

## Programação 2024/25

### LEI, LEI-PL, LEI-CE

#### Desafio 4 – Estrutura Dinâmica

Considere as seguintes definições:

```
typedef struct tipoA cliente, *pCliente;
typedef struct tipoB acesso, *pAcesso;
typedef struct {int h, m;} hora;

struct tipoA{
    int id;                // Identificador único
    int contador;          // Número de utilizações nesse dia
    pAcesso lista;         // Ponteiro para a lista de acessos
    pCliente prox;         // Ponteiro para o próximo cliente
};

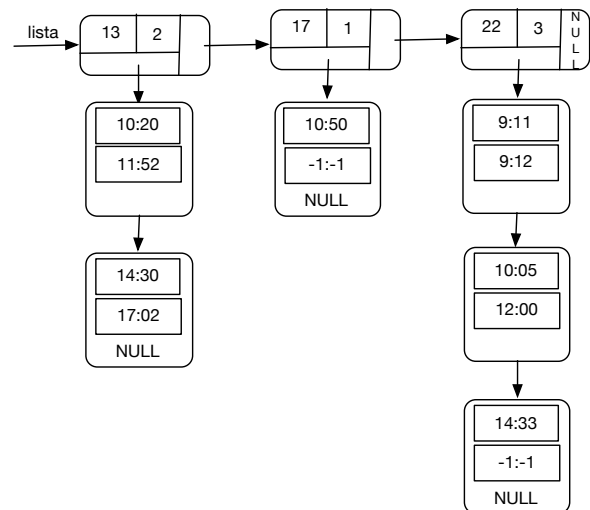
struct tipoB{
    hora in, out;          // Horas de entrada e saída
    pAcesso prox;
};
```

Um parque de estacionamento armazena numa estrutura dinâmica informação sobre a utilização dos seus serviços por parte dos clientes registados.

**Esta informação diz respeito a um único dia.**

Existe uma lista ligada principal com estruturas do tipo *cliente* contendo informação sobre **os clientes registados que já usaram o parque nesse dia**.

Todos os clientes que se encontram na lista ligada principal têm, no mínimo, um acesso ao parque nesse dia. A lista ligada principal está ordenada pelo valor do campo *id*. De cada nó da lista principal sai uma lista secundária com elementos do tipo *acesso* contendo informação sobre os acessos desse cliente ao longo do dia. Cada nó de acesso regista a hora em que se processou a entrada e a posterior saída do parque. As listas secundárias estão ordenadas por hora de utilização, ou seja, novas utilizações estão sempre no final desta lista.



Caso o cliente ainda se encontre no parque, a hora de saída do último nó tem o valor -1:-1. Estas são consideradas utilizações em aberto.