

MATEMÁTICA / INFORMÁTICA
ENSINO da MATEMÁTICA
ENSINO da INFORMÁTICA

PROGRAMAÇÃO II

Folha prática 13

→ Ficheiros de organização indexada

Suponha um ficheiro de alunos definido da seguinte forma:

```

...
TYPE      aluno=record
           numero: [KEY(0)] integer;
           nome: packed array [1..30] of char;
           curso: [KEY(1)] packed array [1..20] of char;
           end;
VAR
  alunos: file of aluno;
  A:aluno;
begin
  OPEN ( FILE_VARIABLE:= alunos, ACCESS_METHOD:=KEYED,
        ORGANIZATION:=INDEXED, HISTORY:=UNKNOWN) ;
...

```

- Pretende-se um programa que permita manipular o ficheiro, devendo ser implementadas as seguintes funções: inserir registos, consultar alunos por número e por curso, apagar registos e construir listagens para o ecrã e para um ficheiro de texto.

- O programa deverá começar por mostrar ao utilizador as várias opções possíveis e depois consoante a escolha do utilizador, executar um procedimento que implemente a opção pretendida. Os procedimentos a implementar deverão ter a estrutura descrita nas alíneas abaixo:

a) Inserir registos

```

      ler número de aluno
      ENQUANTO (número <> 0 )
        ler (nome)
        ler (curso)
        posicionar acesso ao registo pela chave 0: FINDK(alunos, 0, A.numero,
EQL)
        SE ( o registo não existe ) ENTÃO
          escrever registo no ficheiro
        SENÃO
          Informar o utilizador de que o aluno já existe
        FIM SE
        ler novo número de aluno
      FIM ENQUANTO

```

b) Consultar alunos por número

```
ler número de aluno
ENQUANTO ( número <> 0 )
    posicionar acesso ao registo (pela chave 0)
    SE (o registo não existe)
        mensagem a informar o utilizador
    SENÃO
        ler o registo
        mostrar dados no ecrã
    FIM SE
    ler (novo número de aluno )
FIM ENQUANTO
```

c) Consultar alunos por curso

```
ler curso a consultar
ENQUANTO ( curso <> “ “ )
    posicionar acesso ao primeiro registo do curso(pela chave 1)
    SE (não existe nenhum aluno desse curso)
        mensagem a informar o utilizador
    SENÃO
        REPETIR
            ler registo
            SE (A.curso = curso a consultar) ENTÃO
                mostrar dados no ecrã
            FIM SE
        ATÉ QUE ( (A.curso > curso a consultar) or eof(alunos) )
    FIM SE
    ler (novo curso )
FIM ENQUANTO
```

d) Apagar registos

```
ler (número do aluno a apagar )
ENQUANTO (número <> 0 )
    posicionar acesso ao registo (pela chave 0)
    SE (o registo não existe)
        mensagem a informar o utilizador
    SENÃO
        ler o registo
        mostrar dados no ecrã
        Pedir ao utilizador que confirme se quer apagar (s/n)
        SE ( resposta = 's') ENTÃO
            posicionar o acesso ao registo
            apagar registo
        FIM SE
    FIM SE
    ler (novo número )
FIM ENQUANTO
```

e) Listar por número para um ficheiro de texto

```
abrir ficheiro de texto que vai conter a listagem
preparar ficheiro alunos para leitura sequencial pela chave 0: RESETK(alunos,0)
ENQUANTO ( não fim de ficheiro )
    ler registo do ficheiro
    escrever os dados do registo no ficheiro de texto
FIM ENQUANTO
fechar ficheiro de texto
```

f) Listar por cursos para um ficheiro de texto

```
abrir ficheiro de texto que vai conter a listagem
preparar ficheiro alunos para leitura sequencial pela chave 1: RESETK(alunos,1)
ENQUANTO ( não fim de ficheiro )
    ler registo do ficheiro
    escrever os dados do registo no ficheiro de texto
FIM ENQUANTO
fechar ficheiro de texto
```

g) Construa um procedimento para alterar o nome de um aluno.