

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

CAMILA AYA SAITO (15635649)
VICTORIA FAVERO NUNES (15698302)
VINICIUS MORAES DE CARVALHO (15642432)

Trabalho de Estruturas de Dados 1 - Leilão

SÃO CARLOS
2024

INTRODUÇÃO

O grupo optou pela codificação do sistema de compra online, no estilo de leilão, que foi a opção 2 proposta para o trabalho. Nesse sistema, foram implementadas as seguintes funções que o usuário do código poderá utilizar: cadastrar um novo produto, listar todos os lances e produtos cadastrados, dar um novo lance em determinado produto, dar uma sugestão de um produto para se dar um lance, remover um produto da lista e encerrar o leilão. Para a implementação do código, foi utilizado um Tipo Abstrato de Dados (TAD) de uma lista de pilhas de filas, todas encadeadas e dinâmicas, um arquivo principal (main) e um Makefile para rodar todos os arquivos necessários.

ESTRUTURAS DE DADOS

```
typedef struct pessoa{
    char nome[50];
    struct pessoa *prox;
}pessoa;
```

Na estrutura *pessoa*, que servirá para fazer uma fila de pessoas para cada lance dado, há uma variável *nome* (string de 50 caracteres) para armazenar o nome da pessoa que deu o lance e um ponteiro *prox* que aponta para outra estrutura do tipo *pessoa*.

```
typedef struct lance{
    float valor;
    pessoa *inicio_pessoas;
    struct lance *prox;
}lance;
```

Na estrutura *lance*, que armazenará todas as informações de um lance, há uma variável *valor* do tipo float para armazenar o valor do lance que foi dado, um ponteiro *inicio_pessoas* para uma estrutura do tipo *pessoa*, que apontará para a fila de pessoas daquele lance, e um ponteiro *prox* do tipo *struct lance*, que apontará para o próximo lance que será dado. Essa estrutura servirá para fazer a pilha de lances de cada produto.

```
typedef struct produto{
    char nome_produto[50];
    char descricao[300];
    lance *lances;
    struct produto *prox;
}produto;
```

Na estrutura *produto*, que armazenará todos os dados de um produto, há uma string *nome_produto* de 50 caracteres e outra string *descricao* de 300 caracteres, onde serão armazenados o nome daquele produto e sua descrição, respectivamente, um ponteiro *lances* do tipo *lance*, que apontará para a pilha de lances daquele produto, e um ponteiro *prox* do tipo *struct produto*, no qual apontará para o próximo produto da lista. Essa estrutura servirá para estruturar a lista ordenada de produtos cadastrados.

```
typedef struct lista_produtos{
    produto *inicio;
    produto *fim;
    int tamanho;
}lista_produtos;
```

Na estrutura *lista_produtos* estarão as informações principais da lista de todos os produtos cadastrados. Nela, há um ponteiro *inicio* e outro *fim*, ambos do tipo *produto*, que apontarão para o início e para o fim da lista, respectivamente, e há uma variável do tipo inteiro *tamanho* que armazenará o tamanho da lista. Essa será a estrutura que o usuário deverá declarar para manusear a lista.

FUNÇÕES DO TAD

criar_lista_produtos: cria uma lista de produtos l;

inserir_produto: insere um produto na lista de produtos;

inserir_lance: insere um lance na pilha de lances de um produto;

quantidade_produtos: retorna quantos produtos a lista tem;

quantidade_lances: retorna a quantidade de lances que um produto tem;

proximo_produto: retorna o próximo produto da lista;

proximo_lance: retorna o próximo lance da pilha;

proxima_pessoa: retorna a próxima pessoa da fila;

nome_produto: retorna o nome (por parâmetro) o de determinado produto;

descricao_produto: retorna a descrição (por parâmetro) de determinado produto;

valor_lance: retorna o valor (por parâmetro) de um lance de determinado produto;

nome_pessoa: retorna o nome (por parâmetro) da pessoa que deu determinado lance;

vencedor_produto: retorna quem foi o vencedor de determinado produto;

remover_produto: remove um produto da lista;

sugerir_produtos_para_pessoa: sugere um produto para determinada pessoa dar um novo lance;

esvaziar_lista: esvazia toda a lista e libera toda a memória alocada;

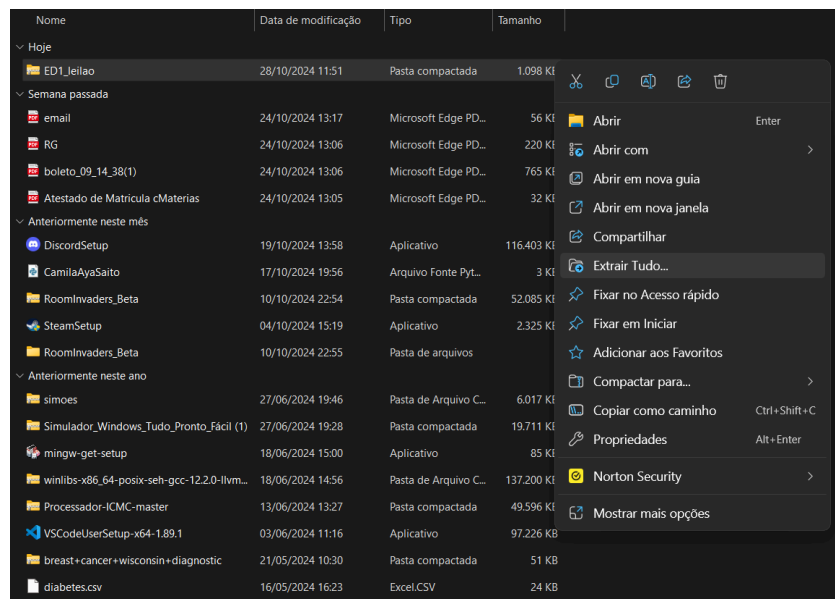
lista_vazia: verifica se a lista está vazia;

produto_nao_existe: verifica se um produto existe na lista ou não;

