

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

SCC0223 - Estrutura de Dados 1
Trabalho 1

Relatório: Sistema de Compra Online (SCO)

Grupo: Anderson Lavinsky, nºUSP 9045195

Joao Bienotti, nºUSP 11218452

Stefane Oliveira, nºUSP 10829970

Responsável: Prof. Dr. Thiago A. S. Pardo

1. INTRODUÇÃO

Este trabalho consistiu em um desenvolvimento de um Sistema de Compra Online (SCO) em linguagem C usando estrutura de Pilha, Fila e Lista em duas versões: estática-sequencial e dinâmica-encadeada. O SCO baseia-se no estilo de leilão, em que vários usuários dão lances nos produtos disponíveis sendo vendidos. O dono do lance maior em um determinado momento leva o prêmio. Quando houver mais de um lance com o mesmo valor, quem deu o lance primeiro leva o prêmio.

Qualquer produto que se queira vender pode ser cadastrado no sistema. Em um determinado momento, vários produtos podem estar disponíveis para o leilão e o usuário pode decidir em qual produto dar o lance. Um usuário pode dar quantos lances desejar para cada produto. Cada novo lance tem que ser igual ou maior do que o lance anterior para um determinado produto. Dessa forma, pode ocorrer de vários usuários darem o mesmo lance para um produto. Nesse caso, deve-se guardar em uma lista os usuários que deram o mesmo lance para o produto e, ao final do leilão, o usuário cujo lance chegou primeiro ficará com o produto em questão. Ao ser encerrado o leilão, são apresentados os usuários ganhadores (maiores lances) para cada produto. A forma como este sistema foi implementado em linguagem C é apresentada a seguir de maneira sucinta.

O SCO foi implementado respeitando as condições exigidas do uso de listas (para os produtos cadastrados, de forma ordenada), de pilhas (para cadastro de lances efetivados para cada produto) e de Filas (para o registro dos usuários que tiveram seus lances aceitos por produto). Com o intuito de facilitar a geração do arquivo objeto (*file.o*) a ser usado pela função `SCOMain.c` (código principal), foram incorporadas todas as estruturas e funções das TADs Fila, Pilha e Lista na TAD `SCO.c` e sua respectiva biblioteca `SCO.h`. Funções estas que são descritas na seção abaixo, separadas por Pilha, Fila e Lista tanto para a versão estática-sequencial e quanto para a dinâmica-encadeada do sistema.

2. ESTRUTURA DE DADOS UTILIZADAS

A função **menuOpcoes()** é do tipo *void* da Biblioteca SCO e tem como objetivo a impressão em tela do menu do Sistema de Compra Online. Utilizada na função **main** em ambas as versões implementadas. As demais são específicas de cada versão.

2.1. Versão estática-sequencial

2.1.1. Estrutura Pilha

As rotinas foram implementadas de acordo com as *structs* adotadas para a realização deste trabalho. Seguem as descrições das *structs*, variáveis e funções.

typedef float elem_p – Tipo: elemento de Pilha. Usado nessa interface para indicar o tipo de valores que podem ser armazenados na Pilha. Neste trabalho, o *elem_p* foi usado para armazenar valores dos lances dados ao(s) produto(s).

typedef struct PILHA – Tipo: *struct* de Pilha. Representa o tipo abstrato usado para armazenar os elementos (*elem_p*) a serem adicionados. A *struct* foi definida com um ponteiro *int topo* e um atributo *elem_p itens[TAM]*. A estrutura concreta não foi definida na interface e, dessa forma, os usuários não têm acesso aos campos.

create(PILHA* P) – Função do tipo *void* da Biblioteca Pilha. Cria uma Pilha vazia alocando um espaço de memória de forma estática contendo o ponteiro *topo* com valor um negativo (-1). Utilizada dentro da função **inserir_produto**.

push(PILHA* P, elem_p* x, int* erro) – Função do tipo *void* da Biblioteca Pilha. Adiciona o valor do lance (*elem_p*) dentro da Pilha na posição *topo*. Ao ser adicionado o atributo *itens[TAM]* armazena o elemento e o ponteiro *topo* é atualizado, caso a Pilha não esteja cheia. Caso a função seja chamada e a Pilha esteja cheia, retorna a *flag* de erro indicando não ser possível realizar a adição do elemento à Pilha. O valor a ser adicionado respeita a estrutura de Pilha, ou seja, será inserido no topo da Pilha. Utilizada nas funções **inverter**, **listar_produtos** e **inserir_lance**.

pop(PILHA* P, elem_p* x, int* erro) – Função do tipo *void* da Biblioteca Pilha. Remove o lance (*elem_p*) do topo da Pilha retornando o seu valor. Ao ser removido o atributo *itens[TAM]* libera o elemento e o ponteiro *topo* é atualizado, caso a Pilha não esteja vazia e o elemento esteja na Pilha. Caso a função seja chamada e a Pilha esteja vazia e/ou o elemento não esteja na Pilha, retorna a *flag* de erro indicando não ser possível realizar a remoção do elemento da Pilha. O valor a ser removido respeita a

estrutura de Pilha, ou seja, será o valor encontrado no topo da Pilha. Utilizada nas funções ***inverter***, ***listar_produtos*** e ***ganhador***.

