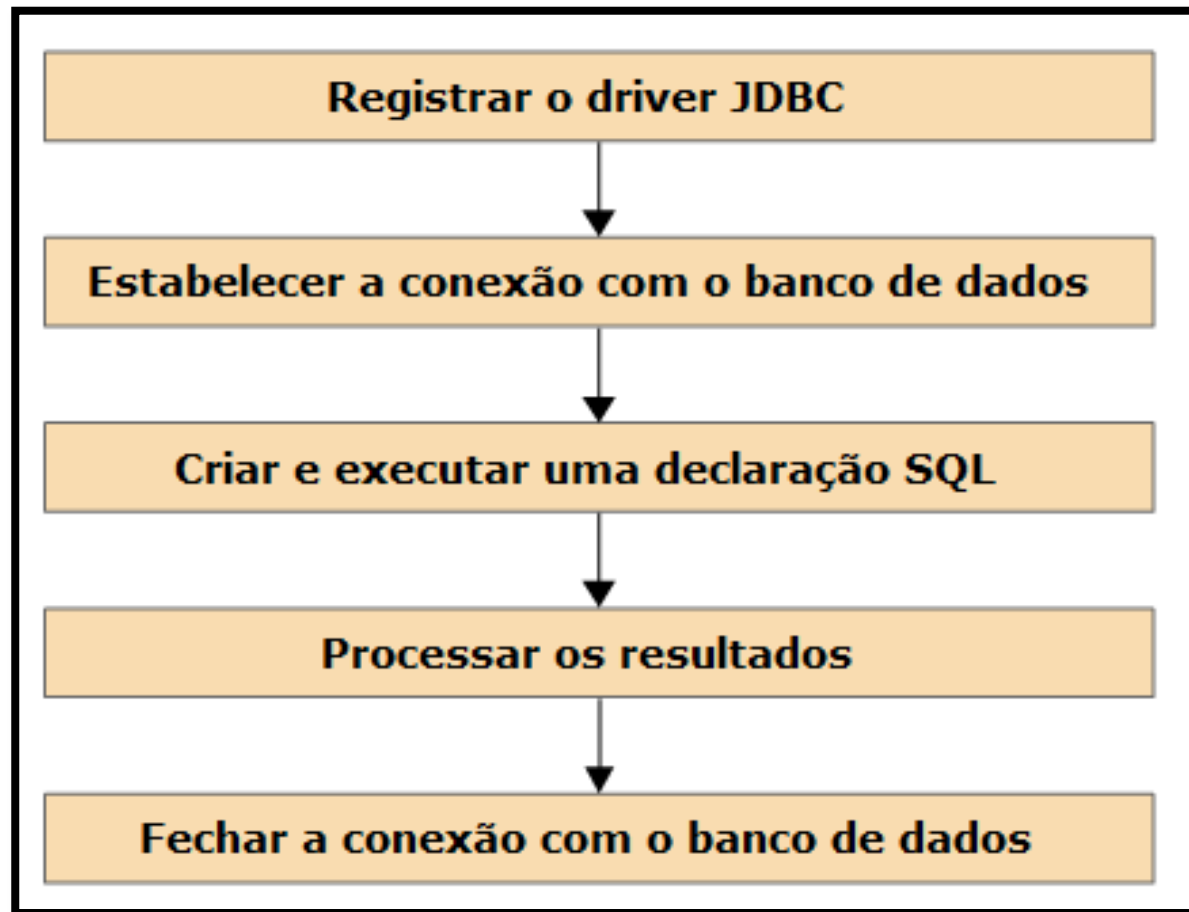


Java Database Connectivity (JDBC) é uma API do Java

# ARQUITETURA

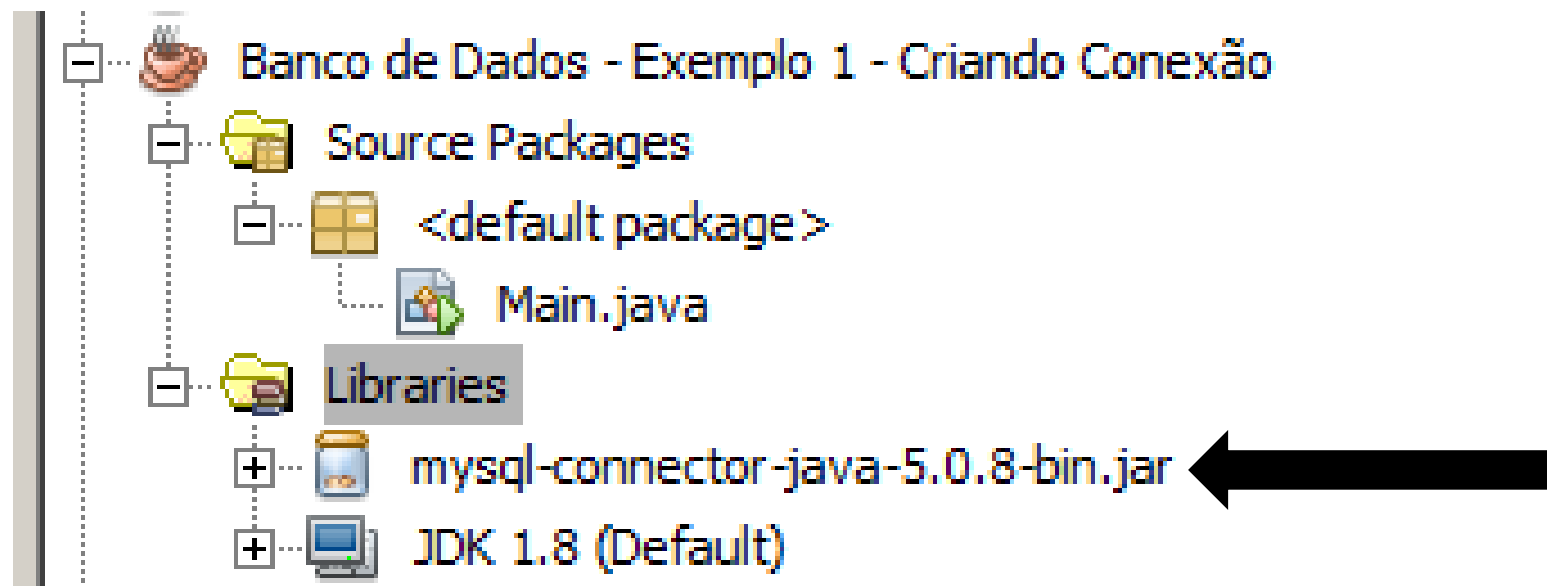
A API JDBC possui **um conjunto de classes** para acessar dados tabulares. Essas classes são escritas na linguagem de programação Java e fornecem uma API padronizada para os desenvolvedores de bancos de dados. Para acessar dados de modo rápido e eficiente nos bancos de dados, os aplicativos do Java usam a JDBC.

