

**Pró-Reitoria Acadêmica
Curso de Ciência da Computação
Trabalho da Disciplina**

SISTEMA DE BILHETERIA DE CINEMA

Autores: Andreas Gomes Marchi
Francisco Natanael Ramos dos Santos
Gustavo Henrique Lemos Xavier
Rafael Silva Rezende

Orientador: Prof. Me. Ranyelson Neres de Carvalho

**ANDREAS GOMES MARCHI
FRANCISCO NATANAEL RAMOS DOS SANTOS
GUSTAVO HENRIQUE LEMOS XAVIER
RAFAEL SILVA REZENDE**

SISTEMA DE BILHETERIA DE CINEMA

Trabalho final apresentado ao Curso de graduação de Bacharelado em Ciência da Computação da Universidade Católica de Brasília, como requisito parcial para obtenção da aprovação na disciplina de Algoritmo e Programação Estruturada.

Orientador: Prof. Me. Ranyelson Neres de Carvalho

Brasília
2024

RESUMO DO PROGRAMA

O sistema é um programa em C para gerenciar a venda e reserva de bilhetes de um cinema denominado “HollowMovie”. Ele oferece funcionalidades para reservar assentos, cancelar bilhetes, imprimir informações sobre as reservas realizadas e gerar relatórios finais com informações financeiras. Cada filme possui um número fixo de assentos organizados em filas e colunas, simulando um cinema real, onde cada filme possui sua própria sala.

FLUXO DO PROGRAMA

1 Inicialização:

- O sistema é configurado com três filmes pré-definidos, cada um com uma matriz de assentos próprios, simulando uma sala de cinema real.
- Um menu interativo que permite ao usuário escolher entre reservar um bilhete, visualizar bilhetes reservados, cancelar uma reserva ou finalizar o programa com um relatório.

2 Reserva de Bilhete:

- O usuário escolhe o filme que deseja assistir.
- Preenche seus dados pessoais pedindo o nome, além de perguntar ao usuário se ele deseja a meia-entrada ou não.
- Escolhe um assento disponível na sala do filme selecionado.
- O sistema calcula o valor total a ser pago e exibe o ID do bilhete gerado, certificando que o usuário saiba o ID do seu ingresso.

3 Cancelamento de Bilhete:

- O programa pede o ID do ingresso ao usuário para cancelar o ingresso
- O sistema valida a existência do bilhete e, se encontrado, libera o assento correspondente e atualiza os valores totais arrecadados, além de devolver o valor ao usuário.

4 Impressão de Bilhetes Reservados:

- Exibe uma lista detalhada com todas as reservas realizadas, ou escolhendo o ingresso por ID, incluindo informações do cliente, filme, assento e valores pagos.

5 Relatório Final:

- Mostra o total de ingressos vendidos, combos adquiridos e o valor arrecadado. E, por fim, finalizando o programa.

6 DECLARAÇÃO

#include <stdio.h>, <stdlib.h>, <string.h>, <locale.h>:

Inclui todas as bibliotecas padrão utilizadas ao decorrer do programa, servem para manipulação de entrada/saída, memória, strings e configurações de localização.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>
```

#define FILMES, FILAS, COLUNAS, PREÇO_INGRESSO, PREÇO_COMBO:

Define todas as constantes usadas ao decorrer do programa para representar limites de filmes, assentos e preços de ingresso/combo.

```
#define FILMES 3
#define FILAS 5
#define COLUNAS 5
#define PREÇO_INGRESSO 20.00
#define PREÇO_COMBO 15.00
```

7 STRUCTS

Filme:

Representa um filme com nome e matriz de assentos (D = Disponível, X = Ocupado). Também é responsável por separar cada sala para cada filme.

```
typedef struct{
    char nome[50];
    char assentos[FILAS][COLUNAS];
} Filme;
```

Bilhete:

Contém dados do cliente, filme escolhido, assento, tipo de ingresso (meia ou inteira), combo, e preço total do bilhete.

```
typedef struct {
    char nomeCliente[80];
```

```

    int filmeID, fila, coluna, bilheteID, combo, meiaEntrada;
    double valorIngresso;
} Bilhete;

```

8 FUNÇÕES

Interface()

Exibe o menu principal com as opções disponíveis. Inclui o cabeçalho estético e as opções de ação para o usuário.