IsFull(PILHA* P) – Função do tipo *int* da Biblioteca Pilha. Testa se a Pilha encontra-se cheia comparando o valor do ponteiro *topo* com o tamanho da Pilha. Retorna o valor *1* em caso positivo, ou seja, Pilha cheia, e *0* em caso negativo. Utilizada nas funções ***push*** e ***inserir_lance***.

IsEmpty(PILHA* P) – Função do tipo *int* da Biblioteca Pilha. Testa se a Pilha encontra-se vazia verificando o valor do ponteiro *topo* se igual a menos um. Retorna o valor *1* em caso positivo, ou seja, Pilha vazia, e *0* em caso negativo. Utilizada nas funções ***pop***, ***tmPilha***, ***inverter***, ***inserir_lance***, ***ganhador*** e ***encerrarLeilao***.

tmPilha(PILHA* P, int* erro) – Função do tipo *int* da Biblioteca Pilha. Retorna o tamanho da Pilha.

inverter(PILHA* P) – Função do tipo *void* da Biblioteca Pilha. Inverte a ordem dos elementos presentes na Pilha. Utilizada na função ***listar_produtos***.

2.1.2. Estrutura Fila

As rotinas foram implementadas de acordo com as *structs* adotadas para a realização deste trabalho. Seguem as descrições das *structs*, variáveis e funções.

typedef char elem_f[10] – Tipo: elemento de Fila. Usado nessa interface para indicar o tipo de valores que podem ser armazenados na Fila. Neste trabalho, a Fila foi usada para armazenar o(s) nome(s) do(s) cliente(s) que efetivou(aram) o(s) lance(s) ao(s) produto(s).

typedef struct FILA – Tipo: *struct* de Fila. Representa o tipo abstrato usado para armazenar os elementos (*elem_f*) a serem adicionados. A *struct* foi definida com os ponteiros *int inicio*, *int fim* e *int total*; além de um atributo *elem_f itens[TAM]*. A estrutura concreta não foi definida na interface e, dessa forma, os usuários não têm acesso aos campos.

Criar(FILA* F) – Função do tipo *void* da Biblioteca Fila. Cria uma Fila vazia alocando um espaço de memória de forma estática contendo os ponteiros *inicio*, *fim* e *total* com valores iguais a zero (0). Utilizada na função ***inserir_produto***.

EstaCheia(FILA* F) – Função do tipo *int* da Biblioteca Fila. Testa se a Pilha encontra-se cheia comparando o valor do ponteiro *total* com o tamanho da Fila. Retorna o valor *1* em caso positivo, ou seja, Pilha cheia, e *0* em caso negativo. Utilizada na função ***Entra***.

EstaVazia(FILA* F) – Função do tipo *int* da Biblioteca Fila. Testa se a Fila encontra-se vazia verificando o valor do ponteiro *total* se igual a zero. Retorna o valor 1 em caso positivo, ou seja, Fila vazia, e 0 em caso negativo. Utilizada nas funções ***Sai*** e ***listar_produtos***.

Entra(FILA* F, char* x, int* erro) – Função do tipo *void* da Biblioteca Filha. Adiciona o elemento específico (*elem_f*) dentro da Fila na posição fim. Ao ser adicionado o atributo *itens[TAM]* armazena o elemento e o ponteiro *fim* é atualizado, caso a Fila não esteja cheia. Caso a função seja chamada e a Fila esteja cheia, retorna a *flag* de erro indicando não ser possível realizar a adição do elemento à Fila. O valor a ser adicionado respeita a estrutura da Fila, ou seja, será adicionado ao final da Fila. A adição do elemento foi realizada utilizando-se a função *strcpy* da Biblioteca Ctype; onde os argumentos usados foram o ponteiro *itens* recebendo o elemento a ser adicionado na posição final da Fila através do ponteiro *fim*, seguido da atualização do ponteiro *fim* da Fila. Utilizada nas funções ***listar_produtos*** e ***inserir_lance***.

