

ESPECIFICAÇÃO DE TRABALHO PRÁTICO

Análise e Desenvolvimento de Sistemas
INF008 – Programação Orientada a Objetos
Professor: Sandro Santos Andrade

1. Objetivo Geral

Desenvolver um sistema orientado a objetos em Java que realize o controle de produtos, usuários e pedidos para um site de comércio eletrônico. O trabalho visa consolidar os conceitos de Programação Orientada a Objetos (POO), incluindo encapsulamento, herança e *collections*, bem como boas práticas de *design* e estruturação de software.

2. Requisitos Funcionais

O sistema deve implementar as seguintes funcionalidades:

- Gerenciamento de produtos:
 - Cadastro e obtenção de produtos.
 - Cada produto deve conter:
 - ID único (gerado automaticamente).
 - Nome.
 - Descrição.
 - Preço.
 - Quantidade em estoque.
 - Categoria (ex.: Eletrônicos, Roupas, Alimentos). Utilize o tipo String para este atributo.
- Gerenciamento de usuários:
 - Cadastro de usuários.
 - Existem dois tipos de usuários:
 - Clientes: podem realizar compras.
 - Administradores: gerenciam produtos e usuários e acessam relatórios.
 - Informações comuns a todos os usuários:
 - ID único (gerado automaticamente).

- Nome completo.
 - Email (único).
 - Senha (armazenada de forma segura).
- Informações adicionais para clientes:
 - Endereço de entrega.
 - Histórico/lista de compras.
- Processamento de pedidos:
 - Clientes podem adicionar produtos a um carrinho de compras.
 - O sistema deverá apresentar os ids, nomes e preços dos produtos e o usuário indicará o id do produto que deseja inserir no carrinho de compras.
 - Finalização do pedido:
 - Atualiza o estoque dos produtos.
 - Gera um número de pedido único.
 - Calcula o total do pedido.
 - Associa o pedido ao cliente no histórico de compras.
 - Pedidos devem armazenar:
 - Número do pedido (gerado automaticamente).
 - Data do pedido.
 - Lista de produtos comprados (com quantidade de cada).
 - Total do pedido.
- Relatórios (acesso apenas por usuários administradores):
 - Maior pedido realizado: encontrar e exibir os dados do pedido com maior valor total. Caso haja mais de um pedido com o mesmo maior valor total apresente o primeiro encontrado.
 - Produto com menor estoque: encontrar e exibir os dados do produto com menor quantidade em estoque. Caso haja mais de um pedido com o mesmo menor estoque apresente o primeiro encontrado.
- Menus e submenus a serem criados:
 - Para ambos os usuários (primeiro menu apresentado):
 - Login
 - Solicitar usuário e senha e realizar a validação.
 - O sistema deve ser entregue com um único usuário "admin" (com senha "admin") criado.
 - Para usuários administradores:
 - Create new product
 - Create new user

- Deverá solicitar se o novo usuário sendo criado é um cliente ou um administrador, seu login e sua senha.
- Report – more expensive order
- Report – product with lowest inventory
- Exit
- Para usuários clientes:
 - Start new order
 - Add product
 - View shopping cart
 - Listar todos os produtos incluídos no carrinho de compras
 - Finish order
 - Exit

3. Requisitos Técnicos

- O sistema deve ser modular e utilizar classes e heranças seguindo os princípios de POO.
- Todo o código-fonte do sistema deve estar em inglês.
- Classes obrigatórias (outras podem ser criadas conforme necessário):
 - Product
 - User (classe base) – esta classe não deverá poder ser instanciada na aplicação
 - Customer
 - Administrator
 - Order
 - ShoppingCart
- Persistência:
 - O sistema deverá serializar os dados na finalização e desserializar os dados na inicialização.
- Autenticação e Autorização:
 - A funcionalidade de login deverá detectar o tipo de usuário em questão e apresentar, em seguida, o menu correspondente.
- Interface com o Usuário:
 - Deverá ser desenvolvida em modo texto, com acesso simples através de números (ex: 1 – Create new product; 2 – Create new user).

Boas Práticas:

- Use nomes significativos para variáveis, métodos e classes.
- Divida a lógica em métodos pequenos e reutilizáveis.
- Siga as convenções do programação do próprio Java.
- Lembre-se que uma solução simples é sempre melhor.

4. Regras de Negócio

- Produtos não podem ser vendidos caso o estoque esteja zerado.
- Um pedido deve conter pelo menos um produto.
- O sistema deve evitar IDs duplicados para qualquer entidade.

5. Critérios de Avaliação

Implementação das funcionalidades (70%) - o sistema atende aos requisitos especificados:

- Login (5%)
- Cadastro de usuários (5%)
- Cadastro de produtos (5%)
- Criação de pedido e inclusão de produtos (25%)
- Relatórios (25%)
- Serialização/Desserialização (5%)

Uso de conceitos de POO (20%): correta aplicação de herança, *collections*, construtores, atributos e métodos, bom desempenho e encapsulamento.

Organização e boas práticas (10%): código simples, bem estruturado, seguindo convenções e bem modularizado.

6. Entregáveis

- Código-fonte do sistema em formato *.zip* ou *.tar.gz*.
 - **OBS:** o pacote *.zip* ou *.tar.gz* não deve conter arquivos *.class*, apenas arquivos *.java*. Limpe o seu projeto antes de criar o pacote.
- Arquivo README.md com:
 - Instruções de compilação e execução.
 - **OBS:** a compilação e execução do projeto não deve requerer a instalação de nenhuma IDE. Se você não quiser utilizar nenhuma *build tool*, como o *maven* ou *gradle*, informe como compilar e executar os arquivos manualmente com os comandos *javac* e *java*. Caso utilize alguma *build tool*, informe os comandos necessário para compilar e executar sua aplicação.

7. Prazo e Forma de Entrega

- O trabalho deve ser enviado até o dia **08/12/2024**.
- O código-fonte e o arquivo README.md deve ser enviado por email para sandroandrade@ifba.edu.br, com o assunto "INF008 T1 <seu-nome-completo>", sem as aspas.
- Trabalhos enviados com outro assunto não serão corrigidos

8. Informações Importantes

- Todos os códigos-fonte entregues serão checados por plágio utilizando as ferramentas **MOSS** (Measure of Software Similarity - <https://theory.stanford.edu/~aiken/moss/>) e **JPlag** (<https://github.com/jplag/JPlag>). Desenvolva sua própria solução, é sua chance de aprender.
- Haverá um avaliação subsequente para elucidação de dúvidas sobre o código-fonte que você desenvolveu. O formato e data serão ainda definidos.
- Todas as dúvidas sobre o trabalho deverão ser abertamente discutidas no grupo da disciplina no Telegram.

Bom trabalho!