```

void Interface() {
    printf("-----\n");
    printf("                Bilheteria HollowMovie                \n");
    printf("-----\n\n");

    printf("Seja Bem-vindo ao melhor cinema de Hallownest!!!\n\n");
    printf("Escolha uma opcao da bilheteira:\n");
    printf("1 - Reservar lugar\n");
    printf("2 - Impressao do bilhete\n");
    printf("3 - Cancelar compra\n");
    printf("4 - Sair\n");
    printf("Escolha a opcao: ");
}

```

reservarBilhete()

Gerencia todo o processo de reserva, praticamente todas as funções do programa passam por essa função:

1. Exibe filmes (exibirFilmes).
2. Captura o filme escolhido e os dados do cliente (dadosCliente).
3. Define meia-entrada e exibe assentos disponíveis (MeiaEntrada, exibirAssentos).
4. Marca assento e calcula valor final (compra, combo).
5. Atualiza totais e incrementa o ID do bilhete.

```

void reservarBilhete(Bilhete *bilhetes, int *numeroCadastro, Filme filmes[FILMES],
    int *BilheteID, int *totalIngressos, int *totalCombos, double
    *totalArrecadado) {
    int op = 1;

    do {
        exibirFilmes(filmes);

        // Escolha do filme
        printf("\nEscolha o filme\n");
        printf("Digite o seu codigo (de 1 a %d): ", FILMES);
        scanf("%d", &bilhetes[*numeroCadastro].filmeID);
        bilhetes[*numeroCadastro].filmeID -= 1;

        // Captura os dados do cliente
        dadosCliente(&bilhetes[*numeroCadastro]);
        bilhetes[*numeroCadastro].meiaEntrada = MeiaEntrada();
        exibirAssentos(&filmes[bilhetes[*numeroCadastro].filmeID]);

        // Realiza a compra
        compra(&filmes[bilhetes[*numeroCadastro].filmeID],
            &bilhetes[*numeroCadastro]);
        bilhetes[*numeroCadastro].combo = combo();

        // Cálculo do valor final
        double valorIngresso = (bilhetes[*numeroCadastro].meiaEntrada ?
            PRECO_INGRESSO / 2 : PRECO_INGRESSO);
        bilhetes[*numeroCadastro].valorIngresso = valorIngresso +
            (bilhetes[*numeroCadastro].combo ? PRECO_COMBO : 0);

        // Exibir o valor final
        printf("\nValor Final a Ser Pago:\n");
        printf("Ingresso: R$ %.2f\n", valorIngresso);
        if (bilhetes[*numeroCadastro].combo) {
            printf("Combo (pipoca + refrigerante): R$ %.2f\n", PRECO_COMBO);
        }
        printf("Valor Total: R$ %.2f\n", bilhetes[*numeroCadastro].valorIngresso);

        // Gerar e exibir o ID do ingresso
        bilhetes[*numeroCadastro].bilheteID = ++(*BilheteID);
        printf("\nID do Ingresso: %d\n", bilhetes[*numeroCadastro].bilheteID);

        (*numeroCadastro)++;

        // Acumular os totais
    }
}

```

```

        (*totalIngressos)++;
        if (bilhetes[*numeroCadastro - 1].combo) {
            (*totalCombos)++;
        }
        (*totalArrecadado) += bilhetes[*numeroCadastro - 1].valorIngresso;

        printf("\nDeseja realizar outra venda (1 - Sim; 2 - Nao): ");
        scanf("%d", &op);
        if(op == 2){
            printf("Obrigado por escolher a HollowMovie!!!\n\n");
        }

    } while (op == 1);
}

```

dadosCliente()

Lê o nome do cliente para personalizar o bilhete.

```

void dadosCliente(Bilhete *bilhete) {
    printf("\nComo o cliente deseja ser chamado: ");
    scanf("%s", bilhete->nomeCliente);
}

```

MeiaEntrada()

Pergunta se o cliente deseja meia-entrada e retorna 1 (sim) ou 0 (não).