Sai(FILA* F, char* x, int* erro) – Função do tipo *void* da Biblioteca Fila. Remove o elemento específico (*elem_f*) do início da Fila retornando o seu valor. Ao ser removido o atributo *itens[TAM]* libera o elemento e o ponteiro *inicio* é atualizado, caso a Fila não esteja vazia e o elemento esteja na Fila. Caso a função seja chamada e a Fila esteja vazia, retorna a *flag* de erro indicando não ser possível realizar a remoção do elemento da Fila. O valor a ser removido respeita a estrutura da Fila, ou seja, será removido do início da Fila. A remoção do elemento foi realizada utilizando-se a função *strcpy* da Biblioteca Ctype; onde os argumentos usados foram o elemento a ser removido na posição inicial da Fila e o ponteiro *itens* contendo a posição do elemento na Fila através do ponteiro *inicio*, seguido da atualização do ponteiro *inicio* da Fila. Utilizada na função ***listar_produtos***.

invertFila(FILA* F) – Função do tipo *void* da Biblioteca Fila. Inverte os elementos de uma Fila. Utilizada na função ***listar_produtos***.

2.1.3. Estrutura Lista

As rotinas foram implementadas de acordo com as *structs* adotadas para a realização deste trabalho. Seguem as descrições das *structs*, variáveis e funções.

typedef char elem[20] – Tipo: elemento de Lista. Usado nessa interface para indicar o tipo de valores que podem ser armazenados na Lista. Neste caso, a Lista foi usada para armazenar valores do tipo *char*, os quais serão os produtos cadastrados.

typedef struct NO – Tipo: *struct* de Lista. Representa a estrutura usada para armazenar os elementos (*elem*) no campo *item*. Além disso, consiste de um campo *lances* do tipo *PILHA* e *nomes* do tipo *FILA*.

typedef struct LISTA – Tipo: *struct* de Lista. Representa o tipo abstrato usado para armazenar os elementos (*elem*) a serem adicionados. A *struct* foi definida com o ponteiro *int total* e o campo *produtos[TAM]* do tipo *NO*.

createL(LISTA* L_prod) – Função do tipo *void* da Biblioteca Lista. Cria uma Lista vazia alocando um espaço de memória de forma estática contendo os ponteiros *inicio*, *fim* e *total* com valores iguais a zero. Esta Lista armazena os produtos a serem leiloados. Utilizada na função ***main***.

EstaCheiaL(LISTA* L_prod) – Função do tipo *int* da Biblioteca Lista. Testa se a Lista encontra-se cheia comparando o valor do ponteiro *total* com o tamanho da Lista. Retorna o valor um (1) em caso positivo, ou seja, Lista cheia, e zero (0) em caso negativo. Utilizada na função ***inserir_produto***.

EstaVaziaL(LISTA* L_prod) – Função do tipo *int* da Biblioteca Lista. Testa se a Lista encontra-se vazia verificando o valor do ponteiro *total* se igual a zero. Retorna o valor 1 em caso positivo, ou seja, Lista vazia, e 0 em caso negativo. Utilizada nas funções ***remover_produto*** e ***listar_produtos***.

inserir_produto(LISTA* L_prod, char objeto[], int* erro) – Função do tipo *void* da Biblioteca SCO. Insere um produto de forma ordenada caso a Lista não esteja cheia atualizando o ponteiro *total*. Em caso contrário, retorna uma *flag* de erro. A ordenação da Lista foi realizada através da função *strcmp* da Biblioteca Ctype, comparando o item já incluso na Lista com o novo a ser adicionado. A função *strcpy* foi utilizada para adicionar o produto para a Lista de produtos. Além disso, são criadas a Fila de nomes e Pilha de lances a serem usadas posteriormente.

remover_produto(LISTA* L_prod, char objeto[], int* erro) – Função do tipo *void* da Biblioteca SCO. Remove o produto da Lista atualizando o ponteiro *total*, caso a Lista não se encontre vazia. Em caso contrário, retorna uma *flag* de erro. A remoção é feita com o auxílio da função *strcmp* da Biblioteca Ctype, mantendo-se a ordem da Lista de produtos. São atualizadas a Fila de nomes e a Pilha de lances. Utilizada na função

listar_produtos(LISTA* L_prod, int* erro) – Função do tipo *void* da Biblioteca SCO. Imprime na tela os produtos existentes na Lista com seu(s) respectivo(s) lance(s) e o(s) nome(s) relacionado(s) a cada lance. Utilizada na função ***main***.

inserir_lance(LISTA* L_prod, char objeto[], float lance, char nome[], int* erro) – Função do tipo *void* da Biblioteca SCO. Insere um lance a um produto existente na

Lista assim como o nome do cliente que está realizando o lance. São atualizadas a Fila de nomes e a Pilha de lances do produto em questão. Utilizada na função **main**.

