

# Ficha 10

## Programação Imperativa

### Ficheiros

Relembre o problema proposto na Ficha 7 sobre a gestão da informação sobre alunos.

Uma forma de representar a informação de cada aluno definiu-se o tipo `Aluno`.

Considere agora que os vários alunos são organizados numa árvore binária de procura, ordenada pelo número do aluno (o tipo `Turma`).

```
typedef struct aluno {
    int numero;
    char *nome;
    int nota;
} Aluno;

typedef struct arv {
    Aluno a;
    struct arv *esq, *dir;
} *Turma;
```

1. Defina as seguintes funções sobre este tipo de dados.

- `int acrescentaAluno (Turma *t, Aluno a)` que acrescenta a informação de um dado aluno a uma turma. A função deverá retornar 0 se a operação for feita com sucesso. Nos casos em que a operação não puder ser feita (por exemplo, porque já existe um aluno com esse número na turma) deverá retornar um valor diferente de zero.
- `Aluno *procura (Turma t, int numero)` que procura o aluno com um dado número na turma. A função deve retornar `NULL` se a informação desse aluno não existir; caso exista deve retornar o endereço onde essa informação se encontra.
- `int aprovados (Turma t, int numero)` que determina quantos alunos obtiveram aproveitamento à disciplina (nota final maior ou igual a 10).
- `void printTurma (Turma t)` que imprime (no ecrã) a informação de todos os alunos (um por linha).

2. Considere agora que se pretende disponibilizar funções de leitura e escrita de um elemento deste tipo (`Turma`) em ficheiro.

Apresente duas alternativas para implementar tais funções:

- A escrita da árvore deve permitir que ao ler essa estrutura do ficheiro, a árvore obtida tenha a mesma forma da árvore que foi escrita no ficheiro.
- A escrita da árvore deverá permitir que ao ler a árvore se obtenha uma árvore equilibrada.

3. Uma alternativa ao exposto atrás, consiste em ter a informação sobre a turma, implementada numa árvore binária de procura directamente num ficheiro. Note que neste caso, os endereços das sub-árvores não serão mais do que endereços de disco.

Re-implemente as funções referidas na primeira alínea para esta solução alternativa.