

# Software de Gerenciamento de Estoque de Mercearia

Algoritmos e Estruturas de Dados 1

Team 1 - 05/02/2023





# Assuntos Abordados

- ▶ 1 - Problemática
- ▶ 2 - Proposta de Solução
- ▶ 3 - Desenvolvimento da Solução
- ▶ 4 - Resultados



# Membros

- ▶ **Giordana Bucci**- 202200513
- ▶ **Guilherme Tavares** - 202200516
- ▶ **Ana Luísa Chagas**- 202200491
- ▶ **Pedro Lobo** - 202200550



# Problemática

Guilherme - 202200516



# Mercearias e Empórios

Típico comércio mundial, existente principalmente em países emergentes com o intuito de fornecer produtos de uso cotidiano, geralmente se localizando em áreas residenciais. Muito comum no Brasil e geralmente contendo produtos de diversas categorias.





# Público Alvo

Geralmente são gerenciadas por famílias que não vêem necessidade ou facilidade no uso da tecnologia em seus comércios.





# Necessidades

Visto o público e o local de aplicação, é necessário a criação de um software que seja otimizado, simplificado e que gere resultados.





# Proposta de Solução

Ana Luísa - 202200491



# Software de Gerenciamento de Estoque

Um software em C que armazena todos os produtos em categorias, informando seu nome, preço e quantidade em estoque. Ordenando-os por preço.





Sendo também capaz de realizar uma compra selecionando produtos e calculando o valor final da operação.





