

UNIP EaD

Projeto Integrado Multidisciplinar

CST em Análise e Desenvolvimento de Sistemas

Sistema em Linguagem C para venda de ingressos de teatro

Recursos tecnológicos como estratégia de transmissão de informação

2019

UNIP EaD

Projeto Integrado Multidisciplinar

CST em Análise e Desenvolvimento de Sistemas

Sistema em Linguagem C para venda de ingressos de teatro

Recursos tecnológicos como estratégia de transmissão de informação

Nome: André Luiz Souza Kiwi – RA: 0506459

Nome: Daniel Buck Silva – RA: 0537043

Nome: Samara Ibrahim Lara – RA: 1953505

Curso: Análise e Desenvolvimento de Sistemas

Semestre: Segundo

2019

GUIA de normalização para apresentação de trabalhos acadêmicos da
Universidade Paulista / Biblioteca Universidade Paulista, UNIP. /
Revisada e atualizada pelas bibliotecárias Alice Horiuchi e Bruna Orgler
Schiavi. – 2012.
51 p. : il. color.

1. NORMALIZAÇÃO. 2. TRABALHOS ACADÊMICOS. 3. ABNT. I.
Biblioteca Universidade Paulista.

UNIP EaD

Projeto Integrado Multidisciplinar

CST em Análise e Desenvolvimento de Sistemas

Sistema em Linguagem C para venda de ingressos de teatro

Recursos tecnológicos como estratégia de transmissão de informação

Aprovado em:

BANCA EXAMINADORA

_____/____/____

Prof. Nome do Professor
Universidade Paulista – UNIP

_____/____/____

Prof. Nome do Professor
Universidade Paulista – UNIP

_____/____/____

Prof. Nome do Professor

Universidade Paulista UNIP

“A vida é uma peça de teatro que não permite ensaios. Por isso, cante, chore, dance, ria e viva intensamente, antes que a cortina se feche e a peça termine sem aplausos.”

(Charles Chaplin)

Resumo

O presente Projeto Integrado Multidisciplinar consiste na elaboração um sistema em linguagem C para venda de ingressos de teatro.

O sistema de venda de ingressos de teatro deverá levar em consideração uma tabela de valores para os ingressos, a qual deverá contemplar meia-entrada (50% de desconto) para estudantes, crianças de 02 a 12 anos, adultos a partir de 60 anos e professores da rede pública de ensino.

Ao concretizar a compra do ingresso, o sistema deverá emitir (mostrar na tela) um ticket com a data, a hora, o nome da peça teatral. Nenhum espectador pode ter um ingresso com o mesmo número da poltrona de outro espectador da mesma peça teatral de um determinado horário.

O sistema deverá informar quando não existir mais lugares disponíveis e, conseqüentemente, não permitir mais a venda de ingressos para a peça em questão. Deverá também fazer a gestão do caixa, isto é, informar no fechamento do caixa todas as movimentações do dia e o saldo do fechamento.

Palavras-chave: Teatro, Sistema, Ingressos.

Abstract

This Multidisciplinary Integrated Project consists of the elaboration of a C language system for the sale of theater tickets.

The theater ticket sales system should take into consideration a ticket price table, which should include half price (50% discount) for students, children from 2 to 12 years old, adults from 60 years old and over. Public school teachers.

Upon completion of the ticket purchase, the system should issue (show on screen) a ticket with the date, time, name of the play. No spectator may have a ticket with the same number as another spectator's seat in the same play at a particular time.

The system should inform you when no more seats are available and therefore no longer allow tickets to be sold for the piece in question. You should also manage the cashier, that is, inform at the cashier closing all movements of the day and the closing balance.

Keywords: Theater, System, Tickets.

Lista de Ilustrações

Imagem 1 – Modelo Cascata (Clássico)	13
Imagem 2 – Casos de Uso	14

Lista de abreviaturas e siglas

PBT – Polibutileno Tereftalato

PA – Poliamida

NiMH – Níquel-Metal Hydride

CPU – Central Processing Unit

HD – Hard Disc

DDR – Double Data Rate

RAM – Random Access Memory

Sumário

Introdução	11
Desenvolvimento	12
1. Engenharia de Software em Projetos	12
2. Ciclo de Vida do Sistema – Modelo Cascata (ou Clássico)	12
3. Casos de Uso	13
4. Lógica de Programação	14
5. Codificação	15
6. Como o Sistema Funciona	26
7. Implantação de Software	29
8. Manutenção se Necessário	30
Conclusão	31
Referência	32

Introdução

Com a proposta do projeto, foi dividido as partes de cada desenvolvedor, para realização do sistema, o modelo em cascata é modelo que mais abrange valor ao projeto, pois analisamos e consultamos mais a proposta do mesmo, para uma realização de mão única sempre pensando e documentando os erros, que desse uma entrega satisfatória e não haver retrabalho.

