

## Trabalho Prático Grau B

### Estruturas Avançadas de Dados I

#### OBJETIVO

---

Nesse trabalho, deverá ser desenvolvida uma aplicação que utilize estrutura de árvore binária de busca balanceada AVL.

#### DESCRIÇÃO GERAL

---

O trabalho consiste em carregar um arquivo contendo informações de pessoas e possibilitar rápido acesso aos registros por meio de consultas por campos individuais.

#### PROBLEMA

---

Deverá ser desenvolvida uma aplicação que seja capaz de carregar um arquivo de entrada contendo as seguintes informações de pessoas (na seguinte ordem):

- número do CPF (numérico);
- número do RG (numérico);
- nome da pessoa (string);
- data de nascimento da pessoa (data no formato DD/MM/AAAA) e
- nome da cidade onde a pessoa nasceu (string).

O arquivo a ser lido estará em um formato CSV, de modo que os campos de cada registro sejam separados por ponto-e-vírgula (;). Seguem algumas linhas de um exemplo de como será o arquivo a ser lido:

12345678910;543216;Fulano de Tal;01/02/1958;Porto Alegre
10987654321;44556601;Cicrana Beltrana Delgrana;25/12/2972;São Leopoldo

Após carregar o arquivo, deverá ser possível realizar as seguintes consultas:

- Consultar uma única pessoa pelo seu CPF e exibir seus dados na tela;
- Consultar todas as pessoas cujo nome comece com uma string informada pelo usuário e exibir na tela todos os dados dessas pessoas na forma de lista.
- Consultar todas as pessoas cuja data de nascimento esteja em um intervalo estabelecido pelo usuário e exibir na tela todos os dados dessas pessoas na forma de lista.

#### PROCESSO DE DESENVOLVIMENTO

---

O programa deverá ser escrito seguindo as seguintes restrições:

- O trabalho pode ser desenvolvido em Java, C, C++, C# ou Python.
- Os registros lidos do arquivo deverão ser, obrigatoriamente, indexados pelos campos definidos na descrição do problema, de modo que todas as consultas possam ser executadas de forma bem rápida. Para isso, será necessária e OBRIGATÓRIA a criação dessa indexação em cada um dos

campos já mencionados de uma Pessoa.

- Utilize a estrutura de árvore binária de busca balanceada AVL, sendo que, para cada campo indexado (no caso, CPF, nome da pessoa e data de nascimento), crie uma árvore binária de busca AVL. Observe que não está sendo solicitada a implementação de três classes de árvores distintas. Nesse ponto, a criação se refere à instanciamento da árvore três vezes, sendo uma voltada para cada campo.
- Deverá ser criada uma interface minimamente navegável possibilitando a realização das consultas e visualização dos dados retornados. Não é necessário utilizar quaisquer recursos de interfaces ricas. Contudo, como incentivo, trabalhos utilizando alguma tecnologia de interface rica receberão 0,5 pontos adicionais ao trabalho, totalizando até 3,5.

**Observação:** Os registros (objetos Pessoa) não deverão estar duplicados na memória. A indexação é feita apenas com chaves e endereços dos registros, e não com cópias inteiras dos registros na memória.

### CRITÉRIOS DE AVALIAÇÃO

- Deve ser funcional! Programa que não executa não é programa. Deve ser compilável e executável. Aplicativos que não executarem serão desconsiderados e será atribuída nota zero para os aprendizes.
- Código estruturado, identado e comentado. Não esqueça de colocar seu nome no código;
- Implementação correta da árvore AVL.

### INSTRUÇÕES PARA ENTREGA

Data de entrega: 12/06/2020 até as 22h23min

Valor: 3,0 pontos

A entrega será realizada **SOMENTE via moodle**, até o final da aula do dia 12/06, ou seja, a tarefa irá fechar as 22h23min. Trabalhos não entregues no prazo, não serão aceitos. Pede-se que seja realizada a entrega de um **arquivo de vídeo** contendo a explicação do código implementado e o programa em execução, além de um **arquivo .zip com os códigos fontes** que compõem o trabalho.

O trabalho pode ser feito em equipes de até 3 (três) alunos.

#### MUITO IMPORTANTE!

Trabalhos copiados da Internet ou dos colegas terão nota zero.

Data de Entrega: 12/06/2020