

## Trabalho Prático 2 – Árvores Binárias

Prof. Mateus Ferreira Satler

DECSI – ICEA - UFOP

### Descrição

O objetivo deste trabalho é desenvolver um sistema de RH para uma empresa. Serão feitas diversas consultas ao sistema (por Nome e CPF) e deseja-se que todas as consultas sejam realizadas de forma rápida. Faça um programa na Linguagem C que permita o **cadastro**, a **remoção** e a **consulta** de funcionários.

Para isso, utilize obrigatoriamente a 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:

```
typedef struct {
    int CPF;
    char Nome[50];
    char Profissao[30];
}Info;

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

No *arvoreCPF;
No *arvoreNome;
```

**Observação:** Não permita que sejam cadastrados dois nomes ou CPFs iguais. Ao final do programa, libere todos os endereços de memória alocados. Ao remover ou adicionar um funcionário na árvore de CPF, não se esqueça de fazê-lo também da árvore de nomes, e vice-versa.

### Funcionalidade

Seu programa deve conter um **menu** com as seguintes opções:

1. Cadastrar um funcionário
2. Remover um funcionário pelo nome
3. Remover um funcionário pelo CPF
4. Buscar e exibir os dados de um funcionário pelo nome
5. Buscar e exibir os dados de um funcionário pelo CPF
6. Exibir os dados de todos os funcionários ordenados pelo Nome
7. Remover todos os funcionários
8. Sair

## O que deve ser entregue

Código fonte do programa em C (bem indentado e comentado) e documentação do trabalho. Entre outras coisas, a documentação deve conter:

1. **Introdução:** descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
2. **Implementação:** descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais sub-rotinas utilizadas, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. **Muito importante:** os códigos utilizados nas implementações devem ser inseridos na documentação.
3. **Listagem de testes executados:** os testes executados devem ser apresentados, analisados e discutidos.
4. **Conclusão:** comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
5. **Bibliografia:** bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização.
6. **Formato:** mandatoriamente em **PDF**.

## Como deve ser feita a entrega

A entrega deve ser feita via *Plataforma Moodle* na forma de um único arquivo em formato **.rar** ou **.zip**, contendo os arquivos do código e a documentação.

## Comentário Gerais

1. O trabalho é individual (grupo de **UM** aluno);
2. Comece a fazer este trabalho logo, pois o prazo para terminá-lo está tão longe quanto jamais poderá estar;
3. Clareza, indentação e comentários no programa também serão avaliados;
4. Trabalhos copiados (e FONTE) terão nota zero;  
Devido a recorrentes problemas com cópias de trabalhos (plágios), os autores de trabalhos copiados também terão todos os demais trabalhos zerados, como forma de punição e coação ao plágio acadêmico;
5. Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.