remover_lance(LISTA* L_prod, char objeto[], float lance, char nome[], int* erro) – Função do tipo *void* da Biblioteca SCO. Remove um lance a um produto existente na Lista assim como o nome do cliente que realizou o lance. São atualizadas a Fila de nomes e a Pilha de lances do produto em questão. Utilizada na função **main**.

recomendacao(LISTA* L_prod, char pessoa[]) – Função do tipo *void* da Biblioteca SCO. Recomenda os produtos que um determinado cliente não efetuou nenhum lance. Utilizada na função **main**.

ganhador(LISTA L_prod, elem objeto, char* gan) – Função do tipo *void* da Biblioteca SCO. Verifica o cliente que tem o melhor lance (considerando o primeiro em caso de empate) para um produto em específico. Utilizada na função **encerrarLeilao**.

encerrarLeilao(LISTA* L_prod) – Função do tipo *void* da Biblioteca SCO. Encerra o Sistema de Compras Online verificando o desfecho do leilão. Ou seja, encerra o leilão listando o(s) usuário(s) e o lance ganhador para cada produto, seguido da reinicialização do sistema para que novos leilões possam ocorrer.

menuOpcoes() – Função do tipo *void* da Biblioteca SCO. Imprime na tela o menu do Sistema de Compra Online a cada etapa de interação do usuário com o programa. Utilizada na função **main**.

ajudaLeilao(char ajuda[N]) – Função do tipo *void* da Biblioteca SCO. Abre e imprime na tela um arquivo de texto informativo sobre cada opção disponível no menu, para auxílio ao usuário no uso do sistema.

2.2. Versão dinâmica-encadeada:

2.2.1. Estrutura Pilha

As rotinas foram implementadas de acordo com as *structs* adotadas para a realização deste trabalho. Seguem as descrições das *structs*, variáveis e funções.

typedef float elem_p – Tipo: elemento de Pilha. Usado nessa interface para indicar o tipo de valores que podem ser armazenados na Pilha. Neste trabalho, o *elem_p* foi usado para armazenar valores dos lances dados ao(s) produto(s).

typedef struct PDEnc – Tipo: *struct* de Pilha. Representa o tipo abstrato usado para armazenar os elementos (*elem_p*) a serem adicionados. A *struct* foi definida com um ponteiro *NoP *topo*. A estrutura concreta não foi definida na interface e, dessa forma, os usuários não têm acesso aos campos.

typedef struct NoP – Tipo struct de No de Pilha contendo um ponteiro *NoP *prox* e um atributo *elem_p info*.

CriarPDEnc(void) – Função do tipo *PDEnc* da Biblioteca Pilha. Cria uma Pilha vazia alocando um espaço de memória de forma dinâmica contendo o ponteiro *topo* apontando para NULL. Utilizada dentro das funções ***inserir_produto*** e ***listar_produtos***.

DestroirPenc2(PDEnc* PE) – Função do tipo *void* da Biblioteca Pilha. Libera o espaço de memória que foi alocado para a Pilha e aponta o ponteiro para NULL. Utilizada nas funções ***Remove_LDEnc***, ***destruirL*** e ***listar_produtos*** e ***inserir_lance***.

Estavazia_PDEnc (PDEnc *PE) – Função do tipo *int* da Biblioteca Pilha. Testa se a Pilha encontra-se vazia verificando o valor do ponteiro *topo* se igual a menos um. Retorna o valor *1* em caso positivo, ou seja, Pilha vazia, e *0* em caso negativo. Utilizada nas funções ***Remove_PDEnc***, ***inserir_lance***, ***encerrarLeilao***, ***insere_PDEnc***.

Insere_PDEnc(PDEnc *PE, elem_p *X, int *erro) – Função do tipo *void* da Biblioteca Pilha. Aloca dinamicamente um *NoP* auxiliar testando em seguida essa alocação e, se ok, atualiza o campo *info* com o novo elemento assim como os campos *prox* e *topo*. Caso contrário, retorna a *flag* de erro. Utilizada nas funções ***listar_produtos*** e ***inserir_lance***.

Remove_PDEnc(PDEnc *PE, elem_p *X, int *erro) – Função do tipo *void* da Biblioteca Pilha. Testa se a Pilha encontra-se vazia verificando o valor do ponteiro *topo* se igual a menos um. Retorna o valor *1* em caso positivo, ou seja, Pilha vazia, e se *0* remove o elemento atualizando os campos *info*, *prox* e *topo*. Utilizada nas funções ***listar_produtos*** e ***ganhador***.

