Missão prática | Nível 1 | Mundo 3



CAMPUS: POLO DISTRITO JK - ANÁPOLIS - GO

CURSO: DESENVOLVIMENTO FULL STACK

NÚMERO DA TURMA: 2024.3

SEMESTRE LETIVO: Segundo semestre de 2024

ALIUNO: Henrique Rodrigues Rabello Vieira

MATRÍCULA: 202301230527

# **RPG0016 - BackEnd sem banco não tem**

Objetivos:

- Implementar persistência com base no middleware JDBC.

- Utilizar o padrão DAO (Data Access Object) no manuseio de dados.

- Implementar o mapeamento objeto-relacional em sistemas Java.

- Criar sistemas cadastrais com persistência em banco relacional.

# Primeiro procedimento:

## Códigos solicitados:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Uma imagem contendo Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

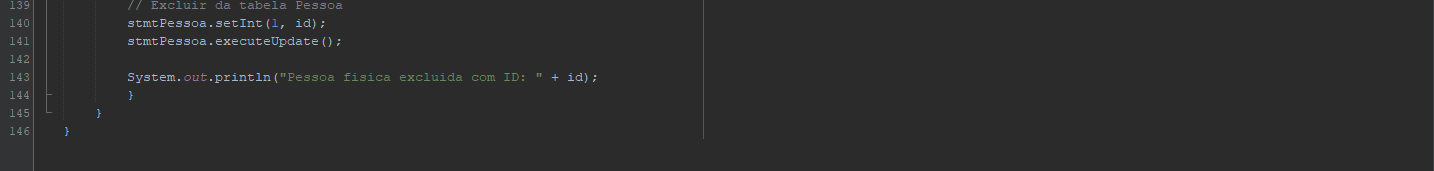
Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente



Texto

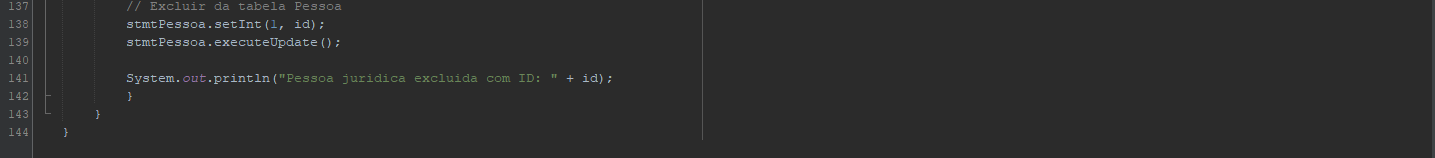
Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente



Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente



## Resultados:

Forma

Descrição gerada automaticamente com confiança baixa

Uma imagem contendo Forma

Descrição gerada automaticamente

Forma

Descrição gerada automaticamente com confiança média

## Análise e conclusão:

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

O JDBC é importante para a comunicação entre a aplicação JAVA e o banco de dados .

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

O preparedStatement é mais seguro e eficiente que statement, pois ele evita a injeção de SQL e permite a definição de paramêtros.

Como o padrão DAO melhora a manutenibilidade do software?

Melhora separando a lógica de acesso da lógica de negócios.

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

Pode ser feita de diversas formas, todas as classes compartilhando uma tabela, cada classe ter sua própria tabela, tendo apenas campos exclusivos e uma chave estrangeira para a tabela da superclasse, ou cada subclasse ter sua tabela o que impacta diretamente o sistema, cada um tendo as suas vantagens.

# Segundo procedimento:

## Códigos solicitados:

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

## Resultados:

Forma

Descrição gerada automaticamente com confiança média

Forma

Descrição gerada automaticamente com confiança média

Forma

Descrição gerada automaticamente com confiança média

Forma

Descrição gerada automaticamente com confiança média



Tela de computador com jogo

Descrição gerada automaticamente

## Análise e conclusão:

Quais as diferenças entre a persistência em arquivo e a persistência em banco de dados?

A persistência em arquivo consiste em armazenar dados em arquivos, podendo ser em xml, json e até arquivos binários. Já em banco de dados as informações são armazenadas de forma estruturada, oferecendo vários recursos e sendo mais adequada para aplicativos em grande escala.

Como o uso de operador lambda simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java?

Reduzindo a necessidade de classes anônimas, simplificando a sintaxe para definir comportamentos que podem ser passados como argumentos para métodos ou armazenados em variáveis.

Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static?

Os métodos chamados pelo main precisam ser estáticos porque são chamados no contexto da classe, não de uma instância, podendo ser chamados sem ter que criar um objeto para instanciar a classe.