# Desenvolvimento



```
● ● ●  
1 #include <stdio.h>  
2 #include <stdlib.h>  
3 #include <string.h>  
4 #include <math.h>  
5  
6 //declaracoes de estruturas usadas para montar o estoque  
7 typedef struct _no {  
8     double preco;  
9     int qtd, cat;  
10    char * nome;  
11    struct _no * prox;  
12    struct _no * ant;  
13 } No;  
14  
15 typedef struct _lista {  
16     No * inicio, * fim;  
17     int tamanho;  
18 } Lista;
```



```
1 // declaração de funções
2 void menu(void);
3 Lista * criar_lista_vazia(void);
4 No * criar_no(double p, int q, int c, char * nome);
5 void add_no_final(Lista * lista, No * no);
6 void remover_elemento(Lista * lista, char * nome);
7 No * buscar(Lista * lista);
8 void ordenacao_preco(Lista * lista);
9 void ordena_preco_cres(Lista * lista);
10 void ordena_preco_dec(Lista * lista);
11 void add_produto(Lista * lista_geral, Lista * vetor[]);
12 void imprimir (Lista * lista);
13 void remover_produto(Lista * lista_geral, Lista * vetor[]);
14 void valor_da_compra(Lista * lista);
```





```
● ● ●
1 #include "estoque.h"
2
3 int main(void) {
4
5     char ans;
6     No * busca = NULL;
7     Lista * lista_geral = criar_lista_vazia();
8
9     Lista * carnes = criar_lista_vazia();
10    Lista * limpeza = criar_lista_vazia();
11    Lista * higiene = criar_lista_vazia();
12    Lista * papelaria = criar_lista_vazia();
13    Lista * padaria = criar_lista_vazia();
14    Lista * hortifrutti = criar_lista_vazia();
15    Lista * vetor[6] = {carnes, limpeza, higiene, papelaria, padaria, hortifrutti};
16    //          0      1      2      3      4      5
17
18    do {
19        menu(); //exibicao do rol de opcoes a serem selecionadas
20        scanf("%c%c", &ans);
21
22        switch (ans) {
23            case '1':
24                printf("Voce escolheu adicionar um produto!\n");
25                add_produto(lista_geral, vetor);
26                break;
27
28            case '2':
29                printf("Voce escolheu remover um produto!\n");
30                remover_produto(lista_geral, vetor);
31                break;
32
33            case '3':
34                printf("Voce escolheu visualizar nosso estoque!\n");
35                imprimir(lista_geral);
36                break;
37        }
38    } while (ans != '0');
39
40    return 0;
41}
```



```
37
38     case '4':
39         printf("Voce escolheu buscar por um produto!\n");
40         busca = buscar(lista_geral);
41         if(busca==NULL) printf("Nome: %s, Preco: %.2lf, Quantidade: %d, Categoria: %d",
42             busca->nome, busca->preco, busca->qtd, busca->cat);
43         else printf("O produto nao foi encontrado!\nAdicione-o pelo comando 1.\n");
44         break;
45
46     case '5':
47         if(lista_geral->tamanho==0){
48             printf("\nErro! Nao ha elementos no estoque para serem ordenados.\n");
49             break;
50         }
51         printf("Voce escolheu ordenar os produtos com base no preco!\n");
52         ordenacao_preco(lista_geral);
53         break;
54
55     case '6':
56         if(lista_geral->tamanho==0){
57             printf("\nErro! Nao ha elementos no estoque para realizar uma compra.\n");
58             break;
59         }
60         printf("Voce escolheu calcular o valor da compra!\n");
61         valor_da_compra(lista_geral);
62         break;
63
64     case 'S':
65         printf("Ate a proxima!\n");
66         break;
67
68     default:
69         printf("Opcao invalida!\n");
70     break;
71 }
72 } while (ans != 'S'); // 'S' e o comando de saida
73
74 return 0;
75 }
```



```
1 #include "estoque.h"
2
3 void menu(void) { // funcao que exibe um rol de opcoes para
o usuario
4     printf("\n===== MENU =====\n");
5     printf("[ 1] Adicionar produto\n");
6     printf("[ 2] Remover produto\n");
7     printf("[ 3] Exibir estoque\n");
8     printf("[ 4] Buscar produto\n");
9     printf("[ 5] Ordenar produtos em funcao do preco\n");
10    printf("[ 6] Calcular valor da compra\n");
11    printf("[ S] Sair\n");
12    printf("=====\\n");
13    printf("Digite uma opcao: ");
14 }
```



```
1  Lista * criar_lista_vazia(void) { // funcao que cria uma
   lista, vazia, que ira estocar produtos
2      Lista * lista = NULL;
3      lista = (Lista *) malloc(sizeof(Lista));
4
5      if (lista == NULL) return NULL;
6
7      lista->inicio = NULL;
8      lista->fim = NULL;
9      lista->tamanho = 0;
10     return lista;
11 }
```



```
1 No * criar_no(double p, int q, int c, char *nome){ //funcao que  
cria os nos relativos as listas de produtos  
2     No * no = NULL;  
3     no = (No *) malloc(sizeof(No));  
4  
5     if (no == NULL) return NULL;  
6  
7     no->preco = p;  
8     no->qtd = q;  
9     no->cat = c;  
10  
11    no->nome = NULL;  
12    no->nome = (char *) malloc((strlen(nome)+1)*sizeof(char));  
13    strcpy(no->nome, nome);  
14  
15    no->prox = NULL;  
16    no->ant = NULL;  
17  
18    return no;  
19  
20 }
```



```
1 void add_no_final(Lista * lista, No * no) { //funcao que adiciona,  
2   ao final da lista, um produto recem-adicionado  
3   if (lista == NULL) return;  
4  
5   if (lista->inicio == NULL) { // inicio da lista  
6     lista->inicio = no;  
7     lista->fim = no;  
8   } else {  
9     no->ant = lista->fim;  
10    lista->fim->prox = no;  
11    lista->fim = no;  
12  }  
13  
14  lista->tamanho++;  
15 }
```



```
1 void add_produto(Lista * lista_geral, Lista * vetor[]){ //cria o produto
    requerido, e ao final chama a funcao que adiciona o item ao final
2     int cat, qtd;
3     double preco;
4     char nome[100];
5     No * produto = NULL;
6
7     printf("[0] Carnes\n[1] Limpeza\n[2] Higiene\n[3] Papelaria\n[4] Pada
ria\n[5] Hortifruti\n");
8     printf("Escolha a o tipo de produto: ");
9     scanf("%d%c", &cat);
10
11    printf("Insira o nome do produto: ");
12    scanf("%[^%c]", nome);
13
14    printf("Insira o preco do produto: ");
15    scanf("%lf%c", &preco);
16
17    printf("Insira a quantidade disponivel em estoque: ");
18    scanf("%d%c", &qtd);
19
20    produto = criar_no(preco, qtd, cat, nome);
21    add_no_final(vetor[cat], produto);
22    add_no_final(lista_geral, produto);
23
24    printf("\nProduto adicionado com sucesso\n");
25 }
```



```
