

Número _____ Nome _____

Uma **Editora** dedica-se à publicação de vários jornais e revistas com diferentes periodicidades. Todas as publicações têm uma data de emissão, um preço e o número de exemplares emitidos. Considere que os jornais e as revistas se distinguem pelo facto de os primeiros poderem ou não apresentar suplemento e as revistas terem um tema associado e os exemplares não vendidos poderem ser devolvidos.

- a) Defina as classes indicando apenas os atributos e a assinatura dos métodos necessários à modelação de uma aplicação que permita gerir as publicações desta Editora.
- b) Escreva uma função que calcule o total de vendas da Editora num determinado ano, tendo em atenção os exemplares de revistas devolvidos.

Número/Nome: _____

Considere a classe **Exame** que guarda o número de testes realizados por cada aluno no exame de ESINF.

```
class Exame
{
    private:
        string id_alun;
        int num_tsts;

    public:
        Exame (string ia, int nt);
        Exame (const Exame &ex);
        ~Exame() ;

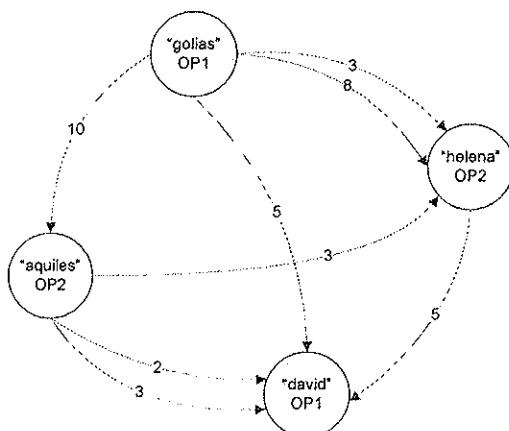
        string getIdAl() const ;
        int getNumTst() const ;
        void setIdAl (string ia) ;
        void setNumTst (int nt) ;

        bool operator == (const Exame& ex) const ;
        Exame& operator = (const Exame& ex) ;
        void escrever(ostream& out) const ;
};
```

1. Defina a classe **Arquivo**, com base na classe template `Lista<T>`, para guardar os exames de modo que no início da lista são colocados todos os alunos que fizeram um só teste, em seguida os alunos que fizeram dois testes e por último os alunos que fizeram três testes. Estando ainda cada grupo ordenado por ordem crescente de identificação do aluno. Indique apenas os atributos e a assinatura dos métodos necessários ao bom funcionamento da classe.
2. Escreva o método `insere` para inserir um exame na classe **Arquivo**.
3. Faça a sobrecarga do operador `==` que verifica se dois arquivos são iguais.

Número _____ Nome _____

Uma empresa de telecomunicações pretende fazer um estudo dos tarifários mais adequados aos seus assinantes, para tal precisa de representar em suporte informático a duração (em segundos) das chamadas telefónicas estabelecidas entre os seus assinantes - pessoas (definidas por nome e operador), conforme esquema da figura seguinte.



Considere a classe **Tarifario** definida usando as classes **ListaAdjGrafo<TV,TR>** e **Pessoa**

```
class Tarifario : public ListaAdjGrafo<Pessoa,int>
{
    public:
        Tarifario () ;
        Tarifario (const Tarifario &tf) ;
        ~Tarifario () ;
        ...
} ;
```



1. Adicione à classe **Tarifario** uma função que calcule a duração média das chamadas estabelecidas entre os assinantes de dois operadores dados. Admita que a classe **Pessoa** disponibiliza o método **string GetOperador() const** ;

2. Considere os métodos abaixo.

```
void exemplo (int a[], int n){
    int x = n/2;

    if (x > 0){
        exemplo(a,x);
        mostrarvect(a,0,x) ;
        mostrarvect(a,x,n-x); }
}
```

```
void mostrarvect (int v[], int pos, int n){
    for (int i=pos; i<n; i++){
        cout << v[i] << " " ;
    }
    cout << endl ;
}
```

Determine a complexidade temporal $T(n)$ do método **exemplo**. Justifique.