

SCC0223 – Estruturas de Dados I

## Trabalho 2 – especificação

### 1. Objetivo

Empregar os conhecimentos adquiridos sobre estruturas de dados (TAD, listas lineares e não lineares e algoritmos relacionados) no desenvolvimento de uma aplicação computacional, seguindo um conjunto de especificações funcionais e de projeto, assim como boas práticas de programação.

### 2. Indicações Gerais

O trabalho deve ser desenvolvido em grupos de até três alunos, no máximo, sendo que:

- Serão aceitos somente trabalhos em linguagem de programação C.
- Data de entrega: 1/dezembro (domingo), até meia noite.
- Os trabalhos deverão ser apresentados para o estagiário PAE. Os grupos devem agendar a apresentação com o estagiário assim que fizerem a submissão pelo Tidia (ver item 4 abaixo).

### 3. Critérios de Avaliação

Os trabalhos serão avaliados de acordo com os seguintes critérios:

- Usabilidade da interface: a interface com o usuário deve ser clara e intuitiva, mesmo que seja via linha de comando, exibindo mensagens claras para o usuário.
- Corretude do programa: o programa deve fazer o que foi especificado.
- Estruturas de dados utilizadas: adequação e eficiência.
- Observação dos “bons modos” da programação: TAD, modularidade do código, documentação interna, indentação, etc.
- A cada dia de atraso, 1 ponto é descontado da nota. Lembrem-se de que (i) a média final dos trabalhos deve ser maior ou igual a 5 para que o aluno seja aprovado e (ii) os trabalhos têm peso 3 na média final.
- O plágio de programas não será tolerado. Quaisquer programas similares terão nota zero, independentemente de qual for o original e qual for a cópia.

### 4. Entrega do Trabalho

A entrega dos trabalhos será na atividade aberta no Tidia. Serão requeridos:

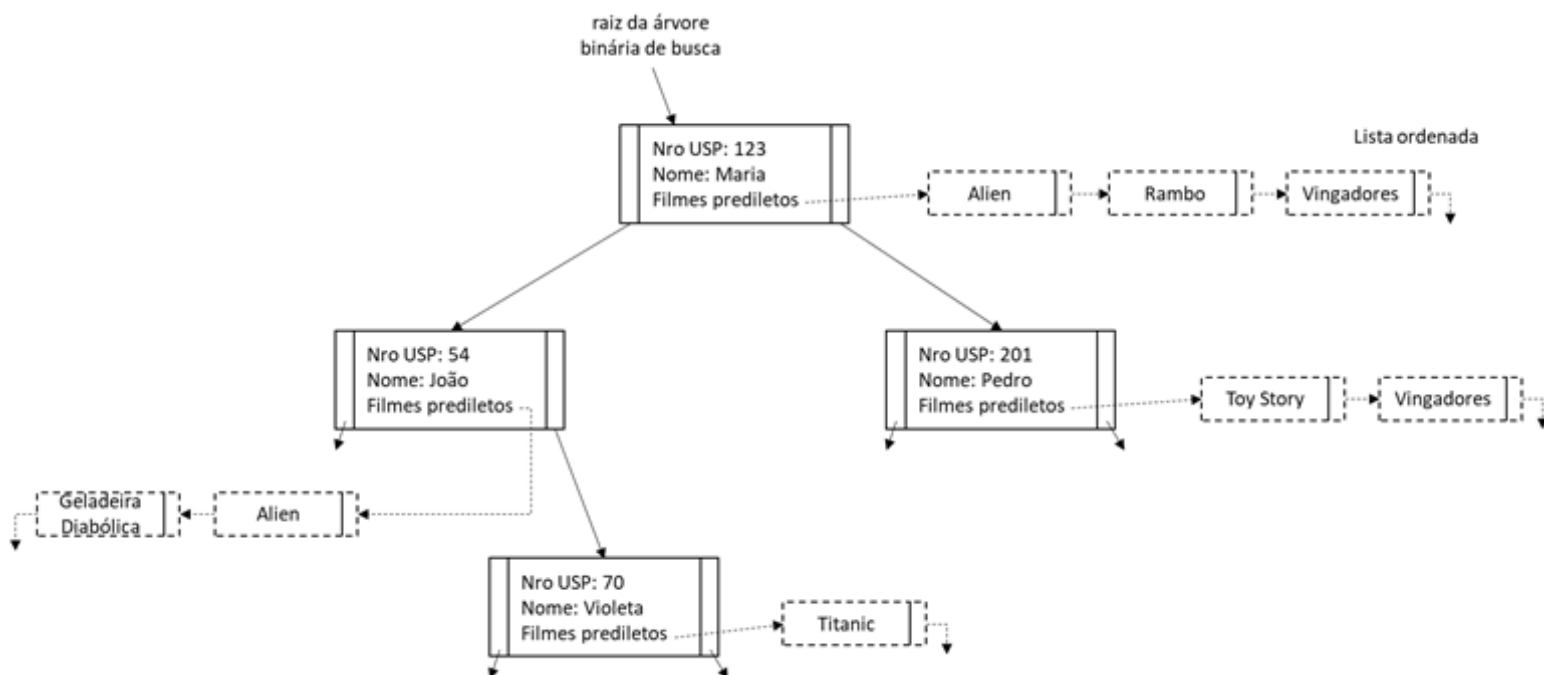
- Um arquivo comprimido (RAR, ZIP, etc.), que deverá ser postado por 1 dos membros do grupo no site da disciplina, e deve conter (a) arquivos de código-fonte do programa, (b) arquivo executável do programa e (c) relatório breve. O relatório, de aproximadamente 5 páginas, deve conter, pelo menos, (a) identificação dos integrantes do grupo, (b) breve descrição do trabalho feito, (c) apresentação das estruturas de dados utilizadas e justificativas necessárias e (d) uma seção detalhando como compilar e rodar o programa, com telas de exemplo da execução do sistema. Deixe claro que compilador usou (que versão, se utilizou alguma biblioteca diferente, qual sistema operacional utilizou, etc.).