1 void remover_elemento(Lista * lista, char * nome) { //funcao que remove da
    lista o produto escolhido
2     No *remover = NULL;
3     No *aux;
4
5     if(lista->inicio != NULL){
6         if(strcmp(lista->inicio->nome, nome) == 0){
7             remover = lista->inicio;
8             lista->inicio = remover->prox;
9             if(lista->inicio != NULL)
10                 lista->inicio->ant = NULL;
11
12         } else {
13             aux = lista->inicio;
14             while(aux->prox != NULL && strcmp(aux->prox->nome, nome) != 0)
15                 aux = aux->prox;
16
17             if(aux->prox != NULL){
18                 remover = aux->prox;
19                 aux->prox = remover->prox;
20                 if(aux->prox != NULL)
21                     aux->prox->ant = aux;
22
23             }
24         }
25     }
26 }
```



```
1 void remover_produto(Lista * lista_geral, Lista * vetor[]) { //funcao que
    "chama" a funcao remover_elemento para lista geral e especifica
2     No * remover = NULL;
3     remover = buscar(lista_geral);
4
5     if(lista_geral->inicio == NULL) {
6         printf("Nosso estoque esta vazio!\n");
7         printf("Adicione itens pela opcao 1.\n");
8         return;
9     }
10
11    if(remover == NULL) {
12        printf("Nao temos este item em nosso estoque!");
13        return;
14    }
15
16    remover_elemento(lista_geral, remover->nome);
17    remover_elemento(vetor[remover->cat], remover->nome);
18    lista_geral->tamanho--;
19    vetor[remover->cat]->tamanho--;
20    free(remover);
21
22    printf("Produto removido com sucesso!\n");
23 }
```



```
1 No *buscar(Lista *Lista){ // funcao que realiza a busca por um produto de  
2     // inventario cadastrado  
3     No * no = NULL;  
4     No * aux;  
5     char nome[100];  
6     printf("Atencao: o sistema diferencia letras maiusculas e minusculas!  
7     \nEscreva atentamente.\n\n"); //para prevenir possiveis erros  
8     printf("Nome do produto: ");  
9     scanf("%[^\\n]*c", nome);  
10    aux = Lista->inicio;  
11    while(aux && strcmp(aux->nome, nome) != 0)  
12        aux = aux->prox;  
13    if(aux)  
14        no = aux;  
15    return(no);  
16 }
```



```
1 void ordenacao_preco(Lista * lista){
2     int op;
3
4     printf("\n\nEscolha:\n[1] para ordenar em ordem crescente\n[2] para o
5         ordenar em ordem decrescente\n\n");
6     scanf("%d%*c", &op);
7
8     switch (op){
9     case 1:
10        ordena_preco_cres(lista);
11        imprimir(lista);
12        break;
13
14    case 2:
15        ordena_preco_dec(lista);
16        imprimir(lista);
17        break;
18
19    default:
20        printf("\nErro! Opcao invalida.\n");
21        break;
22
23
24 }
```



```
1 void ordena_preco_dec(Lista * Lista){ //assim como a funcao anterior, o  
2   //ordena por preco, mas por decrescencia de preco  
3   No *i, *j;  
4   double tempPreco;  
5   int tempQtd, tempCat;  
6   char *tempNome;  
7  
8   for(i = Lista->inicio; i != NULL; i = i->prox) {  
9     for(j = i->prox; j != NULL; j = j->prox) {  
10       if(j->preco > i->preco) {  
11         tempPreco = i->preco;  
12         i->preco = j->preco;  
13         j->preco = tempPreco;  
14         tempQtd = i->qtd;  
15         i->qtd = j->qtd;  
16         j->qtd = tempQtd;  
17         tempCat = i->cat;  
18         i->cat = j->cat;  
19         j->cat = tempCat;  
20         tempNome = i->nome;  
21         i->nome = j->nome;  
22         j->nome = tempNome;  
23       }  
24     }  
25   }  
26 }
```



```
1 void ordena_preco_cres(Lista * lista) { //ordena, por selection sort, o
2     estoque em ordem crescente de preco
3     No *i, *j;
4     double tempPreco;
5     int tempQtd, tempCat;
6     char *tempNome;
7
8     for(i = lista->inicio; i != NULL; i = i->prox) {
9         for(j = i->prox; j != NULL; j = j->prox) {
10            if(j->preco < i->preco) {
11                tempPreco = i->preco;
12                i->preco = j->preco;
13                j->preco = tempPreco;
14                tempQtd = i->qtd;
15                i->qtd = j->qtd;
16                j->qtd = tempQtd;
17                tempCat = i->cat;
18                i->cat = j->cat;
19                j->cat = tempCat;
20                tempNome = i->nome;
21                i->nome = j->nome;
22                j->nome = tempNome;
23            }
24        }
25    }
```



```
1 void valor_da_compra(Lista * Lista){ //calcula o valor da compra do cliente
2     int qtd_geral, qtd_indv, i;
3     double valor = 0;
4     char nome[100];
5     No * produto = NULL;
6     No * aux;
7
8     printf("\nEscolha a quantidade de tipos de itens que o(a) senhor(a) dese
ja comprar: ");
9     scanf("%d*c", &qtd_geral);
10
11    for(i = 0; i < qtd_geral; i++){
12        printf("Insira o nome do produto: ");
13        scanf("%[^\\n]*c", nome);
14        printf("Insira a quantidade que deseja comprar: ");
15        scanf("%d*c", &qtd_indv);
16
17        aux = Lista->inicio;
18        while(aux && strcmp(aux->nome, nome) != 0) //buscando o produto esco
lhido
19            aux = aux->prox;
20
21        if(aux) //verifica se o produto existe
22            produto = aux;
23
24        if(produto != NULL){
25            if(0 < qtd_indv <= produto->qtd) //verifica se existe a quantida
de escolhida no estoque
26                valor += (produto->preco * qtd_indv);
27            else {
28                if(qtd_indv <= 0) printf("A quantidade deve ser maior que 0.
\n");
29                if(qtd_indv > produto->qtd) printf("Infelizmente nao temos e
ssa quantidade em estoque.\n");
30            }
31
32        } else printf("Produto fora de estoque.\n");
33    }
34
35    printf("Valor total da compra: R$%.2lf", valor);
36 }
```



```
1 void imprimir (Lista *Lista) { //exibe, na tela, a Lista de produtos
2     No * aux = Lista->inicio;
3     if(Lista->tamanho==0){
4         printf("O estoque esta vazio!\nAdicione produtos pela opcao
5             1.\n");
6         return;
7     }
8     printf("\n\nEstoque:\n");
9     while(aux != NULL) {
10         printf("Nome: %s, Preco: %.2lf, Quantidade: %d\n", aux->nome,
11               aux->preco, aux->qtd);
12         aux = aux->prox;
13     }
14     printf("Tamanho: %d\n", Lista->tamanho);
15 }
```