Em uma reunião com idealizadores foi proposto um projeto de sistema para a realização de cadastro e venda de ingresso para um teatro que seja simples e responsivo, onde o usuário realiza o cadastro, faz a escolha da peça que gostaria de assistir o horário escolhesse o assento, imprima seu cupom e finaliza a compra. Visando também o gestor do negócio foi possível implementar um sistema de caixa que fizesse a contabilidade de venda dos ingressos.

Nesse projeto utilizamos o DEV++ para desenvolvimento do sistema, com o conhecimento de Linguagem e técnica de programação fizemos na linguagem C utilizamos endentação em árvore, e na matéria de Engenharia de software utilizamos o modelo em Cascata como dito acima.

Desenvolvimento

1. Engenharia de Software em Projetos

Para realização de um projeto de sistemas é preciso uma equipe de desenvolvedores, planejamento, ferramentas, para que seja atendido os requisitos da proposta, e isso é necessário que todos estejam reunidos para uma boa execução e resolver problemas.

O projeto proposto de um sistema de atendimento ao cliente que faça cadastro, desenvolvido para que o usuário preencha seus dados sem interferência de outras opções, nas vendas é realizada a escolha de opções de peças teatrais horários e valor dos bilhetes e impressão do ingresso, o sistema funciona de forma interativa para atender os requisitos dos usuários e do negócio, no caso o fechamento de caixa.

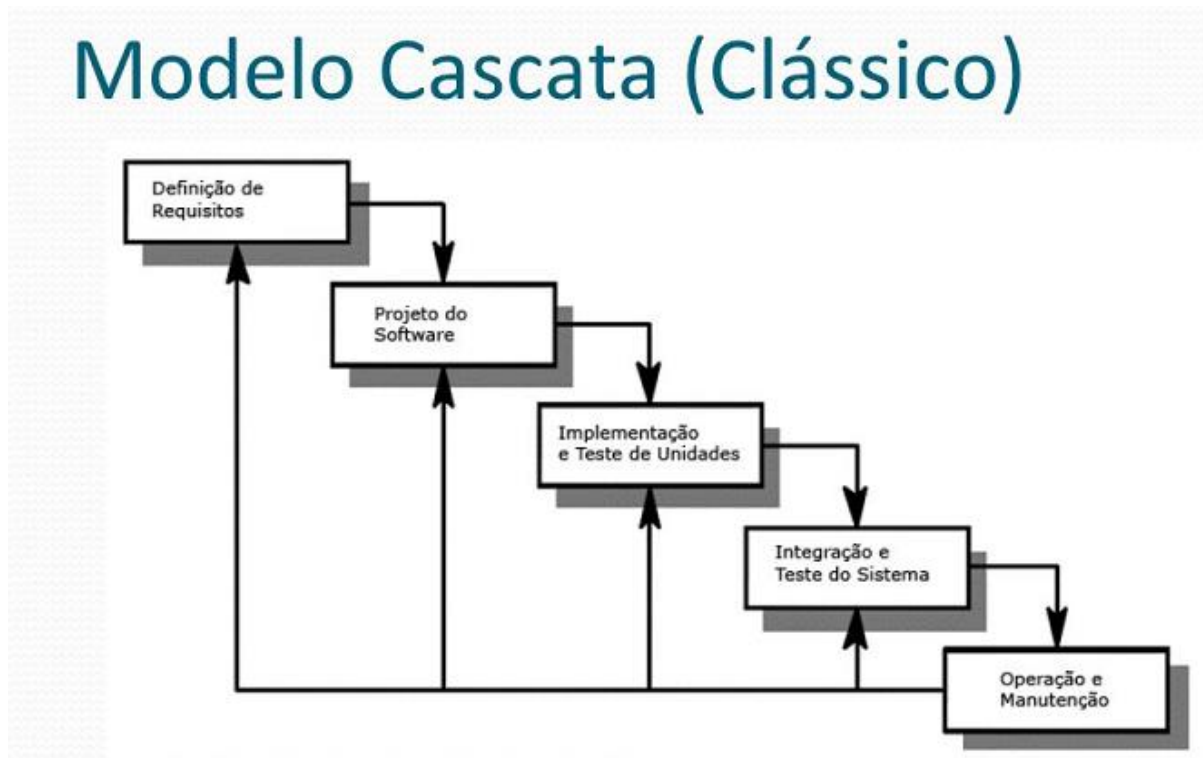
Entre várias reuniões, foram feitas várias propostas de implementação, iniciando com o ciclo codifica-remenda, que é método obsoleto para a atualidade pois não tem um planejamento organizado e todos fazem de tudo um pouco, mas aos poucos o ciclo de vida iria amadurecendo e ganhando outras proporções, e as reuniões do grupo foram ficando mais constantes até chegar no modelo de ciclo de vida de vida em cascata.