2.2.2. Estrutura Fila

typedef char elem_f – Tipo: elemento de Fila. Usado nessa interface para indicar o tipo de valores que podem ser armazenados na Fila. Neste trabalho, a Fila foi usada para armazenar o(s) nome(s) do(s) cliente(s) que efetivou(aram) o(s) lance(s) ao(s) produto(s).

typedef struct FDEnc – Tipo: *struct* de Fila. Representa o tipo abstrato usado para armazenar os elementos (*elem_f*) a serem adicionados. A *struct* foi definida com os ponteiros *NoF *inicio* e *NoF *fim* e os atributos *int total* e *elem_f itens*. A estrutura concreta não foi definida na interface e, dessa forma, os usuários não têm acesso aos campos.

typedef struct NoF – Tipo struct de No de Fila contendo um ponteiro *NoF *prox* e um *elem_f *info*.

Criar_FDEnc(void) – Função do tipo *FDEnc* da Biblioteca Fila. Cria uma Fila vazia alocando um espaço de memória de forma dinâmica contendo os ponteiros *inicio*, *fim* e *total* com valores iguais a *NULL*, *NULL* e zero (0), respectivamente. Utilizada na função ***inserir_produto***.

fila_vazia(FDEnc *FE) – Função do tipo *int* da Biblioteca Fila. Testa se a Fila encontra-se vazia comparando o valor do ponteiro *total* com o tamanho da Fila. Retorna o valor um (1) em caso positivo, ou seja, Fila vazia, e zero (0) em caso negativo. Utilizada na função ***Rem_FDEnc***.

Proc_FDEnc (FDEnc *FE, elem_f *X) – Função do tipo *int* da Biblioteca Fila. Procura um elemento na Fila retornando um valor um (1) quando achado e zero (0) quando não ou Fila vazia.

Destroir_FDEnc (FDEnc *FE) – Função do tipo *void* da Biblioteca Fila. Percorre todos os *NoFs* para liberar (“desalocar”) o espaço a memória que foi reservada anteriormente, finalizando com a Fila sendo liberada. Utilizada nas funções ***destruirL*** e ***RemoverLDenc***.

Inserer_FDEnc (FDEnc *FE, elem_f *X, int *erro) – Função do tipo *void* da Biblioteca Fila. Aloca um espaço de memória para o elemento específico (*elem_f*) ser adicionado na Fila na posição final. Ao ser adicionado o atributo *itens* armazena o elemento e o ponteiro *fim* e *total* são atualizados, caso a Fila não esteja cheia. Caso a função seja chamada e a Fila esteja cheia, retorna a *flag* de erro indicando não ser possível realizar a adição do elemento à Fila. O valor a ser adicionado respeita a estrutura da Fila, ou seja, será adicionado ao final da Fila. A adição do elemento foi realizada utilizando-se a função *strcpy* da Biblioteca Ctype; onde os argumentos usados foram o ponteiro *itens* recebendo o elemento a ser adicionado na posição final da Fila através do ponteiro *fim*, seguido da atualização do ponteiro *fim* da Fila. Utilizada nas funções ***listar_produtos*** e ***inserir_lance***.

Rem_FDEnc(FDEnc *FE, int *erro) – Função do tipo *elem_f* da Biblioteca Fila. Remove o elemento da Fila e atualiza os ponteiros *inicio*, *prox* e *total*. Utilizada nas funções ***invertFila***, ***listar_produtos*** e ***ganhador***.

invertFila(FDEnc *FE) – Função do tipo *void* da Biblioteca Fila. Inverte os elementos de uma Fila com o auxílio de uma variável local para armazenar os elementos e voltar para a estrutura de Fila original. A inversão do elemento foi realizada utilizando-se a função *strcpy* da Biblioteca Ctype; onde os argumentos usados foram o elemento a ser removido na posição inicial da Fila e variável auxiliar local. Utilizada nas funções ***listar_produtos*** e ***ganhador***.

2.2.3. Estrutura Lista

typedef char elem_l – Tipo: elemento de Lista. Usado nessa interface para indicar o tipo de valores que podem ser armazenados na Lista. Neste caso, a Lista foi usada para armazenar valores do tipo *char*, os quais serão os produtos cadastrados.

typedef struct NoL – Tipo: *struct* de Lista. Representa a estrutura usada para armazenar os elementos (*elem_l*) no campo **info*. Além disso, consiste de outros ponteiros *PDEnc *lances* para a estrutura de Pilha, *FDEnc *nomes* para estrutura de Fila e um *noL *prox*.

typedef struct LDEnc – Tipo: *struct* de Lista. Representa o tipo abstrato usado para armazenar os elementos (*elem_l*) a serem adicionados. A *struct* foi definida com os ponteiros *NoL inicio* e *fim* e outro ponteiro *int total*.

criarL(void) – Função do tipo *void* da Biblioteca Lista. Essa função é que cria Lista vazia alocando um espaço de memória de forma estática contendo os ponteiros *inicio*, *fim* e *total* com valores iguais a zero. Esta Lista armazena os produtos a serem leiloados. Utilizada na função ***main***.

destruirL(LDEnc *L_prod) - Função do tipo *void* da Biblioteca Lista. Percorre toda a Lista, nó (*NoL*) por nó, para liberar a memória (desalocar) reservada para os *NoLs*, liberando memórias de Pilha e Fila associada ao *NoL*. Em seguida libera a memória reservada a Lista. Utilizada na função ***encerrarLeilao***.

