

ÁRVORES

DATA DE ENTREGA 07/05/2022 – NO MATERIAL DIDÁTICO – desconto se atrasar o envio no sistema.

Os alunos devem desenvolver, individualmente ou em grupo de até 2 pessoas, o sistema computacional descrito abaixo. Ao final devem ser entregues o código fonte desenvolvido, com todos os recursos presentes e um vídeo onde cada integrante apresenta a sua participação.

Descrição do Problema:

Considere que você é o responsável pela implementação do sistema de RH de uma empresa multinacional. Serão feitas muitas consultas ao sistema (por Nome e CPF) e deseja-se que todas as consultas sejam realizadas de forma rápida. Faça um programa que permita o cadastro, a remoção e a consulta de funcionários.

Utilize estruturas de dados do tipo Árvore Binária de Busca para armazenar e organizar os dados dos funcionários. Serão necessárias duas árvores: uma ordenada por Nome e uma por CPF. Entretanto, para evitar redundância de dados (informações iguais armazenadas em locais diferentes), o cadastro deverá ser feito em uma estrutura separada que será referenciada pelas árvores (*info). Utilize as estruturas abaixo:

```
struct Info{
    long int CPF;
    string Nome;
    string Profissao;
};

struct No{
    Info *info;
    No *esq;
    No *dir;
};

No *arvoreCPF;
No *arvoreNome;
```

Nessas estruturas não poderão ser cadastrados nomes ou cpf repetidos. Quando o usuário digitar um cadastro qualquer, deverá ser criado apenas uma única estrutura tipo Info. Ela deverá ser apontada pelas duas árvores.

Devem ser implementadas as seguintes operações:

- Inserir uma nova informação;
- Remover um registro pesquisando por CPF ou por nome;
- Pesquisar um registro por CPF ou por nome;
- Listar todos as informações cadastradas (como se fosse um relatório, mas em tela) de forma ordenada por nome ou cpf;
- Ao encerrar a aplicação, todas as posições/espacos de memória alocados devem ser liberados;

Quesitos:

- Código-fonte (8,5 pontos):
 - Correta implementação de todas as operações e funções indicadas;
 - Funcionamento do programa conforme enunciado;
 - O programa deve, necessariamente, fazer uso do conceito de estruturas de dados tipo árvore de busca no seu desenvolvimento.
 - Os índices de apoio a localização de cada informação devem ser armazenados em uma árvore binária de busca;
 - Estrutura lógica de programação (indentação, comentários, nomenclatura de termos, elementos de usabilidade, etc...);
 - Uso da linguagem C++;
- Vídeo (1,5 pontos)
 - Vídeo gravado e disponível para o acesso onde cada integrante do grupo mostra a sua participação no desenvolvimento do código, explicando o que desenvolveu em detalhes.
 - O vídeo poderá ser enviado junto em formato mp4 ou colocado em um stream de vídeo (youtube, vimeo ou outro) e enviado como link de acesso.
- **Nota Extra (1,0 pontos)**
 - Será atribuído um valor adicional na nota ao grupo que implementar essa aplicação utilizando como estrutura de suporte aos índices uma árvore AVL.

OBSERVAÇÕES:

- O código-fonte e o relatório devem ser postados no Material Didático em um arquivo **ZIP/RAR**. Outro formato terá desconto de 1 ponto. Se o professor não conseguir “abrir” o arquivo, a nota relativa ao código-fonte será 0.
- Cadastre a sua dupla/nome no momento do envio no Material Didático. Identifique os integrantes em um comentário no início do “main.cpp” (e outros arquivos de código-fonte).
- Não serão aceitos trabalhos após o dia de entrega.
- Trabalhos que não contenham uma (ou mais) das partes descritas acima (código fonte ou vídeo) serão avaliados somente pelas partes constantes.
- Em caso de necessidade ou alguma dúvida, o professor se reserva o direito de chamar o grupo ou um membro dele para realizar uma apresentação individual sobre o trabalho.
- Trabalhos com grau de similaridade terão a nota dividida pelo número de entregas similares. Dica: não repassem código-fonte para os colegas e não poste o código em sistemas de controle de versão publicamente.
- Caso a nota final, em função da nota extra, ultrapasse o valor máximo permitido, será atribuído esse valor a nota final.