2. Ciclo de vida do Sistema – Modelo Cascata (ou Clássico)

Esse método é conhecido por ser desenvolvido em sequência sempre para frente, por meio de fases, com esse método passamos a realizar testes pensando no usuário, a cada erro foi documentado para que houvesse mudanças necessárias e o código do sistema funcionar corretamente.

A ideia é de constante progresso, vendo a proposta, fazendo o código, compilando, prestando atenção aos erros, documentamos, e anotamos os erros para que realizássemos a solução mais rapidamente, fazendo assim uma entrega eficiente.

Imagem 1 – Modelo Cascata (Clássico)



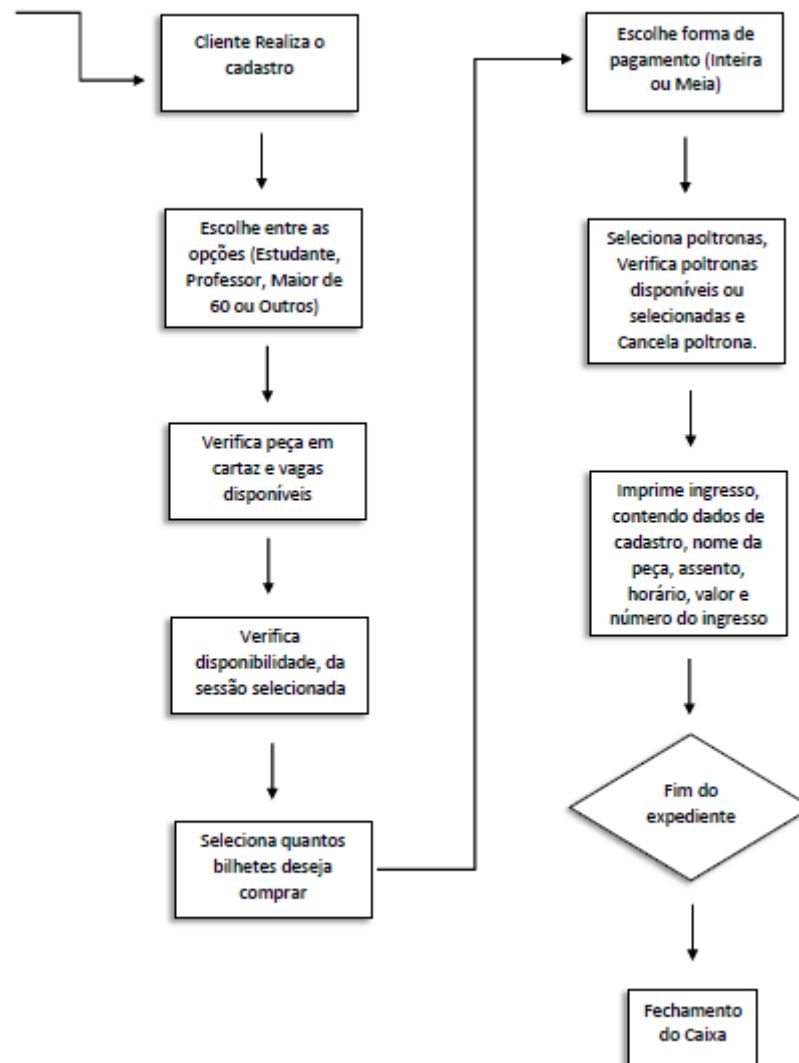
Fonte: Próprio autor (André Luiz Souza Wiki)

3. Casos de Uso

O diagrama é usado para orientar o que deve ser feito pelo sistema, de forma a promover a interação entre os atores e o cenário utilizado pelo sistema, definindo cada caso de uso, tornando o projeto de fácil entendimento. Os integrantes e os atores são agentes externos de um elemento de um diagrama de casos de uso, que apresenta cada funcionalidade do sistema, assim a implementação dos requisitos funcionais do sistema identificará os seus respectivos atores, indicando a ligação com o determinado caso de uso.