```

int MeiaEntrada() {
    int opcao;
    do {
        printf("\nCliente deseja meia entrada?\n");
        printf("1 - Sim; 2 - Nao: ");
        scanf("%d", &opcao);
        if (opcao == 1) {
            return 1;
        } else if (opcao == 2) {
            return 0;
        } else {
            printf("\nOpcao invalida. Digite novamente.\n");
        }
    } while (opcao < 1 || opcao > 2);
    return -1;
}

```

exibirFilmes()

Exibe os filmes em cartaz e seus códigos.

```
void exibirFilmes(Filme filmes[FILMES]) {
    printf("\nFilmes em cartaz\n\n");
    for (int i = 0; i < FILMES; i++) {
        printf("%d - %s\n", i + 1, filmes[i].nome);
    }
}
```

exibirAssentos()

Mostra os assentos do filme, indicando disponíveis (D) e ocupados (X).

```
void exibirAssentos(Filme *filme) {
    printf("\nAssentos Disponiveis (D para Disponivel, X para Ocupado):\n\n");

    printf("    ");
    for (int i = 0; i < COLUNAS; i++) {
        printf("%2d ", i + 1);
    }
    printf("\n    ");
    for (int i = 0; i < COLUNAS; i++) {
        printf("----");
    }
    printf("\n");

    for (int i = 0; i < FILAS; i++) {
        printf("%2d | ", i + 1);
        for (int j = 0; j < COLUNAS; j++) {
            printf(" %c ", filme->assentos[i][j]);
        }
        printf("\n");
    }
}
```


Compra()

Permite ao cliente escolher fila e coluna. Valida a disponibilidade e atualiza o status do assento.

```
void compra(Filme *filme, Bilhete *bilhete) {
    int fila, coluna;
    do {
        printf("\nEscolha uma fila (1 a %d): ", FILAS);
        scanf("%d", &fila);
        printf("Escolha uma coluna (1 a %d): ", COLUNAS);
        scanf("%d", &coluna);

        if (filme->assentos[fila - 1][coluna - 1] == 'X') {
            printf("Esse assento ja foi reservado. Escolha outro.\n");
        } else {
            filme->assentos[fila - 1][coluna - 1] = 'X';
            bilhete->fila = fila - 1;
            bilhete->coluna = coluna - 1;
            printf("Assento reservado com sucesso!\n");
            break;
        }
    } while (1);
}
```

Combo()

Pergunta ao cliente se deseja adicionar um combo e retorna 1 (sim) ou 0 (não).

```
int combo() {
    int opcao;
    printf("\nDeseja adicionar combo (pipoca + refrigerante) por R$ %.2f?\n",
PRECO_COMBO);
    printf("1 - Sim; 2 - Nao: ");
    scanf("%d", &opcao);
    return opcao == 1 ? 1 : 0;
}
```

imprimirBilhetesReservados()

Possui duas formas de impressão: imprimir todos os ingressos reservados, ou buscar o ingresso reservado pelo seu ID e mostrá-lo. Além de permitir realizar várias buscas de uma única vez.

```
void imprimirBilhetesReservados(Bilhete *bilhetes, int numeroCadastro,
Filme filmes[FILMES]) {
    int i, numeroBilhete = 0, op = 1, selecionar = 0;

    do{
        printf("Selecione as opcao abaixo!\n\n");
        printf("1 - Imprimir todos os bilhetes.\n");
        printf("2 - Buscar bilhete por ID.\n\n");
        printf("Digite a opcao: ");
        scanf("%d", &selecionar);

        switch(selecionar){
            case 1:
                if(numeroCadastro == 0){
                    printf("Sem bilhetes para pesquisa\n\n");
                }else{
                    for (i = 0; i < numeroCadastro; i++) {
                        Bilhete bilhete = bilhetes[i];
                        imprimirBilhete(bilhete, filmes);
                    }
                }
                break;

            case 2:
                if(numeroCadastro == 0){
                    printf("Sem bilhetes para pesquisa\n\n");
                }else{
                    do{
                        printf("Digite ID do bilhete para a pesquisa:
");

                        scanf("%d", &numeroBilhete);

                        for (i = 0; i < numeroCadastro; i++) {
                            Bilhete bilhete = bilhetes[i];

                            if(bilhete.bilheteID == numeroBilhete){
                                imprimirBilhete(bilhete, filmes);

                                break;
                            }
                        }
                    } while (1);
                }
            }
        }
    } while (op == 1);
}
```