## 5. Especificação: *amigos de graduação indo ao cinema*

Nesta seção, são apresentadas as especificações funcionais do programa que será desenvolvido. Para questões que não estejam especificadas, use o bom senso e documente suas decisões no relatório a ser entregue. Caso necessite, contate o estagiário PAE ou o professor da disciplina para tirar eventuais dúvidas.

Desenvolva um sistema de cadastro de alunos de uma turma de graduação e suas preferências pessoais sobre filmes, de tal forma que o sistema auxilie um aluno a lidar com questões relacionadas.

Mais detalhadamente, o sistema deve coletar os dados básicos dos alunos, como nome e número USP, além de perguntar e armazenar os nomes de seus filmes prediletos (em ordem alfabética, sendo que cada aluno pode indicar quantos filmes quiser – pelo menos 1 filme deve ser indicado), sobre os quais algumas tarefas serão executadas. Os alunos e seus dados devem ser armazenados em uma árvore binária de busca (sendo que o número USP deve ser a chave); deve haver uma lista ordenada de filmes relacionada a cada aluno. Não há limite prévio para o número de alunos ou de filmes cadastrados no sistema. A figura abaixo mostra um exemplo hipotético da estrutura de dados resultante.



No geral, um aluno pode realizar as seguintes operações no sistema:

- 1) criar um cadastro no sistema, informando os dados necessários;
- 2) listar todos os alunos cadastrados no sistema (atenção: você deverá utilizar algum dos algoritmos de percurso em árvore);
- 3) buscar um usuário específico no sistema, que responde se ele está ou não cadastrado;
- 4) listar em ordem alfabética todos os filmes diferentes cadastrados no sistema (a listagem exibida não deve ter repetições);
- 5) buscar um filme no sistema, que responde se ele foi ou não citado por algum usuário;
- 6) quando um usuário quiser ver um novo filme a ser lançado no cinema, o sistema deve indicar 1 colega de turma que possa gostar do filme para ir junto ao cinema (atenção: você deverá utilizar alguma métrica de “similaridade de preferências” para indicar o

- colega); por exemplo, na figura anterior, se Pedro quiser ir ao cinema, o sistema deve recomendar Maria para ir junto, pois ela tem 1 filme em comum com Pedro e há chances de que goste do filme que Pedro quer ver;
- 7) quando o usuário quiser aumentar sua experiência e ver um filme bem diferente do que usualmente assiste, o sistema deve indicar 1 colega de turma com o perfil mais diferente do seu, para que o usuário possa contata-lo pessoalmente depois e pedir uma sugestão de filme; por exemplo, na figura anterior, se Maria fizer essa consulta, o sistema deve indicar Violeta, que seria o usuário com menos filmes em comum com Maria;
  - 8) o usuário pode produzir um arquivo texto com todas as informações armazenadas no sistema;
  - 9) o usuário pode questionar o sistema sobre os dados técnicos da árvore binária de busca; nesse caso, o sistema deve exibir pelo menos (a) o número de nós existentes na árvore, (b) a altura total da árvore e (c) a maior diferença entre alturas que existe entre as sub-árvores de algum nó da árvore;
  - 10) o usuário pode resolver se retirar do cadastro mantido pelo sistema (nesse caso, a árvore deve ser reestruturada com a retirada do nó correspondente ao usuário);
  - 11) pelo menos uma funcionalidade extra; por exemplo, considere uma função para listar os filmes mais queridos pela turma toda ou uma função para fazer a leitura dos usuários a serem cadastrados e suas preferências a partir de um arquivo.

**Atenção:**

- Toda a implementação deve ser dinâmica e encadeada!
- Lembre-se de inicializar e finalizar apropriadamente as estruturas de dados utilizadas.