

### Ficha de Trabalho n.º 7

#### Objectivos: Estruturas dinâmicas: **Árvores Binárias**.

Os exercícios propostos visam criar um programa que faça a gestão de um conjunto de pessoas utilizando uma árvore binária.

1. Defina uma estrutura **Pessoa**, associando-lhe o tipo de dados **\*ptPESSOA**, com os seguintes dados:
  - a) Nome (20 caracteres);
  - b) Idade em anos (inteiro);
  - c) Peso em quilos (número real);
  - d) Altura em metros (número real);
2. Defina uma estrutura **No**, associando-lhe o tipo de dados **\*ptNO**, adequada para a construção de uma árvore binária de elementos e com um campo informação do tipo **ptPESSOA**.
3. Defina uma estrutura **Arvore**, associando-lhe o tipo de dados **\*ptARVORE**, com os dados:
  - a) Número de nós (inteiro);
  - b) Nó raiz da árvore (ptNO).
4. Escreva uma função **ptARVORE criar\_arvore()** que crie (aloque e inicialize) uma árvore binária.
5. Escreva uma função **ptNO criar\_no()** que crie (aloque e inicialize) um novo nó.
6. Escreva uma função **void ler\_no(ptNO N)** que registe os dados de um dado nó.
7. Escreva uma função **int comparar\_nos(ptNO A, ptNO B)** que compare dois nós, utilizando como único critério de comparação a idade das pessoas associadas.
8. Escreva uma função **int nos\_iguais(ptNO A, ptNO B)** que verifique a igualdade entre dois nós, utilizando como único critério de comparação a idade das pessoas associadas.
9. Escreva uma função **void inserir\_no(ptNO no\_novo, ptARVORE A)** que introduza um nó na árvore, sendo a sua ordenação dada pelo campo **idade** da pessoa.
10. Escreva uma função **ptNO pesquisar(int idade, ptARVORE A)** que verifique se existe alguma pessoa com uma determinada idade na árvore.
11. Escreva uma função **void mostrar\_no(ptNO no\_mostrar)** que mostre os dados de um determinado nó.
12. Escreva funções que permitam realizar a travessia da árvore binária, mostrando a informação de todos os nós, segundo:
  - a) a técnica *inorder* - **void mostrarInOrder(ptARVORE A)**
  - b) a técnica *preorder* - **void mostrarPreOrder(ptARVORE A)**
  - c) a técnica *postorder* - **void mostrarPostOrder(ptARVORE A)**

13. Escreva uma função **void libertar\_no(ptNO no\_libertar)** que liberte o espaço alocado para um dado nó e respectivas informações associadas.
14. Escreva uma função **ptNO remover(ptNO no\_remove, ptNO \*praiz)** que remova um dado nó de uma árvore, utilizando o campo idade como chave.
15. Escreva uma função **int contar\_nos(ptARVORE A)** que calcule a quantidade de nós existentes numa dada árvore.
16. Escreva uma função **int contar\_nos\_sup\_idade(ptARVORE A, int idade\_min)** que determine o número de pessoas com uma idade superior a uma determinada idade mínima.
17. Escreva uma função **int contar\_nos\_inf\_peso(ptARVORE A, float peso\_max)** que determine o número de pessoas com um peso inferior a um determinado peso máximo.
18. Escreva uma função **int altura\_arvore(ptARVORE A)** que determine a altura de uma dada árvore.
19. Escreva uma função **int menuPrincipal()** que permita ao utilizador escolher entre as diversas funcionalidades do programa.

```
(1)  Inserir um novo NÓ na Árvore
(2)  Pesquisar NÓS da Árvore (por uma idade)
      Mostrar todos os NÓS da Árvore usando a técnica:
      (3)  inorder
      (4)  preorder
      (5)  postorder
(6)  Retirar um NÓ da Árvore (através da idade)
(7)  Mostrar o número total de NÓS da Árvore
(8)  Mostrar o número de NÓS na Árvore com uma idade mínima
(9)  Mostrar o número de NÓS na Árvore com um peso máximo
(10) Mostrar a altura da Árvore
(0)  SAIR
```

20. Escreva a função principal **void main()** de modo a integrar as funções elaboradas anteriormente.