

## Listas Múltiplas – ML

A Multi-Lista pode ser implementada considerando-se a existência dos TDA-Listas, essa abordagem é vantajosa, pois implica no reaproveitamento de todo o código já estudado para estes TDA's.

Aqui será abordado um conceito de TDA-ML conforme a Figura 1, ou seja, como um conjunto de LDDE's (ou LDSE's) agregadas por um vetor (o vetor seria uma “espinha dorsal” agregando as listas dinâmicas). Nesse caso, a ML passa a constituir uma “camada” sobre as LDDE's conforme é exibido na Figura 2. Na mesma figura, observe que o esquema de arquivos, até agora em uso, recebe o acréscimo de um TDA\_ML.H e TDA\_LDDE.C devidamente *incluídos* para compilação. Abaixo, também seguem alguns fragmentos de código, porém, implementação completa deste TDA ficará como exercício.

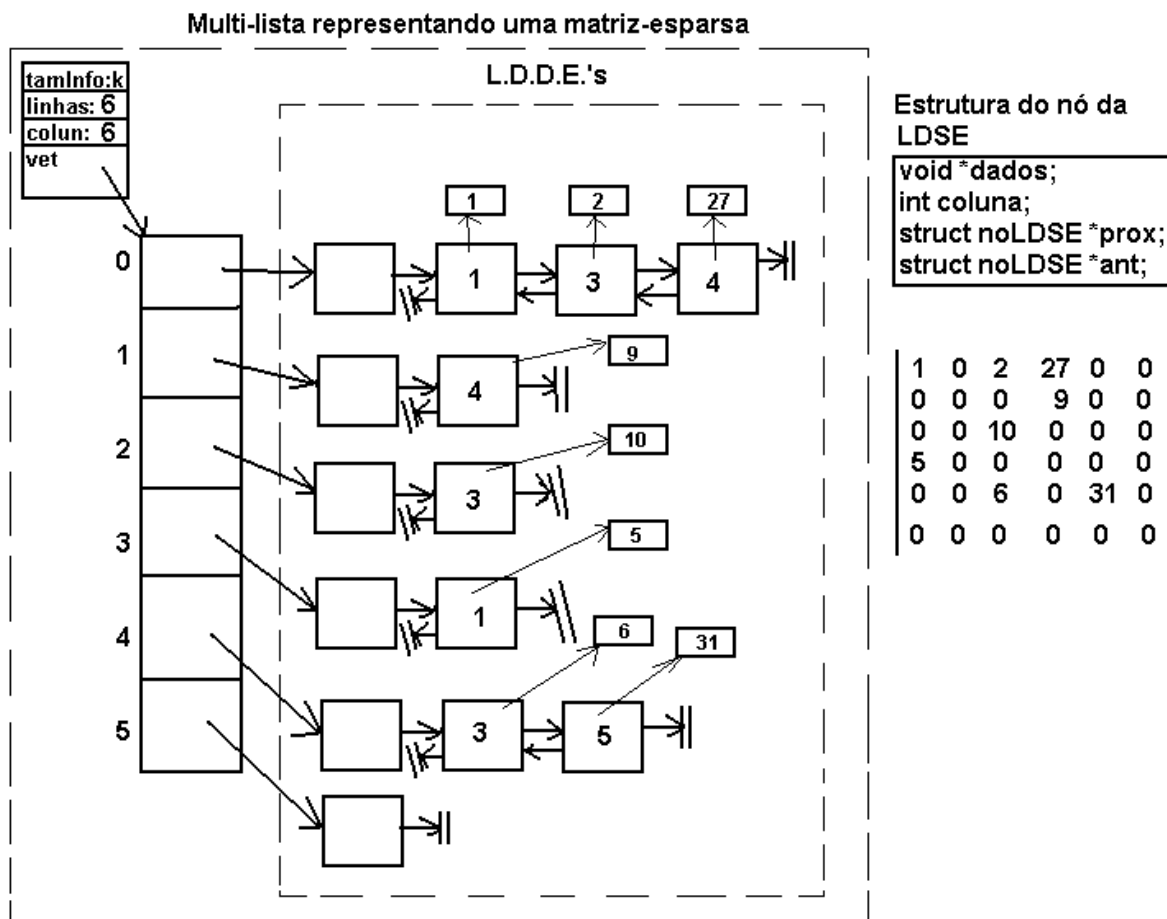


Figura 1 - Multi-Lista aplicada à representação de uma matriz esparsa.