# Resultados



```
===== MENU =====
[ 1] Adicionar produto
[ 2] Remover produto
[ 3] Exibir estoque
[ 4] Buscar produto
[ 5] Ordenar produtos em funcao do preco
[ 6] Calcular valor da compra
[ S] Sair
=====
Digite uma opcao: █
```



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em funcao do preco
  - [ 6] Calcular valor da compra
  - [ S] Sair
- =====

Digite uma opcao: 1

Voce escolheu adicionar um produto!

- [0] Carnes
- [1] Limpeza
- [2] Higiene
- [3] Papelaria
- [4] Padaria
- [5] Hortifruti

Escolha o tipo de produto: 3

Insira o nome do produto: borracha

Insira o preco do produto: 2.99

Insira a quantidade disponivel em estoque: 100

Produto adicionado com sucesso



```
===== MENU =====
[ 1] Adicionar produto
[ 2] Remover produto
[ 3] Exibir estoque
[ 4] Buscar produto
[ 5] Ordenar produtos em funcao do preco
[ 6] Calcular valor da compra
[ S] Sair
=====
Digite uma opcao: 2
Voce escolheu remover um produto!
Atencao: o sistema diferencia letras maiusculas e minusculas!
Escreva atentamente.

Nome do produto: borracha
Produto removido com sucesso!
```



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em funcao do preco
  - [ 6] Calcular valor da compra
  - [ S] Sair
- =====

Digite uma opcao: 2

Voce escolheu remover um produto!

Atencao: o sistema diferencia letras maiusculas e minusculas!

Escreva atentamente.

Nome do produto: pao

Nao temos este item em nosso estoque!



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em funcao do preco
  - [ 6] Calcular valor da compra
  - [ S] Sair
- =====

Digite uma opcao: 3

Voce escolheu visualizar nosso estoque!