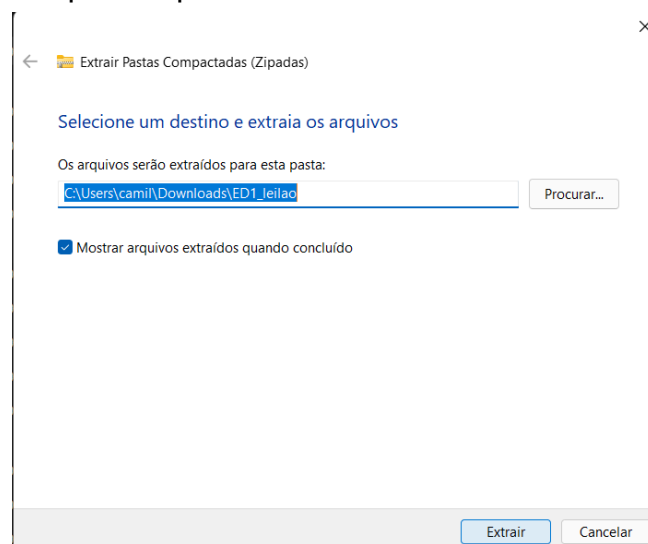
COMPILAÇÃO DO PROGRAMA

Para compilar o programa, foi utilizado o MinGW (compilador para sistemas Windows conseguirem usar o GCC) versão 6.3.0, no sistema operacional Windows. No programa, foram utilizadas as bibliotecas <stdio.h>, <stdlib.h> e <string.h>, que são bibliotecas padrões do C, além do TAD <leilao.h> implementado.

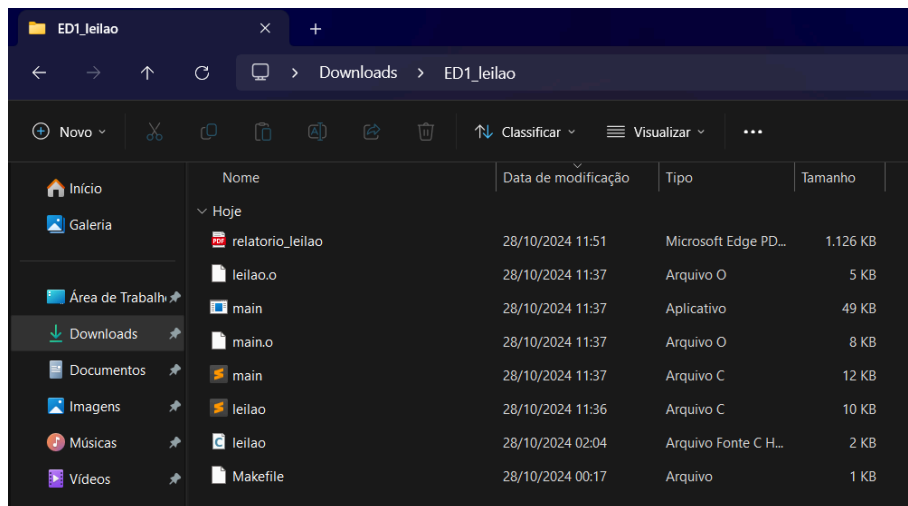
Primeiramente, é necessário extrair os arquivos da pasta compactada. Para isso, clique com o botão direito sobre a pasta e selecione a opção *Extrair tudo*.



Após isso, selecione o local para onde deseja instalar a pasta com os arquivos extraídos e depois clique em *Extrair*.



Com isso, os arquivos estarão extraídos e prontos para serem utilizados.



Para executar o programa, é necessário ter o MinGW instalado no sistema, abrir o prompt de comando e acessar, por linha de comando, a pasta (com os arquivos já extraídos) em que os arquivos estão (*cd 'caminho da pasta'*). No exemplo, os arquivos estão na pasta *C:\Users\camil\Downloads\ED1_leilao*, então o comando será *cd C:\Users\camil\Downloads\ED1_leilao*.

```
Prompt de Comando

Microsoft Windows [versão 10.0.22631.4317]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\camil>cd C:\Users\camil\Downloads\ED1_leilao|
```

Depois, é necessário compilar os arquivos. Para isso, utilize o comando *mingw32-make.exe*, que, através do *Makefile*, irá compilar todos os arquivos necessários se uma só vez.

```
Prompt de Comando

Microsoft Windows [versão 10.0.22631.4317]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\camil>cd C:\Users\camil\Downloads\ED1_leilao

C:\Users\camil\Downloads\ED1_leilao>mingw32-make.exe
gcc main.o leilao.o -o main
```

Por fim, para executar o programa, digite o comando *main.exe*, que irá fazer o programa rodar.

```
Prompt de Comando - main.e X + v
Microsoft Windows [versão 10.0.22631.4317]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\camil>cd C:\Users\camil\Downloads\ED1_leilao

C:\Users\camil\Downloads\ED1_leilao>mingw32-make.exe
gcc main.o leilao.o -o main

C:\Users\camil\Downloads\ED1_leilao>main.exe
-----

**LEILAO ICMC**
1) Cadastrar um produto
2) Listar produtos e lances
3) Dar um lance
4) Encerrar leilao
5) Remover produto
6) Sugestao de produtos
0) Sair

Escolha uma das opcoes acima:
```

E caso se queira limpar o prompt de comando, digita-se o comando `c/s`.