EstaCheiaL(LISTA* L_prod) – Função do tipo *int* da Biblioteca Lista. Testa se a Lista encontra-se cheia comparando o valor do ponteiro *total* com o tamanho da Lista. Retorna o valor um (1) em caso positivo, ou seja, Lista cheia, e zero (0) em caso negativo. Utilizada na função ***inserir_produto***.

EstaVaziaL(LISTA* L_prod) – Função do tipo *int* da Biblioteca Lista. Testa se a Lista encontra-se vazia verificando o valor do ponteiro *total* se igual a zero. Retorna o valor um (1) em caso positivo, ou seja, Lista vazia, e zero (0) em caso negativo. Utilizada nas funções ***remover_produto*** e ***listar_produtos***.

inserir_produto(LDEnc *L_prod, elem_l *objeto, int *erro) – Função do tipo *void* da Biblioteca SCO. Insere um produto de forma ordenada caso a Lista não esteja cheia atualizando o ponteiro *total*. Em caso contrário, retorna uma *flag* de erro. A ordenação da Lista foi realizada através da função *strcmp* da Biblioteca Ctype, comparando o item já incluso na Lista com o novo a ser adicionado. A função *strcpy* foi utilizada para adicionar o produto para a Lista de produtos. Além disso, são criadas a

Fila de nomes e Pilha de lances a serem usadas posteriormente. Utilizada na função **main**.

Remover_LDEnc(LDEnc *L_prod, elem_f *X, int *erro) – Função do tipo *void* da Biblioteca SCO. Remove o produto da Lista atualizando o ponteiro *total*, caso a Lista não se encontre vazia. Em caso contrário, retorna uma *flag* de erro. A remoção é feita com o auxílio da função *strcmp* da Biblioteca Ctype, mantendo-se a ordem da Lista de produtos. São atualizadas a Fila de nomes e a Pilha de lances. Utilizada na função **main**.

listar_produtos(LDEnc *L_prod, int *erro) – Função do tipo *void* da Biblioteca SCO. Imprime na tela os produtos existentes na Lista com seu(s) respectivo(s) lance(s) e o(s) nome(s) relacionado(s) a cada lance. Utilizada na função **main**.

inserir_lance(LDEnc *L_prod, char objeto[], float lance, char nome[], int *erro) – Função do tipo *void* da Biblioteca SCO. Insere um lance a um produto existente na Lista assim como o nome do cliente que está realizando o lance. A verificação de produto na Lista foi realizada através da função *strcmp* da Biblioteca Ctype. São atualizadas a Fila de nomes e a Pilha de lances do produto em questão. Utilizada na função **main**.

recomendacao(LDEnc *L_prod, char pessoa[]) – Função do tipo *void* da Biblioteca SCO. Recomenda os produtos que um determinado cliente não efetuou nenhum lance. Utilizada na função **main**.

ganhador(LDEnc *L_prod, elem_f *produto, char *gan, int *erro) – Função do tipo *void* da Biblioteca SCO. Verifica o cliente que tem o melhor lance (considerando o primeiro em caso de empate) para um produto em específico. A consulta ao produto foi realizada através da função *strcmp* e a cópia do nome ganhador do produto com auxílio da função *strcpy* (ambas funções pertencentes à Biblioteca Ctype). Utilizada na função **encerrarLeilao**.

encerrarLeilao(LDEnc* L_prod) – Função do tipo *void* da Biblioteca SCO. Encerra o Sistema de Compras Online verificando o desfecho do leilão. Ou seja, encerra o leilão listando o(s) usuário(s) e o lance ganhador para cada produto, seguido da reinicialização do sistema para que novos leilões possam ocorrer. Utilizada na função **main**.

menuOpcoes() – Função do tipo *void* da Biblioteca SCO. Imprime na tela o menu do Sistema de Compra Online a cada etapa de interação do usuário com o programa. Utilizada na função **main**.

ajuda (void) – Função do tipo *void* da Biblioteca SCO. Abre e imprime na tela uma sequência de texto informativo sobre cada opção disponível no menu, para auxílio ao usuário no uso do sistema.