Estoque:

Nome: picanha, Preco: 59.90, Quantidade: 20  
Nome: detergente, Preco: 1.99, Quantidade: 100  
Nome: escova de dente, Preco: 15.00, Quantidade: 30  
Nome: borracha, Preco: 2.99, Quantidade: 40  
Nome: pao, Preco: 0.50, Quantidade: 23  
Nome: alface, Preco: 1.00, Quantidade: 15  
Tamanho: 6



```
===== MENU =====
[ 1] Adicionar produto
[ 2] Remover produto
[ 3] Exibir estoque
[ 4] Buscar produto
[ 5] Ordenar produtos em funcao do preco
[ 6] Calcular valor da compra
[ S] Sair
=====
```

```
Digite uma opcao: 3
Voce escolheu visualizar nosso estoque!
O estoque esta vazio!
Adicione produtos pela opcao 1.
```



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em funcao do preco
  - [ 6] Calcular valor da compra
  - [ S] Sair
- 

Digite uma opcao: 4

Voce escolheu buscar por um produto!

Atencao: o sistema diferencia letras maiusculas e minusculas!

Escreva atentamente.

Nome do produto: detergente

Nome: detergente, Preco: 1.99, Quantidade: 90, Categoria: 1



```
===== MENU =====
[ 1] Adicionar produto
[ 2] Remover produto
[ 3] Exibir estoque
[ 4] Buscar produto
[ 5] Ordenar produtos em funcao do preco
[ 6] Calcular valor da compra
[ S] Sair
=====
```

Digite uma opcao: 4

Voce escolheu buscar por um produto!

Atencao: o sistema diferencia letras maiusculas e minusculas!

Escreva atentamente.

Nome do produto: bolo

O produto nao foi encontrado!

Adicione-o pelo comando 1.



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em funcao do preco
  - [ 6] Calcular valor da compra
  - [ S] Sair
- =====

Digite uma opcao: 5

Voce escolheu ordenar os produtos com base no preco!

Escolha:

- [1] para ordenar em ordem crescente
- [2] para ordenar em ordem decrescente

1

Estoque:

Nome: pao, Preco: 0.50, Quantidade: 23  
Nome: alface, Preco: 1.00, Quantidade: 15  
Nome: detergente, Preco: 1.99, Quantidade: 100  
Nome: borracha, Preco: 2.99, Quantidade: 40  
Nome: escova de dente, Preco: 15.00, Quantidade: 30  
Nome: picanha, Preco: 59.90, Quantidade: 20  
Tamanho: 6



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em funcao do preco
  - [ 6] Calcular valor da compra
  - [ S] Sair
- =====

Digite uma opcao: 5

Voce escolheu ordenar os produtos com base no preco!

Escolha:

- [1] para ordenar em ordem crescente
- [2] para ordenar em ordem decrescente

2

Estoque:

Nome: picanha, Preco: 59.90, Quantidade: 20  
Nome: escova de dente, Preco: 15.00, Quantidade: 30  
Nome: borracha, Preco: 2.99, Quantidade: 40  
Nome: detergente, Preco: 1.99, Quantidade: 100  
Nome: alface, Preco: 1.00, Quantidade: 15  
Nome: pao, Preco: 0.50, Quantidade: 23  
Tamanho: 6



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em funcao do preco
  - [ 6] Calcular valor da compra
  - [ S] sair
- =====

Digite uma opcao: 5

Erro! Nao ha elementos no estoque para serem ordenados.



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em função do preço
  - [ 6] Calcular valor da compra
  - [ S] Sair
- =====

Digite uma opção: 6

Você escolheu calcular o valor da compra!

Escolha a quantidade de tipos de itens que o(a) senhor(a) deseja comprar: 3

Insira o nome do produto: pao

Insira a quantidade que deseja comprar: 10

Insira o nome do produto: picanha

Insira a quantidade que deseja comprar: 3

Insira o nome do produto: escova de dente

Insira a quantidade que deseja comprar: 3

Valor total da compra: R\$229.70



===== MENU =====

- [ 1] Adicionar produto
  - [ 2] Remover produto
  - [ 3] Exibir estoque
  - [ 4] Buscar produto
  - [ 5] Ordenar produtos em funcao do preco
  - [ 6] Calcular valor da compra
  - [ S] Sair
- =====

Digite uma opcao: 6

Erro! Nao ha elementos no estoque para realizar uma compra.



```
===== MENU =====
[ 1] Adicionar produto
[ 2] Remover produto
[ 3] Exibir estoque
[ 4] Buscar produto
[ 5] Ordenar produtos em funcao do preco
[ 6] Calcular valor da compra
[ S] Sair
=====
Digite uma opcao: S
Ate a proxima!
```



# Benefícios



Organização



Redução de Perdas por Vencimento



Aumento das Vendas



Rapidez no Atendimento

# Obrigado

