

**TRABALHO DE ESTRUTURAS DE DADOS II PARA A PRIMEIRA AVALIAÇÃO**

- 1) **(4,0 pontos)** Faça um programa em C de uma Biblioteca de Streams. As informações são organizadas em uma árvore binária pelo nome da stream. Para cada stream deve se ter: o nome da stream, site, endereço para uma lista de categorias de programas. Cada nó da lista deve conter os dados da categoria: tipo (notícias, esporte, entretenimento), nome da categoria, endereço para árvore binária de programas. Para árvore binária de programas de uma categoria deve se ter: nome do programa, periodicidade (diário, semanal, mensal,...), tempo do programa, horário de início, ao vivo ou sob demanda, nome apresentador. Além disso deve existir uma lista de apresentadores contendo o nome do apresentador, categoria que ele trabalha, stream que ele trabalha atualmente e um endereço para uma lista de streams que o mesmo já trabalhou, contendo o nome da stream e o período que trabalho na stream (data início, data término). Quando o usuário executar o programa o mesmo deve permitir:
- i) Cadastrar Stream: cadastrar dados de streams organizados em uma árvore binária pelo nome da stream, o usuário pode cadastrar uma stream a qualquer momento, não permita cadastro repetido.
  - ii) Cadastrar Categorias: cadastrar lista de categorias para uma determinada stream, onde os dados da Categoria de um determinada stream devem ser organizados em uma lista circular simples dinâmica pelo nome da categoria a qualquer momento, lembre-se uma categoria só pode ser cadastrada para uma stream já castrada e uma categoria não pode se repetir para uma mesma stream.
  - iii) Cadastrar Programas: cadastrar programas de uma categoria para uma determinada stream uma árvore binária organizada pelo nome do programa, lembre-se um programa só pode ser cadastrado para um a categoria que já existe e o programa não pode se repetir para uma mesma categoria. Além disso, o apresentador deve estar cadastrado na lista de apresentador e ser da mesma categoria e trabalhar na stream que o programa pertence.
  - iv) Cadastrar Apresentador: cadastrar apresentadores em uma lista ordenada dinâmica duplamente encadeada pelo nome do apresentador. Lembre-se que a stream que ele trabalha deve estar cadastrada. A lista de streams que ele já trabalhou pode ser um vetor ordenado pelo nome da stream, mas a mesma deve já ter sido cadastrada.
  - v) Mostrar todas as streams cadastradas.
  - vi) Mostrar todas as categorias cadastradas para uma determinada stream.
  - vii)Mostrar todos os programas de uma determinada categoria de uma determinada stream.
  - viii)Mostrar todas as streams que tem uma determinada categoria.
  - ix) Mostrar todos os programas de uma determinada stream que acontecem em um determinado dia e horário.
  - x) Mostrar todas as streams que tem um determinado tipo de categoria.
  - xi) Mostrar todos os programas de um determinado dia da semana de uma determinada categoria de uma stream.
  - xii)Mostrar todos os apresentadores de uma determinada stream.
  - xiii)Mostrar todos os apresentadores de um determinada categoria independente da stream que o mesmo trabalha.
  - xiv)Mostrar os dados de um determinado programa de uma categoria de uma stream.
  - xv) Permita remover um program de uma determinada categoria de uma stream.
  - xvi) Permita remover uma categoria de uma stream, só pode ser removida se não tiver nenhum programa cadastrado nela.
  - xvii)Permita alterar a stream que um apresentador trabalha atualmente. Lembre-se que não pode haver programa naquela stream apresentado pelo apresentador ( programa removido ou alterado o apresentador).

- 2) **(4,0 pontos)** Repita todo o processo dos exercícios 1 mas agora implemente o código usando uma Árvore AVL.
- 3) **(2,0 pontos)** Em relação ao exercício 1 e 2 verifique os tempos:
- (a) de inserção de uma determinada quantidade de elementos de um árvore de stream; (a quantidade de elementos mínima necessária de acordo com a máquina que está executando o programa.
  - (b) de busca de todos programas de uma categoria de uma determinada stream (no item vii);

Para verificar os tempos de execução pesquise pela função time do C. Insira no código comandos para obter o tempo inicial antes de fazer a pesquisa e o tempo final depois de concluir a pesquisa.

Obs. 1: Lembre-se que não pode haver impressão entre o tempo inicial e o tempo final, pois impressão consome muito tempo.

Obs. 2: Para validar o tempo de busca, o mesmo deve ser repetido 30 vezes. Faça uma média para obter o resultado final.

**Equipe:** os programas podem ser feitos em dupla, mas **os relatórios são individuais**. Se os programas forem feitos em dupla, a dupla deve ser identificada no envio do código.

**Data de Entrega:** agendada no Sigaa

**Entregar:** Código Fonte, Relatório(Conforme Modelo em PDF)

**Forma de Entrega:** pelo SIGAA, caso tenha algum problema enviar por e-mail(julianaoc@ufpi.edu.br).

**Entrevista Individual:** agendar horário com a Professora.

**Obs.: Todos os alunos devem entregar o código e relatório, mesmo que o código esteja igual a do outro membro da dupla.**