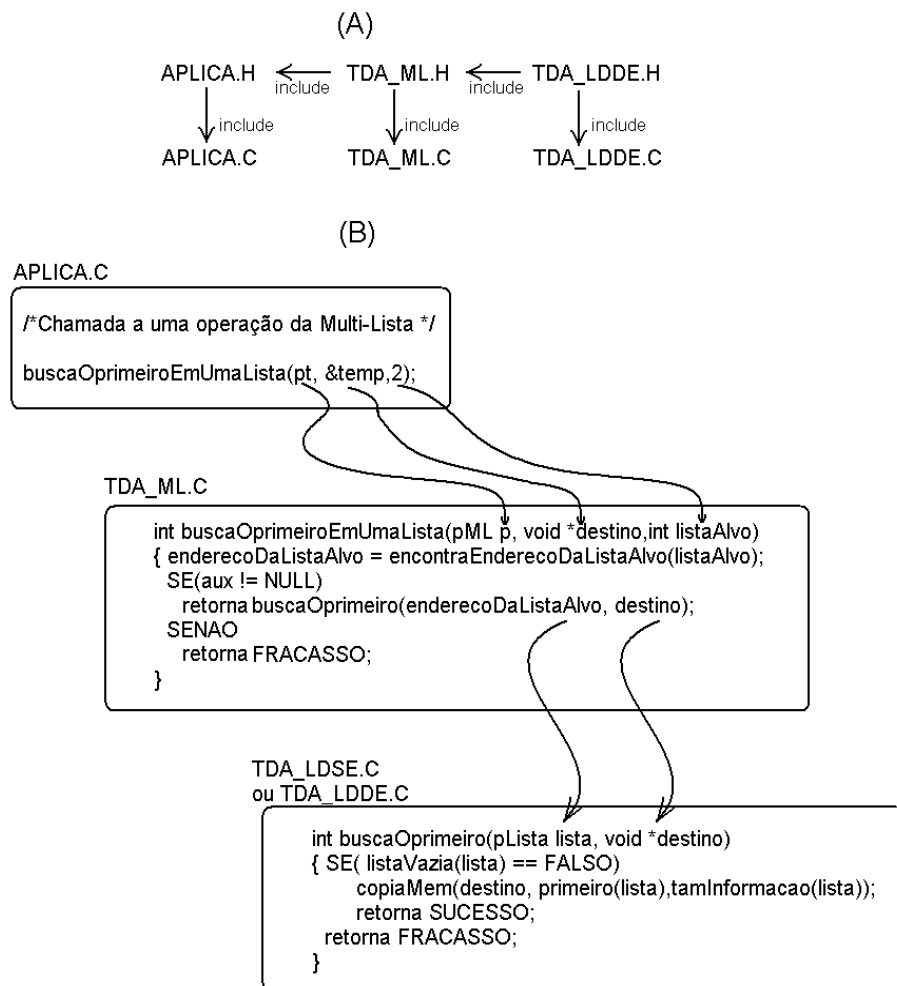
```
Fragmento do arquivo TDA_ML.H
#include "TDA_LDDE.H"
...
typedef struct ML *pML, *ppML;
...
```

```

Fragmento do arquivo TDA_ML_PRIV.H
#include "TDA_ML.H"
/* Descritor da Multi-Lista*/
typedef struct ML{
    int NumListas;
    ppLDDE vet;
}ML;

```

Outra estratégia de implementação da ML poderia ser a utilização de uma abordagem totalmente encadeada, onde uma lista dinâmica (ao invés de um vetor) serviria de “espinha dorsal” para o conjunto de listas dinâmicas agregadas, essa abordagem é interessante para representar uma Multi-Fila ou uma Multi-Pilha ou até mesmo um Grafo através do conceito de matriz de adjacências (Figura 3).



**Figura 2 – (A)Arquitetura para viabilizar (B) a ML como uma camada sobre a LDDE.**

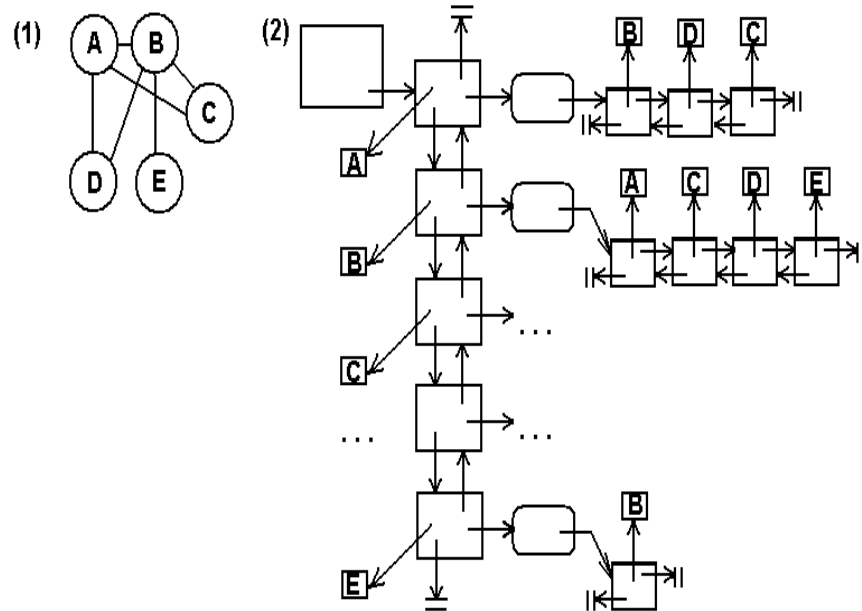


Figura 3 - Um Grafo e sua representação através da ML.