Dessa forma o cliente não terá contato algum com o software, apenas fará a compra do ingresso para que funcione dessa forma o usuário atendente será o responsável pela alimentação das solicitações no sistema, conforme ilustração abaixo.

Imagem 2 – Casos de Uso



Fonte: Própria autora (Samara Lara)

4. Lógica de Programação

Para atingir os objetivos no desenvolvimento de algoritmos, utilizamos as técnicas da lógica de programação, para não provocar ambiguidade na sequência das instruções que são executadas até que determinada condição se verifique, matematicamente, e se constitua um conjunto de processos e símbolos que os representem para efetuar um cálculo. Representaremos a seguir o fluxograma que demonstra a

lógica do sistema, pois a representação gráfica do algoritmo por desenhos geométricos, seguindo uma lógica sequencial, coesa, permite melhor entendimento.

5. Codificação

```
1: #include<stdio.h>
2: #include<stdlib.h>
3: #include<unistd.h>
4: #include<string.h>
5:
6: int ingresso, peca, horario = 0, pagar;
7: int cpf, tipoPessoa, b, idade, numero;
8: char estudante, professor, nome, idoso, outro;
9: char horarioDescricao[100];
10: char pecaDescricao[100];
11: char valorDescricao[100];
12:
13: int menu,
14: iterator,
15: poltronaOptionPicked,
16: pickedSeat,
17: cancelar,
18: max_poltronas,
19: position,
20: ocupadas,
21: mainMenuOptionPicked,
22: numeroPoltrona;
23:
24: int availableSeats = 20;
25:
26: int poltronaWasDeleted = 0;
27: int poltronaWasSelected = 0;
28:
29: int containerPoltrona[20] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
30: int qtyPecas;
31: int qtyIngressos;
32:
33: int pagarTotal;
34: int valorIngresso;
35:
36: void mainMenu()
37: {
38: printf("\n\n");
39: printf("\n=== BEM VINDO AO TEATRO DOS SONHOS ===\n");
40: printf("\n\n");
41: printf("*****\n");
42: printf("[1] - Fazer cadastro\n");
43: printf("[2] - Comprar Ingressos\n");
44: printf("[3] - Escolher poltrona\n");
45: printf("[4] - Imprimir Ingresso\n");
46: printf("[5] - Faturamento\n");
```

```

47: printf("[6] - Encerrar o Sistema\n");
48: printf("*****\n");
49: printf("\nSelecione uma opcao: ");
50: scanf("%d", &mainMenuOptionPicked);
51:
52: }
53: //FUNCTIONS
54: void fazerCadastroMenu()
55: {
56: system("cls");
57: printf("\n\n");
58: printf("\n=== BEM VINDO AO TEATRO DOS SONHOS ===\n");
59: printf("\n\n");
60: printf("*** DADOS DO CLIENTE ***\n\n");
61: printf("Nome: ");
62: scanf("%s", &nome);
63: printf("Idade: ");
64: scanf("%i", &idade);
65: printf("CPF: ");
66: scanf("%i", &cpf);
67: printf("\nESCOLHA UMA DAS OPCOES:");
68: printf("\n[1] - Estudante \n[2] - Professor \n[3] - Maior de 60 anos \n[4] -
Outros\n");
69: scanf("%d", &tipoPessoa);
70:
71: if(tipoPessoa == 1)
72: {
73: printf("Seu desconto de ESTUDANTE foi aprovado\n%c", estudante);
74: }
75: else if(tipoPessoa == 2)
76: {
77: printf("Seu desconto de PROFESSOR foi aprovado\n%c", professor);
78: }
79: else if(tipoPessoa == 3)
80: {
81: printf("Seu desconto de IDOSO foi aprovado\n%c", idoso);
82: }
83: else if(tipoPessoa == 4)
84: {
85: printf("Sem desconto se for inteira\n%c", outro);
86: }
87: }
88:
89: void compraIngressoMenu()
90: {
91: system("cls");
92: printf("\n\n");
93: printf("\n=== BEM VINDO AO TEATRO DOS SONHOS ===\n");
94: printf("\n\n");
95: printf("*** VENDA DE INGRESSOS ***\n\n");
96: printf("Escolha uma das pecas em cartaz:\n\n");
97: printf("[1] Peter Pan - O musical\n[2] Como respirar em dias de chuva\n\n");
98: printf("\n*** ATENCAO ***\nSomente %d lugares disponiveis!\n\n",
availableSeats);
99: printf("Digite [0] para sair\n\n");
100: scanf("%i",&peca);
101:
102: if(peca == 1)
103: {

