



Estácio

**UNIVERSIDADE ESTÁCIO DE SÁ
POLO CRISTO REI - CURITIBA**

**DESENVOLVIMENTO FULL STACK
MISSÃO PRÁTICA NÍVEL 3
LEONARDO CARVALHO MAGALHÃES**

Título da Prática

Mapeamento Objeto-Relacional e DAO e utilizando CRUD em JAVA.

Objetivo da Prática

O objetivo desta prática é implementar operações CRUD (Create, Read, Update, Delete) utilizando Java e JDBC para manipulação de dados em um banco de dados SQL Server.

Resultados da Execução dos Códigos:

Os resultados da execução dos códigos serão apresentados em seguida, conforme as consultas solicitadas.

Análise e Conclusão:

Qual a importância dos componentes de middleware, como o JDBC?

Os componentes de middleware, como o JDBC (Java Database Connectivity), são fundamentais para a comunicação entre aplicações Java e bancos de dados. Eles fornecem uma API consistente para acesso a dados, permitindo que os desenvolvedores escrevam código que possa se conectar a diferentes tipos de bancos de dados de forma transparente. Isso aumenta a portabilidade e a flexibilidade das aplicações.

Qual a diferença no uso de Statement ou PreparedStatement para a manipulação de dados?

- **Statement:** É utilizado para executar instruções SQL estáticas e simples. Ele não previne SQL Injection, pois os dados são diretamente inseridos nas instruções SQL.

- **PreparedStatement:** É utilizado para executar instruções SQL parametrizadas, oferecendo maior segurança contra SQL Injection, já que os parâmetros são definidos separadamente da instrução SQL.

Como o padrão DAO melhora a manutenibilidade do software?

O padrão DAO (Data Access Object) melhora a manutenibilidade do software ao separar a lógica de acesso a dados da lógica de negócios. Isso facilita a modificação e a extensão da aplicação, uma vez que mudanças no banco de dados ou nas operações de acesso a dados não afetam outras partes do sistema.

Como a herança é refletida no banco de dados, quando lidamos com um modelo estritamente relacional?

Em um modelo relacional estritamente, a herança pode ser refletida de diferentes formas:

- **Tabela única:** Uma única tabela armazena todos os tipos de entidades, com colunas adicionais para discriminar os tipos específicos.
- **Tabelas separadas:** Tabelas diferentes são criadas para cada classe base e derivada, com chaves estrangeiras para manter o relacionamento.
- **Tabela por classe concreta:** Cada classe concreta tem sua própria tabela, armazenando todos os dados necessários