```

        }else if(i == numeroCadastro-1){
            printf("Bilhete nao encontrado!\n\n");
        }
    }

    printf("Deseja pesquisar outro bilhete 1 - SIM e 2 - NAO: ");
    scanf("%d", &op);

    }while(op == 1);
}
break;

default:
    printf("Opcao nao encontrada!\n\n");
    break;
}

printf("Deseja fazer outra pesquisa: 1 - SIM e 2 - NAO: ");
scanf("%d", &op);

}while(op == 1);
}

```

imprimirBilhete()

Essa função exibe na tela todos os detalhes de um ingresso específico. Ela é usada pela função imprimirBilhetesReservados() para mostrar os dados dos ingressos.

```

void imprimirBilhete(Bilhete bilhete, Filme filmes[FILMES]){
    printf("\n----- Bilhetes Reservado ----- \n");
    printf("\nBilhete ID: %d\n", bilhete.bilheteID);
    printf("Cliente: %s\n", bilhete.nomeCliente);
    printf("Filme: %s\n", filmes[bilhete.filmeID].nome);
    printf("Assento: Fila %d, Coluna %d\n", bilhete.fila + 1,
bilhete.coluna + 1);
    printf("Meia Entrada: %s\n", bilhete.meiaEntrada ? "Sim" :
"Nao");
    printf("Combo: %s\n", bilhete.combo ? "Sim" : "Nao");
    printf("Valor Total: R$ %.2f\n", bilhete.valorIngresso);
}

```

```

        printf("-----\n\n");
    }
}

```

cancelarBilhete()

Permite cancelar um bilhete pelo ID. Atualiza a matriz de assentos, totais de ingressos/combos, e remove o bilhete da lista. No int main, está colocado na terceira opção do switch-case.

```

void cancelarBilhete(Bilhete *bilhetes, int *numeroCadastro, Filme filmes[FILMES],
int *totalIngressos, int *totalCombos, double *totalArrecadado) {
    int id, encontrado = 0;
    printf("\nInforme o ID do bilhete a ser cancelado: ");
    scanf("%d", &id);

    for (int i = 0; i < *numeroCadastro; i++) {
        if (bilhetes[i].bilheteID == id) {
            // Marcar o assento como disponível

            filmes[bilhetes[i].filmeID].assentos[bilhetes[i].fila][bilhetes[i].coluna] = 'D';

            // Atualizar os totais
            if (*totalIngressos > 0) {
                (*totalIngressos)--;
            }
            if (bilhetes[i].combo && *totalCombos > 0) {
                (*totalCombos)--;
            }
            if (*totalArrecadado >= bilhetes[i].valorIngresso) {
                (*totalArrecadado) -= bilhetes[i].valorIngresso;
            } else {
                printf("\nErro: valor arrecadado e insuficiente para subtrair\n");
            }

            // Remover o bilhete cancelado
            for (int j = i; j < *numeroCadastro - 1; j++) {
                bilhetes[j] = bilhetes[j + 1];
            }
            (*numeroCadastro)--;

            printf("\nBilhete cancelado com sucesso!\n");
            printf("O valor do bilhete foi devolvido!");
            encontrado = 1;
            break;
        }
    }
}

```

```

    }

}

if (!encontrado) {
    printf("\nBilhete nao encontrado!\n");
}
}

```

gerarRelatorioFinal()

Calcula e exibe um relatório com totais de ingressos vendidos, combos e valor arrecadado. Sendo a última opção do switch-case no int main.

```

void gerarRelatorioFinal(int totalIngressos, int totalCombos, double
totalArrecadado) {
    printf("\n----- Relatorio Final ----- \n");
    printf("Total de ingressos vendidos: %d\n", totalIngressos);
    printf("Total de combos vendidos: %d\n", totalCombos);
    printf("Total arrecadado: R$ %.2f\n", totalArrecadado);
    printf("\nObrigado por escolher a HollowMovie!!!\n");
    printf("Volte Sempre!\n");
}

```

9 MAIN

1. Configura o ambiente (setlocale) e inicializa filmes e bilhetes.
2. Preenche matriz de assentos com D, mostrando que os assentos estão disponíveis.
3. Exibe menu e executa ações com base na entrada do usuário (switch).