```



```

104: printf("Peca 1: Peter Pan - O musical\n\n");
105: strcpy(pecaDescricao, "Peter Pan - O musical");
106: printf("Datas e Sessoes\n\n");
107: printf("[1] Terca-feira as 12:00\n");
108: printf("[2] Quinta-feira as 13:00\n");
109: printf("[3] Sabado as 14:00\n");
110: printf("\nComprar bilhete para qual data e sessao? ");
111: scanf("%i", &horario);
112: qtyPecas = +1;
113:
114: if(horario == 1)
115: {
116: strcpy(horarioDescricao, "Terca-feira as 12:00");
117: printf("\nInforme a quantidade de bilhetes que deseja comprar: ");
118: scanf("%i", &ingresso);
119:
120:
121: if(ingresso <= 20)
122: {
123: printf("-----\n");
124: printf("Escolha o tipo de ingresso:\n\n");
125: printf("[1] Inteira: R$ 20.00\n");
126: printf("[2] Meia: R$ 10.00\n");
127: scanf("%i", &pagar);
128: printf("-----\n");
129:
130: if(pagar == 1)
131: {
132: valorIngresso = 20;
133: printf("Forma de Pagamento: Inteira\n\n");
134: printf("Data e sessao escolhidas: Terca-feira as 12:00 - Peter
Pan - O musical\n\n");
135: printf("Seu bilhete foi adquirido com sucesso\n");
136: printf("Tenha um otimo espetaculo!\n");
137: strcpy(valorDescricao, "Inteira: R$ 20.00");
138: }
139: else if(pagar == 2)
140: {
141: valorIngresso = 10;
142: printf("Forma de Pagamento: Meia\n\n");
143: printf("Data e sessao escolhidas: Terca-feira as 12:00 - Peter
Pan - O musical\n\n");
144: printf("Seu bilhete foi adquirido com sucesso\n");
145: printf("Tenha um otimo espetaculo!\n");
146: strcpy(valorDescricao, "Meia: R$ 10.00");
147: }
148: }
149: else
150: {
151: printf("Nao ha mais bilhetes disponiveis para esta peca.\n");
152: }
153: }
154: else
155: if(horario == 2)
156: {
157: strcpy(horarioDescricao, "Quinta-feira as 13:00");
158: printf("\nInforme a quantidade de bilhetes que deseja comprar: ");
159: scanf("%i", &ingresso);
160:

```

```

161: if(ingresso <= 20)
162: {
163: printf("-----\n");
164: printf("Escolha o tipo de ingresso:\n\n");
165: printf("[1] Inteira: R$ 20.00\n");
166: printf("[2] Meia: R$ 10.00\n");
167: scanf("%i", &pagar);
168: printf("-----\n");
169:
170: if(pagar == 1)
171: {
172: valorIngresso = 20;
173: printf("Forma de Pagamento: Inteira\n\n");
174: printf("Data e sessao escolhidas: Quinta-feira as 13:00 -
Peter Pan - O musical\n\n");
175: printf("Seu bilhete foi adquirido com sucesso\n");
176: printf("Tenha um otimo espetaculo!\n");
177: strcpy(valorDescricao, "Inteira: R$ 20.00");
178: }
179: else if(pagar == 2)
180: {
181: valorIngresso = 10;
182: printf("Forma de Pagamento: Meia\n\n");
183: printf("Data e sessao escolhidas: Quinta-feira as 13:00 -
Peter Pan - O musical\n\n");
184: printf("Seu bilhete foi adquirido com sucesso\n");
185: printf("Tenha um otimo espetaculo!\n");
186: strcpy(valorDescricao, "Meia: R$ 10.00");
187: }
188: }
189: else
190: {
191: printf("Nao ha mais bilhetes disponiveis para esta peca.\n");
192: }
193: }
194: else
195: if(horario == 3)
196: {
197: strcpy(horarioDescricao, "Sabado as 14:00");
198: printf("\nInforme a quantidade de bilhetes que deseja comprar: ");
199: scanf("%i", &ingresso);
200:
201: if(ingresso <= 20)
202: {
203: printf("-----\n");
204: printf("Escolha o tipo de ingresso:\n\n");
205: printf("[1] Inteira: R$ 20.00\n");
206: printf("[2] Meia: R$ 10.00\n");
207: scanf("%i", &pagar);
208: printf("-----\n");
209:
210: if(pagar == 1)
211: {
212: valorIngresso = 20;
213: printf("Forma de Pagamento: Inteira\n\n");
214: printf("Data e sessao escolhidas: Sabado as 14:00 - Peter
Pan - O musical\n\n");
215: printf("Seu bilhete foi adquirido com sucesso\n");
216: printf("Tenha um otimo espetaculo!\n");

