

## Trabalho Final

### 1. Introdução

O objetivo deste trabalho é praticar a programação utilizando a linguagem C através da implementação de um sistema de cadastro para uma loja de vinhos. O sistema deve ser implementado utilizando exclusivamente a linguagem C e exercitando os conceitos estudados em aula, com ênfase em ponteiros e alocação dinâmica da memória.

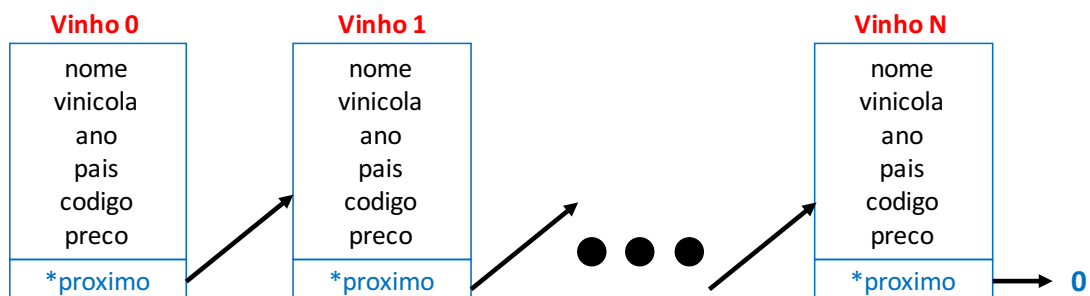
### 2. Funcionamento e detalhes técnicos

O sistema deve permitir o operador realizar as seguintes funcionalidades:

- Cadastrar novos vinhos. No cadastro são armazenadas as seguintes informações:
  - ✓ Nome do vinho;
  - ✓ Nome da vinícola;
  - ✓ Ano do vinho;
  - ✓ País de origem;
  - ✓ Código identificador do vinho na loja;
  - ✓ Preço anunciado.
- Remover vinhos cadastrados utilizando o código identificador do vinho na loja;
- Buscar um vinho específico utilizando o código identificador do vinho na loja;
- Ordenar os livros da loja no sistema por nome do vinho e da vinícola, ano do vinho, país de origem, código identificador ou preço;
- Imprimir todos os vinhos cadastrados na tela.

Para representar o cadastro de vinhos, utilizar uma lista encadeada. Para facilitar o trabalho, **o aluno pode escolher** entre implementar o programa usando uma lista simplesmente encadeada, **ou** duplamente encadeada, conforme especificado nas figuras abaixo:

#### 2.1. Lista simplesmente encadeada (SE)

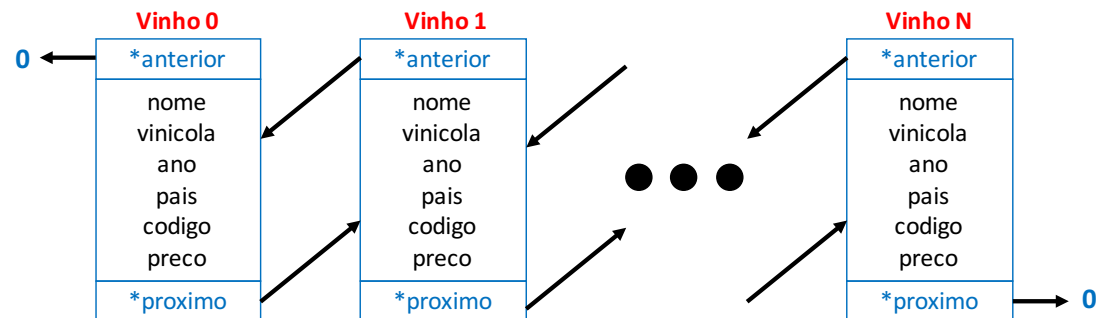


Para tanto, a seguinte *struct* deverá ser utilizada para representar cada vinho:

```
struct VINHO {
    char        nome[30];
    char        vinicola[30];
    int         ano;
    char        pais[20];
    int         codigo;
    float       preco;
    struct VINHO *proximo;
};
```

Note que o membro “*proximo*” é o único ponteiro para uma “*struct VINHO*”, o qual irá permitir a ligação entre os nós da lista.