3. Funcionamento do Sistema de Compra Online (SCO)

A forma de gerar os programas executáveis para ambas versões do SCO consistiu primeiramente em usar a linha de comando

```
gcc -c SCO_TAD.c
```

colocando-se os arquivos .c e .h em uma mesmo local (pasta) no computador para que fosse gerado o arquivo objeto, o qual foi usado para gerar o arquivo executável do sistema. Para isso foi usado uma outra linha de comando

```
gcc SCO_TAD.o SCOMain.c -o SCO.exe
```

comandos estes que foram executados em um terminal do prompt do windows, por exemplo. De posse do arquivo os testes foram realizados mantendo-se o mesmo ambiente (*prompt* de comando). Ao executar o programa ele apresenta uma tela principal com os menus disponíveis para interação (Fig.1).

Figura 1. Tela principal do Sistema de Compra Online. Ambas versões possuem a mesma tela.

```
##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
```

A opção 1 (Cadastrar um produto) solicita ao usuário a entrada de um produto e/ou suas características para que seja cadastrado em uma Lista de produtos, como

mostrado na Figura 2. Podendo realizar novamente outro cadastro. A cada cadastro a lista de produtos é atualizada.

Figura 2. Opção 1 do Sistema de Compra Online.

```
##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
1
Entre com o nome do produto: televisao
Produto cadastrado com sucesso!
```

A opção 2 (Remover um produto) solicita ao usuário que entre com o nome do produto que queira remover e, se confirmado em Lista, a ação será concluída.

Figura 3. Opção 2 do Sistema de Compra Online.

```
##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
2
Entre com o nome do produto: televisao
Produto retirado com sucesso!
```

A Opção 3 (Efetuar um lance) solicita ao usuário a entrada de seu nome e produto que esteja almejando efetivar um lance. A Fig. 4 mostra a ilustração de uma efetivação de lance de um usuário a um produto.

Figura 4. Opção 3 do Sistema de Compra Online.

```

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

2 O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
  Entre com o nome do produto: televisao
  Produto removido com sucesso!

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

3 O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
  Entre com o seu nome: Stefane
  Entre com o nome do produto: televisao
  Entre com o valor do lance: 450
  Produto nao cadastrado no leilao!

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)

```

A Opção 4 (Listar produtos e lances) exibe em tela todos os produtos que se encontrem cadastrados em Lista juntamente com o(s) nome(s) daquele(s) que efetuou o lance(s), como mostrado na Fig. 5 abaixo.

Figura 5. Opção 4 do Sistema de Compra Online.

```

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
4
Produto 1: televisao
Lance de R$ 450.00 dado por Stefane
Listagem completa!

```

A Opção 5 (Indicar outros produtos para lances) solicita ao usuário que entre com seu nome para que possa ser recomendado outros produtos cadastrados que ainda não foram efetivados com lance(s) por ele (Fig.6).

Figura 6. Opção 5 do Sistema de Compra Online.

```

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

    O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
3
    Entre com o seu nome: Marcia
    Entre com o nome do produto: celular
    Entre com o valor do lance: 700
    Seu lance foi aceito com sucesso!

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

    O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
5
    Entre com o seu nome: Marcia
para Marcia: nao gostaria de dar um lance pela(o):
apartamento radio ?

```

A Opção 6 (Encerrar leilão) verifica e imprime em tela cada produto cadastrado no sistema de leilão com o lance efetivado vencedor e o seu respectivo ganhador. Os demais produtos que não receberam lance(s) também são impressos, mostrando um panorama final do Sistema de Compras Online (SCO).

Figura 7. Opção 6 do Sistema de Compra Online.

```

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

    O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
6
Leilao encerrado
Nao foi dado lances para o produto 'cama'
Nao foi dado lances para o produto 'celular'
Marcia comprou o produto 'televisao' por 450.00

##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

    O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
4
Produto 1: celular
    Nao ha lance(s) para esse produto
    Listagem completa!

```

A Opção 7 (Ajuda) imprime em tela uma breve explicação sobre o funcionamento do Sistema de Compra Online. Na versão estática sequencial, esta opção faz acesso a um arquivo de texto (AjudaSCO.txt) que traz em tela as informações, como pode ser visto na Figura 8, e na versão dinâmica-encadeada é realizada através de uma função que se encontra na SCO_TAD2.c (**ajuda**), como pode ser visto na Figura 9.