# PROCESSO DE DESENVOLVIMENTO DE APLICATIVOS USANDO JDBC



# INCLUIR O DRIVER JDBC

Incluir o driver JDBC (arquivo JAR) no projeto.



# ESTABELECEER CONEXÃO

- Registrar o driver JDBC através da classe **Class**.
- “Pegar” a conexão através do método **getConnection()**. Necessário passar os parâmetros do banco de dados: **URL, Usuário e Senha**.

# Exemplo

- Criar no banco de dados um esquema "aula2\_escola". Criar a tabela "alunos" com as seguintes colunas: id, nome, idade e nota.
- OBS: id -> PK e AI



# Exemplo – Cadastrar aluno

```
//1 - Registrar o JDBC
```

```
Class.forName("com.mysql.jdbc.Driver");
```

```
//2 - Conexão com o BD
```

```
String esquema = "aula2_escola";
```

```
String usuario = "root";
```

```
String senha = "bancodedados";
```

```
String local = "jdbc:mysql://localhost/"+esquema;
```

# Exemplo – Cadastrar aluno

```
//3 - Criar query
```

```
String query = "INSERT INTO alunos (nome, idade, nota) VALUES (?, ?, ?)";
```

```
//4 - Configurar a query
```

```
stmt = conexao.prepareStatement(query);
```

```
stmt.setString(1, "Marcos"); //nome
```

```
stmt.setInt(2, 12);           //idade
```

```
stmt.setDouble(3, 8.5);       //nota
```

# Exemplo – Cadastrar aluno

```
//5 - Executar a query  
stmt.execute();  
System.out.println("Cadastrou o aluno com sucesso!!!");  
  
//6 - Fechar o Banco de Dados  
stmt.close();  
conexao.close();  
System.out.println("Fechou o banco de dados.");
```

# Exercício 1 – Criar método

- Criar o método `cadastrarAluno()` dentro da Classe `AlunoDAO`.
- dica: `public void cadastrarAluno(String nome, idade, nota)`

## **Exercício 2 – Criar método professor**

- Criar no mesmo esquema "aula2\_escola", a tabela "professores" com as seguintes colunas: id, nome, idade, salario e materia.
- OBS: id -> PK e AI
- Criar o método cadastrarProfessor() dentro da Classe ProfessorDAO

# Exemplo – Consultar Aluno

```
//1 - Registrar o JDBC
```

```
Class.forName("com.mysql.jdbc.Driver");
```

```
//2 - Conexão com o BD
```

```
String esquema = "aula2_escola";
```

```
String usuario = "root";
```

```
String senha = "bancodedados";
```

```
String local = "jdbc:mysql://localhost/"+esquema;
```

# Exemplo – Consultar Aluno

```
//3 - Criar query
```

```
String query = "SELECT *FROM aluno WHERE id = (?)";
```

```
//4 - Configurar a query
```

```
stmt = conexao.prepareStatement(query);
```

```
stmt.setInt(1, 4); //consulta o id = 4
```

# Exemplo – Consultar Aluno

```
//5 - Executar a query
rs = stmt.executeQuery();
int id2;
String nome2 = null;
int idade2 = 0;
double nota2 = 0;
while(rs.next())
{
    id2 = rs.getInt("id");
    nome2 = rs.getString("nome");
    idade2 = rs.getInt("idade");
    nota2 = rs.getDouble("nota");
}
```



## Exemplo – Consultar Aluno

```
//6 - Processar os resultados
System.out.println("Nome: "+nome2);
System.out.println("Idade: "+idade2);
System.out.println("Nota: "+nota2);

//7 - Fechar banco de dados
stmt.close();
rs.close();
conexao.close();
System.out.println("Fechou banco de dados.");
```

## Exercício 3 – Consultar aluno

- Criar o método `consultarAlunoPelold()` dentro da Classe `AlunoDAO`.
- dica: `public void consultarAlunoPelold(int id)`

## **Exercício 4 – Criar métodos professor**

- **Na Classe Main:**
- **int opcao = 1;**
- **int id = 3;**
- **1 – Mostra todas as informações do professor**
- **2 – Mostra apenas o nome do professor**
- **3 – Verifica se o professor tem ou não mais de 30 anos**
  
- **Na Classe ProfessorDAO**
- **mostrarTodasAsInformacoes(int id)**
- **mostrarApenasNome(int id)**
- **verificarIdadeMaior30(int id)**