

## Atividade: Árvore Binária de Busca

### Problema

Uma empresa de e-commerce armazena informações sobre seus produtos e quer otimizar a busca por eles. Cada produto possui um código numérico único que representa seu **ID**, além de outras informações como **nome**, **descrição** e **preço**.

Atualmente, os produtos estão armazenados em uma lista desordenada, e o sistema demora para encontrar informações específicas, especialmente quando o número de produtos é grande. A empresa quer melhorar o desempenho das buscas para proporcionar uma melhor experiência ao cliente.

### Objetivo

A empresa precisa de um sistema onde seja possível:

1. **Inserir** um novo produto de forma eficiente.
2. **Remover** um produto com base no seu ID.
3. **Buscar** um produto rapidamente pelo seu ID.

Além disso, o sistema deve permitir listar os produtos em ordem crescente de ID.

### Solução com Árvore Binária de Busca (BST)

Para resolver o problema, a empresa decide implementar uma **Árvore Binária de Busca (BST)**, onde cada nó da árvore representará um produto, e a chave usada para organizar a árvore será o ID do produto.

1. **Inserção**: Inserir um novo produto é feito ao percorrer a árvore para encontrar a posição correta, mantendo a propriedade de que, para cada nó, o valor de sua subárvore à esquerda é menor que o valor do nó e o da subárvore à direita é maior.

2. **Busca**: Buscar um produto por ID é rápido, já que basta navegar pela árvore comparando o ID procurado com os nós, seguindo para a esquerda ou para a direita conforme o valor é menor ou maior, respectivamente.

3. **Remoção**: A remoção de um produto pelo seu ID na BST é possível, com tratamento especial para diferentes casos (nó sem filhos, com um filho ou com dois filhos).

4. **Ordenação:** Para listar os produtos em ordem crescente, basta fazer uma travessia em-ordem (in-order traversal) da árvore, que visita os nós em ordem crescente.

#### **Exemplo**

1. O produto com **ID 30** é inserido na árvore.
2. O produto com **ID 20** é inserido e se torna o filho esquerdo do produto com ID 30.
3. O produto com **ID 40** é inserido e se torna o filho direito do produto com ID 30.
4. Quando a busca pelo **ID 20** é feita, o sistema começa pela raiz (ID 30), verifica que 20 é menor, e se move para o nó esquerdo, encontrando o produto rapidamente.

Usando uma Árvore Binária de Busca, a empresa consegue otimizar o tempo de busca, inserção e remoção para uma média de  **$O(\log n)$**  operações, melhorando a eficiência em relação a uma lista desordenada, especialmente em grandes volumes de produtos.