

## Aula 5

**Objectivo:** Construção de classes para estruturas de dados comuns

### Problema 1

Tendo em consideração os problemas já resolvidos, construa uma classe abstracta (sem uma implementação interna que retire generalidade ao módulo) que permita a utilização e/ou especialização de listas de elementos.

**1.1.** Comece por criar uma classe também abstracta: `ELEMENTO_DE_LISTA`. Esta classe para além dos métodos que lhe associam e devolvem um elemento qualquer (implementado com `void*`), deve também ter uma função booleana (`int` em C) com o nome `es_tu` e como parâmetro um qualquer critério de procura, também `void*`.

**1.2.** Construa a classe `LISTA`, com os seguintes métodos (defina a assinatura mais apropriada e simples):

```
adiciona      // adiciona elemento (ELEMENTO_DE_LISTA) à lista
apaga        // apaga o elemento actual da lista
limpa        // faz com que a lista fique vazia
primeiro     // faz com que o elemento actual seja o primeiro
seguinte     // faz com que o elemento actual seja o seguinte
actual       // devolve o elemento (ELEMENTO_DE_LISTA) actual da lista
e_fora       // diz se não há elemento actual
e_primeiro   // diz se o elemento actual é o primeiro
e_ultimo     // diz se o elemento actual é o último
procura      // percorre a lista a partir da posição actual
              // questionando 'es_tu' a cada elemento
              // se bem sucedida o elemento actual da lista
              // passará a ser o elemento procurado, caso
              // contrário ficará fora da lista
```

**1.3.** Implemente uma lista recorrendo a um array (`LISTA_COM_ARRAY`).

**1.4.** Refaça (e experimente) a lista de figuras dos problemas das aulas anteriores tirando proveito destas novas classes.

**1.5.** Generalize a `LISTA` por forma a permitir que se tenha múltiplos critérios de procura sobre a mesma. (Sugestão: faça com que a lista fique independente do critério [procedimento] de procura.)