Figura 8. Opção 7 do Sistema de Compra Online versão estática sequencial.

```
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

0 que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
7
    Ajuda sobre o Sistema de Compras Online (SCO).

Opcao (1) Cadastrar um produto:
    O usuario cadastrara um produto por vez especificando alguma(s) caracteristica(s) do mesmo, podendo realizar novamente outro cadastro. A cada cadastro a lista de produtos e atualizada.

Opcao (2) Remover um produto:
    O usuario removera um produto especifico por vez constado na lista de produtos, podendo remover outro produto novamente.

Opcao (3) Efetuar um lance:
    O usuario efetuara um lance por vez em um produto, podendo este efetivar novo lance no mesmo produto ou em outro de sua escolha.

Opcao (4) Listar produtos e lances:
    Sera(o) exibido(s) o(s) produto(s) disponivel(eis) em leilao seguido do(s) lance(s) e o(s) cliente(s) que o(s) efetivou(ram).

Opcao (5) Indicar outros produtos para lances:
    O usuario entrara com seu nome e lhe sera(o) apresentado(s) outro(s) produto(s) disponivel(is) a lance(s).

Opcao (6) Encerrar leilao:
    Verifica e imprime cada produto em leilao com o lance efetivado vencedor e o seu respectivo ganhador. Os demais produtos que nao receberam lance(s) tambem sao impressos, mostrando um panorama final do Sistema de Compras Online (SCO).

Opcao (7) Ajuda:
    Imprime na tela esta breve explicacao sobre o Sistema de Compras Online (SCO).

Opcao (8) Sair:
    Saida completa do Sistema de Compras Online (SCO).
```

Figura 9. Opção 7 do Sistema de Compra Online versão dinâmica e encadeada.

```
##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

0 que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
7
Opcao 1) Cadastrar um produto:
    O usuario cadastrara um produto por vez especificando alguma(s) caracteristica(s) do mesmo, podendo realizar novamente outro cadastro. A cada cadastro a lista de produtos e atualizada.

Opcao 2) Remover um produto:
    O usuario removera um produto especifico por vez constado na lista de produtos, podendo remover outro produto novamente.

Opcao 3) Efetuar um lance:
    O usuario efetuara um lance por vez em um produto, podendo este efetivar novo lance no mesmo produto ou em outro de sua escolha.

Opcao 4) Listar produtos e lances:
    Sera(o) exibido(s) o(s) produto(s) disponivel(eis) em leilao seguido do(s) lance(s) e o(s) cliente(s) que o(s) efetivou(ram).

Opcao 5) Indicar outros produtos para lances:
    O usuario entrara com seu nome e lhe sera(o) apresentado(s) outro(s) produto(s) disponivel(is) a lance(s).

Opcao 6) Encerrar leilao:
    Verifica e imprime cada produto em leilao com o lance efetivado vencedor e o seu respectivo ganhador. Os demais produtos que nao receberam lance(s) tambem sao impressos, mostrando um panorama final do Sistema de Compras Online (SCO).

Opcao 7) Ajuda:
    Imprime na tela esta breve explicacao sobre o Sistema de Compras Online (SCO).

Opcao 8) Sair:
    Saida completa do Sistema de Compras Online (SCO).
```

E finalmente a Opção 8 (Sair) realiza a saída do programa sistema de leilão. Uma mensagem final é impressa em tela, como pode ser visto na Figura 10 abaixo.

Figura 10. Opção 8 do Sistema de Compra Online versão dinâmica e encadeada.

```
##### Sistema de Compra Online (SCO) #####
Operacoes disponiveis no SCO:
1) Cadastrar um produto
2) Remover um produto
3) Efetuar um lance
4) Listar produtos e lances
5) Indicar outros produtos para lances
6) Encerrar leilao
7) Ajuda
8) Sair

      O que deseja fazer? (Escolha uma das opcoes acima entre 1 e 8)
8

      Obrigado por usar o nosso sistema!
```

4. Conclusão

O objetivo deste trabalho foi alcançado o qual consistiu no desenvolvimento de um Sistema de Compra Online (SCO) no estilo de leilão, em que vários usuários dão lances nos produtos disponíveis sendo vendidos. O dono do lance maior em um determinado momento leva o prêmio. Quando houver mais de um lance com o mesmo valor, quem deu o lance primeiro leva o prêmio. A implementação deste sistema foi realizada através das estruturas de Pilha, Fila e Lista respeitando-se suas funcionalidades em duas versões a dinâmica e encadeada e estática sequencial.

REFERÊNCIAS

ROBERTS, E. S. **Programming Abstractions in C: A Second Course in Computer Science**, Addison-Wesley; Edição 1, 819 pg, 1997.

KELLEY, AI, POHL, I. **A book on C: programming in C**, Addison-Wesley Professional; 4 edition, 752 pg, 1998.

BACKES, A. **Linguagem C : Completa e Descomplicada**, Elsevier; Edição: 1ª, 400 pg, 2012.