Programação Orientada a Objetos em Java   
3ª Avaliação da Nota Final.

Davi de Almeida Souza, Gabriele Camili Nascimento da Silva

Universidade Estácio de Sá (NITERÓI I)   
24020-340 – Niterói – RJ – Brasil

Davi190203@gmail.com, oclimaar@gmail.com

**Abstract.** This article aims to develop a prototype of a registration system in Java language with the four basic CRUD operations in a database. Named as "SuaWishList," the objective of this database is to create a wishlist of various products, where users can store the name and price of one or more items and organize them into lists that specify the type of the product. The article will present the main stages of development, emphasizing the implemented functionalities and the benefits of this logistics system.

**Resumo.** Este artigo busca desenvolver um protótipo de sistema de cadastro em Linguagem Java com as quatro operações básicas de CRUD em banco de dados. Seu nome é “SuaWishList”e o objetivo desse banco de dados é criar uma lista de desejos de vários produtos dos quais o usuário pode armazenar nome e preço de um ou mais itens e organizá-los em listas que especificaram o tipo deste produto. Neste artigo, serão apresentadas as etapas principais do desenvolvimento, enfatizando as funcionalidades implementadas e os benefícios desse sistema de logística.

# 1. Introdução

O nome escolhido para o programa foi “SuaWishList”, visando que o programa irá justamente criar uma lista de desejos para os itens que podem ser adquiridos futuramente

# 2. Objetivo do Sistema

O objetivo do sistema é facilitar a organização de compras futuras, fazendo com que o usuário possa separar suas compras futuras em listas específicas que categorizaram os produtos presentes nelas, fazendo com que a organização pessoal e financeira seja facilitada ao máximo.

# 3. Requisitos do Sistema

Junto com as funcionalidades de CRUD, a SuaWishList pode criar listas diversas, adicionar ou modificar itens a listas específicas que armazenaram nome e preço de cada item, ler suas listas passadas e deletá-las a qualquer momento.(Ex: Lista: Automóveis, Nome: Fiesta, Preço: 200.000)

# 4. Casos de Utilização

A SuaWishList apresenta vários casos de uso, visando atender as necessidades de organização de compras futuras. Alguns exemplos possíveis são: listas de compras, materiais escolares, carros, jogos, materiais de construção, entre outras possíveis formas de armazenar itens de um determinado tipo.

# 5. Explicação do Código

A classe ‘**Application’** é a classe principal que contém a lógica do sistema de gerenciamento de listas. Ela possui uma conexão (‘**Connection’**) com o banco de dados e um objeto ‘**Statement’** para executar consultas SQL.

O método ‘**clear()’** é responsável por limpar a tela do console, tanto em sistemas Windows quanto em outros sistemas operacionais.

O método ‘**FrontEnd()**’ é o ponto de entrada do programa. Ele exibe um menu para o usuário e espera a entrada da escolha. Dependendo da escolha do usuário, ele chama os métodos correspondentes, como **‘criar\_lista()’, ‘inserir\_dados()’, ‘ler\_dados()’, ‘atualizar\_dados()’** e **‘deletar\_lista()’**.

O método **‘criar\_lista()**’ permite que o usuário crie uma nova lista. Ele solicita o nome da lista, cria uma tabela com esse nome no banco de dados e redireciona o usuário para inserir os dados nessa lista.

O método **‘inserir\_dados()’** permite que o usuário insira dados em uma lista existente. Ele mostra as listas disponíveis no banco de dados e solicita o nome da lista na qual o usuário deseja inserir os dados. Em seguida, solicita o nome e o preço do item a ser inserido e executa uma consulta SQL para adicionar esses dados à tabela correspondente.

O método **‘ler\_dados()’** permite que o usuário leia os dados de uma lista existente. Ele mostra as listas disponíveis no banco de dados e solicita o nome da lista que o usuário deseja ler. Em seguida, executa uma consulta SQL para recuperar os dados da tabela correspondente e os exibe no console.

O método **‘atualizar\_dados()’** permite que o usuário atualize os dados de uma lista existente. Ele mostra as listas disponíveis no banco de dados e solicita o nome da lista que o usuário deseja atualizar. Em seguida, exibe os dados atuais da tabela correspondente e solicita o ID da linha que deseja atualizar, o nome da coluna a ser atualizada e o novo valor. Por fim, executa uma consulta SQL para atualizar os dados na tabela.

O método **‘deletar\_lista()’** permite que o usuário exclua uma lista existente. Ele mostra as listas disponíveis no banco de dados e solicita o nome da lista que o usuário deseja excluir. Em seguida, executa uma consulta SQL para excluir a tabela correspondente do banco de dados.

Em relação ao banco de dados, o código utiliza o **‘java.sql’** para se conectar a um banco de dados PostgreSQL. São executadas consultas SQL para criar tabelas, inserir dados, ler dados, atualizar dados e excluir tabelas.

No código da SuaWishList, também é feita uma iteração sobre os metadados do banco de dados para listar as tabelas disponíveis.

A escolha dos elementos utilizados no código, como as classes e métodos, é baseada nas funcionalidades necessárias para um sistema de gerenciamento de listas. São utilizadas classes do pacote **‘java.sql’** para a interação com o banco de dados e métodos para executar as operações de criação, inserção, leitura, atualização e exclusão de dados nas tabelas.

# 6. Conclusão

A implementação do código apresenta um sistema básico de gerenciamento de listas usando um banco de dados relacional. Ele oferece funcionalidades para criar uma nova lista, inserir dados nessa lista, ler os dados, atualizá-los e excluir listas.

No geral, o código implementado oferece uma solução básica para o problema de gerenciamento de listas usando um banco de dados relacional. No entanto, existem várias áreas de melhoria que podem ser consideradas para tornar o sistema mais robusto, seguro e fácil de usar.