



## Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências da Computação

SCC0122 - Estruturas de Dados (2025)

Professora: Mirela Teixeira Cazzolato

Monitores: Lucas Lombardi Castro e Leonardo Rossi Luiz

# Trabalho 09 - Gerenciador de Catálogo de Produtos com AVL

## 1. Introdução

Uma loja e-commerce está desenvolvendo um novo sistema para organizar seu catálogo de produtos. Para garantir uma boa performance nas operações de busca, inserção e remoção de produtos, a equipe de desenvolvimento decidiu evitar o uso de listas lineares, que são ineficientes à medida que a quantidade de produtos aumenta.

A principal necessidade do sistema é organizar os produtos com base em um código de identificação único (id), permitindo que as operações sejam realizadas de forma otimizada. A estrutura de dados que melhor se adapta a esse requisito é a **Árvore AVL**, vista em sala de aula, que é uma Árvore de Busca Auto Balanceável. Esta estrutura garante que as operações de busca, inserção e remoção tenham complexidade  $O(\log N)$  no pior caso.

Sua equipe foi contratada para desenvolver o protótipo deste sistema de gerenciamento de catálogo.

## 2. Descrição do Problema

O programa deverá ler uma sequência de comandos, um por linha, para manipular um catálogo de produtos. Cada produto terá um código (inteiro), um nome (string sem espaços) e um preço (float).

O catálogo deve ser mantido em uma AVL, onde a chave de ordenação é o **código do produto**. A principal diferença é que neste trabalho, a árvore deve ser mantida balanceada a todo momento. Após cada operação de inserir ou remover, a estrutura deve verificar se alguma rotação é necessária.

Objetivo	Comando	Descrição da Operação
<b>Inserir Produto</b>	inserir <codigo> <nome> <preco>	Lê os dados do produto e o insere na árvore. Se um produto com o mesmo código já existir, uma mensagem de erro deve ser mostrada ao usuário.

<b>Remover Produto</b>	remover <codigo>	Remove o produto com o código especificado. Se o produto for removido, imprime uma mensagem de confirmação. Se não for encontrado, informa o usuário.
<b>Buscar Produto</b>	buscar <codigo>	Busca pelo produto com o código especificado. Se encontrado, exibe todos os seus dados. Se não existir, informa o usuário.
<b>Listar em Pré-Ordem</b>	preordem	Exibe todos os produtos no catálogo, seguindo o percurso em pré-ordem (Raiz, Esquerda, Direita).
<b>Listar em Ordem</b>	inordem	Exibe todos os produtos em ordem crescente de código, seguindo o percurso em ordem (Esquerda, Raiz, Direita).
<b>Listar em Pós-Ordem</b>	posordem	Exibe todos os produtos no catálogo, seguindo o percurso em pós-ordem (Esquerda, Direita, Raiz).
<b>Encerrar</b>	sair	Encerra a execução do programa.

### 3. Especificações e Critério de Avaliação

Atenção! Para a avaliação do projeto, não será considerada apenas a nota do Run Codes, mas também o cumprimento das seguintes especificações:

- O código deve estar na linguagem C. É permitido o uso da biblioteca string.h.
- O código deve ser organizado usando um TAD para a **Árvore AVL** e um TAD para o **Produto** (contendo código, nome e preço).
- A **árvore** e o **produto** devem ser **alocados dinamicamente**, e deslocados pelo TAD.
- A legibilidade do código é um fator importante. Use nomes de variáveis e funções que sejam claras e comente o código de forma apropriada.
- A função usuário (main()) não deve manipular diretamente a estrutura de dados. Para criar, modificar ou desalocar o deque, utilize apenas as funções fornecidas pelo TAD.

### 4. Exemplos de Entrada e Saída

Exemplo de Entrada	Exemplo de Saída
inserir 10 ProdutoA 100 inserir 20 ProdutoB 200 inserir 30 ProdutoC 300 preordem	Produto [10, ProdutoA, 100.00] inserido. Produto [20, ProdutoB, 200.00] inserido. Produto [30, ProdutoC, 300.00] inserido. Produtos em pre-ordem:

emordem	- [20, ProdutoB, 200.00] - [10, ProdutoA, 100.00] - [30, ProdutoC, 300.00] Produtos em emordem:
sair	- [10, ProdutoA, 100.00] - [20, ProdutoB, 200.00] - [30, ProdutoC, 300.00]

Exemplo de Entrada	Exemplo de Saída
inserir 50 ProdutoA 500 inserir 20 ProdutoB 200 inserir 30 ProdutoC 300 preordem	Produto [50, ProdutoA, 500.00] inserido. Produto [20, ProdutoB, 200.00] inserido. Produto [30, ProdutoC, 300.00] inserido. Produtos em pre-ordem:
emordem	- [30, ProdutoC, 300.00] - [20, ProdutoB, 200.00] - [50, ProdutoA, 500.00] Produtos em emordem:
sair	- [20, ProdutoB, 200.00] - [30, ProdutoC, 300.00] - [50, ProdutoA, 500.00]

## 5. Submissão

1. **Envie** seu código fonte para o run.codes. Para isso, compacte (.zip), os arquivos, junto com um documento **Makefile**. (Será executado o comando “make all” para compilar seu código, e o comando “make run” para executar o seu código)
2. **Tire dúvidas!** Se tiver dúvida sobre algo, mande um email para os monitores, ou participe da monitoria presencial.
3. É permitido o uso da biblioteca string.h. Para usar essa biblioteca, não é necessário adicionar flags no Makefile.