## 2.2. Lista duplamente encadeada (DE)



Para tanto, a seguinte *struct* deverá ser utilizada para representar cada vinho:

```
struct VINHO {
    char        nome[30];
    char        vinicola[30];
    int         ano;
    char        pais[20];
    int         codigo;
    float       preco;
    struct VINHO *anterior;
    struct VINHO *proximo;
};
```

Note que os membros “*anterior*” e “*proximo*” são dois ponteiros para a “*struct VINHO*”, que permitem a ligação entre os nós da lista.

## 2.3. Ambas as listas encadeadas

A referência para o início da lista deve ser dada por uma variável global chamada “*lista*”:

```
struct VINHO *lista;
```

Essa variável deve ser inicializada na função `main` com o valor `0` na inicialização do sistema e, conforme o andamento do sistema, deverá apontar sempre para o primeiro vinho da lista. Para a implementação das funcionalidades do sistema, utilizar as seguintes funções:

```
void cadastrar_vinho(char nome[30], char vinicola[30], int ano,
char pais[20], int codigo, float preco);
void remover_vinho(int codigo);
void buscar_vinho(int codigo);
void ordenar_vinho(int opcao);
void imprimir_vinho();
```

Guia para implementação dessas funções:

- **cadastrar\_vinho**: Essa função recebe um vinho (nome, vinícola, ano, país, código e preço) e inclui um novo elemento na lista de vinhos cadastrados. Para a inclusão de vinhos, navegar pela lista utilizando o ponteiro `proximo` de cada nodo. O último vinho sempre terá o valor `0` escrito no ponteiro `próximo`. Dessa forma, pode-se identificar o final da lista buscando por um `0` no ponteiro `próximo` durante a navegação. Se o aluno escolher implementar uma lista duplamente encadeada, o primeiro vinho sempre terá o valor `0` escrito no ponteiro `anterior`.
- **remover\_vinho**: Recebe um código e busca pelo vinho correspondente. Quando o vinho é encontrado, essa função deve removê-lo da lista. Caso o vinho não seja encontrado, a função deverá enviar uma resposta de erro ao usuário.
- **buscar\_vinho**: Recebe um código e busca pelo vinho correspondente. Quando o vinho é encontrado, essa função deve imprimir as informações desse vinho na tela. Caso o vinho não seja encontrado, a função deverá enviar uma resposta de erro ao usuário.
- **ordenar\_vinho**: Essa função ordena os vinhos armazenados no sistema de acordo com a opção do usuário:
  - ✓ Se a opção for igual a 0, os vinhos deverão ser ordenados pelo ano dos vinhos.
  - ✓ Se a opção for igual a 1, os vinhos deverão ser ordenados pelo preço.
  - ✓ Se a opção for igual a 2, os vinhos deverão ser ordenados pelo país de origem.
  - ✓ Se a opção for igual a 3, os vinhos deverão ser ordenados pelo código identificador.
  - ✓ Se a opção for igual a 4, os vinhos deverão ser ordenados pelo seu nome e vinícola.
- **imprimir\_vinho**: Essa função deverá imprimir a lista completa de vinhos.

A interface para o usuário fazer uso dessas funções deverá ser projetada na função `main`. Para tanto, o operador deverá escolher entre as seguintes opções durante a execução do sistema:

- ✓ **c** – Cadastrar um novo vinho;
- ✓ **r** – Remover um vinho;
- ✓ **b** – Buscar um vinho;
- ✓ **o** – Ordenar a lista de vinhos;
- ✓ **i** – Imprimir a lista de vinhos;
- ✓ **s** – Sair do programa.

### 3. Avaliação

Leia com atenção os critérios de avaliação:

- Implementação adequada da lógica.
- Leitura apropriada das entradas do usuário e apresentação das informações na forma de texto no terminal.
- O trabalho será em **duplas** ou **individual**. Os arquivos contendo o código-fonte (.c) devem ser compactados e submetidos pelo Moodle até a data e hora especificadas. O arquivo compactado deve ter os nomes e os últimos sobrenomes de ambos os alunos, da seguinte forma:
  - ✓ **nome1\_sobrenome1\_nome2\_sobrenome2.zip**
- O código-fonte deve estar indentado e comentado adequadamente.
- A nota do trabalho depende da apresentação deste no laboratório, na data marcada. Trabalhos entregues, mas não apresentados terão sua nota anulada automaticamente. Durante a apresentação será avaliado o domínio da resolução do problema, podendo inclusive ser possível invalidar o trabalho quando constatada a falta de conhecimento sobre o código implementado.
- **A cópia parcial ou completa do trabalho terá como consequência a atribuição de nota ZERO ao trabalho dos alunos envolvidos.**