

TRABALHO PARA NOTA – C3.A2

OBJETIVO DO TRABALHO:

- Aplicar o conhecimento sobre Árvore Binária de Pesquisa desenvolvendo uma aplicação computacional referente a uma área de negócio.

INSTRUÇÕES:

- Valor do Trabalho: **8,0 pontos**;
- Implemente a aplicação na linguagem de programação Java, utilizando a estrutura de dados identificada no objetivo;
- O trabalho deve ser feito em grupo com até 3 integrantes ou individualmente;
- Poste o trabalho no AVA com todas as classes desenvolvidas em arquivos separados (.java), pode compactar as classes/pacotes;
- Data de Limite para postagem: **26/06/2025 às 18:50**.
- Apresentação do trabalho para a professora: **26/06/2025 durante a aula**.

Atenção: o trabalho terá menos 0,5 ponto por dia de atraso a partir da data limite.

COMPOSIÇÃO DA NOTA:

- Entrega do trabalho na data especificada;
- Execução de todas as funcionalidades exigidas nas especificações de implementação (ver abaixo). Cada erro encontrado na execução do programa equivalerá a 0,5 ponto descontado na nota;
- Arguição individual sobre o funcionamento da aplicação caso seja necessário;
- A pontuação está informada em cada operação a ser realizada (ver abaixo);
- Verifique todas as consistências necessárias para o bom funcionamento dos métodos das classes, apresentando mensagens de erro e/ou informativa;
- **O trabalho só será avaliado com apresentação para a professora. Trabalhos somente postados no AVA não serão avaliados.**

Atenção: Não deixe seu código fonte nas máquinas dos laboratórios, porque alguém pode copiá-lo. Caso isto acontecer, isto é, sejam apresentados programas similares/idênticos, eles não serão aceitos para a avaliação.

ESPECIFICAÇÕES DE IMPLEMENTAÇÃO:

Gerenciamento de Produtos de Supermercado

Crie um aplicativo para gerenciar os produtos de um supermercado.

1. Acesso ao Sistema (Segurança Básica)

Ao iniciar o programa, o usuário deve informar um **código de acesso** e uma **senha** para ter acesso. Apenas será permitido o acesso se o código for 123456 e a senha for admin. Qualquer outra combinação impedirá a entrada no sistema.

2. Funcionalidades de Gerenciamento de Produtos

O aplicativo do supermercado deve permitir as seguintes operações sobre os produtos, utilizando uma **Árvore Binária de Pesquisa (ABP)** como estrutura de armazenamento principal:

a) **Cadastrar Produto (1,0 ponto)** O usuário deverá cadastrar as seguintes informações:

- **Nome do Produto:** (Ex: "Arroz", "Sabão em Pó", "Maçã")
- **Código de Barras:** (Identificador numérico ou alfanumérico único)
- **Preço Unitário:** (Valor monetário)
- **Quantidade em Estoque:** (Número de unidades disponíveis)
- **Categoria:** (Ex: "Mercearia", "Limpeza", "Hortifruti", "Bebidas", "Laticínios")

O **Nome do Produto** será a chave de busca principal na árvore e não poderá ser duplicado. O **Código de Barras** também não poderá ser duplicado entre os produtos. Antes de adicionar um novo produto, o sistema deve verificar se o Nome do Produto ou o Código de Barras já existem na árvore. Se qualquer um dos dois for encontrado, a inclusão não deve ser permitida.

b) Consultar Dados de um Produto (1,0 ponto) Esta opção deve exibir todos os dados de um produto específico, identificando-o pela chave de busca (o **Nome do Produto**).

c) Contagem por Categoria (1,0 ponto) Implemente um método para contar quantos produtos pertencem a cada **Categoria** cadastrada (ex: 5 produtos em "Mercearia", 3 em "Limpeza", etc.) e exibir essa contagem.

d) Listar Produtos por Faixa de Preço (1,0 ponto) Adicione uma função para exibir os produtos que se encontram dentro de um determinado intervalo de **Preço Unitário** (ex: produtos que custam entre R\$ 5,00 e R\$ 15,00).

e) Produto Mais Barato / Mais Caro (1,0 ponto) Crie funções para encontrar e exibir os dados do produto com o **menor Preço Unitário** e do produto com o **maior Preço Unitário**.

f) Excluir Produto (1,0 ponto) Implemente a funcionalidade para remover um produto da árvore. A remoção em árvores binárias de pesquisa tem casos específicos que desafiam o algoritmo (nó folha, nó com um filho, nó com dois filhos). A busca para exclusão será pelo **Nome do Produto**.

g) Atualizar Dados do Produto (1,0 ponto) Permita a modificação de informações de um produto já cadastrado (**Preço Unitário**, **Quantidade em Estoque**, **Categoria**). O **Nome do Produto** e o **Código de Barras** devem ser mantidos como identificadores únicos e, portanto, inalteráveis por esta função (para alterá-los, seria necessário excluir e cadastrar novamente).

3. Experiência do Usuário (UX) e Validações (1,0 ponto)

- **Validação de Entrada:** Adicione validações robustas para todas as entradas do usuário (por exemplo, garantir que o Código de Barras tenha um formato esperado, que o Preço Unitário e a Quantidade em Estoque sejam números positivos/não negativos, que a categoria não seja vazia etc.).
- **Mensagens de Erro e Sucesso:** Forneça feedback claro ao usuário sobre o resultado de cada operação (ex: "Produto 'Arroz Integral 1kg' cadastrado com sucesso!", "Erro: Nome do Produto já existe!", "Produto não encontrado para consulta.").

4. Estrutura de Classes (Organização dos Arquivos)

As classes essenciais para a implementação são:

- **Classe Produto:** Contém os atributos de dados do produto (Nome, Código de Barras, Preço, Estoque, Categoria).
- **Classe No (ou NoProduto):** Representa um nó da árvore binária, contendo o dado do produto (Produto) e referências para os nós filhos (esquerda e direita).
- **Classe ArvoreBinariaPesquisa (ou ArvoreProdutos):** Contém os atributos e métodos para gerenciar a árvore de produtos.
- **Classe Menu (ou SistemaSupermercado):** Gerencia a interação com o usuário, apresentando as opções e chamando os métodos correspondentes da ArvoreBinariaPesquisa.

Atenção: As linhas de códigos têm que estar indentadas. Caso o trabalho não esteja funcionando corretamente, escreva uma mensagem na sua postagem do trabalho informando o problema existente.

Bom trabalho, bom estudo!