```

```

217: strcpy(valorDescricao, "Inteira: R$ 20.00");
218:
219: }
220: else if(pagar == 2)
221: {
222: valorIngresso = 10;
223: printf("Forma de Pagamento: Meia\n\n");
224: printf("Data e sessao escolhidas: Sabado as 14:00 - Peter
Pan - O musical\n\n");
225: printf("Seu bilhete foi adquirido com sucesso\n");
226: printf("Tenha um otimo espetaculo!\n");
227: strcpy(valorDescricao, "Meia: R$ 10.00");
228: }
229: }
230: else
231: {
232: printf("Nao ha mais bilhetes disponiveis para esta peca.\n");
233: }
234: }
235:
236: }
237: else
238: if(peca == 2)
239: {
240: printf("Peca 2: Como respirar em dias de chuva\n\n");
241: strcpy(pecaDescricao, "Como respirar em dias de chuva");
242: printf("Datas e Sessoes\n\n");
243: printf("[1] Terca-feira as 12:00\n");
244: printf("[2] Quinta-feira as 13:00\n");
245: printf("[3] Sabado as 14:00\n");
246: printf("\nComprar bilhete para qual data e sessao? ");
247: scanf("%i", &horario);
248:
249: if(horario == 1)
250: {
251: strcpy(horarioDescricao, "Terca-feira as 12:00");
252: printf("\nInforme a quantidade de bilhetes que deseja comprar: \n");
253: scanf("%i", &ingresso);
254:
255: if(ingresso <= 20)
256: {
257: printf("-----\n");
258: printf("Escolha o tipo de ingresso:\n\n");
259: printf("[1] Inteira: R$ 20.00\n");
260: printf("[2] Meia: R$ 10.00\n");
261: scanf("%i", &pagar);
262: printf("-----\n");
263:
264: if(pagar == 1)
265: {
266: valorIngresso = 20;
267: printf("Forma de Pagamento: Inteira\n\n");
268: printf("Data e sessao escolhidas: Terca-feira as 12:00 - Como
respirar em dias de chuva\n\n");
269: printf("Seu bilhete foi adquirido com sucesso\n");
270: printf("Tenha um otimo espetaculo!\n");
271: }
272: else if(pagar == 2)
273: {

```

```

274: valorIngresso = 10;
275: printf("Forma de Pagamento: Meia\n\n");
276: printf("Data e sessao escolhidas: Terca-feira as 12:00 - Como
respirar em dias de chuva\n\n");
277: printf("Seu bilhete foi adquirido com sucesso\n");
278: printf("Tenha um otimo espetaculo!\n");
279: }
280: }
281: else
282: {
283: printf("Nao ha mais bilhetes disponiveis para esta peca.\n");
284: }
285: }
286: if(horario == 2)
287: {
288: strcpy(horarioDescricao, "Quinta-feira as 13:00");
289: printf("\nInforme a quantidade de bilhetes que deseja comprar: \n");
290: scanf("%i", &ingresso);
291:
292: if(ingresso <= 20)
293: {
294: printf("-----\n");
295: printf("Escolha o tipo de ingresso:\n\n");
296: printf("[1] Inteira: R$ 20.00\n");
297: printf("[2] Meia: R$ 10.00\n");
298: scanf("%i", &pagar);
299: printf("-----\n");
300:
301: if(pagar == 1)
302: {
303: valorIngresso = 20;
304: printf("Forma de Pagamento: Inteira\n\n");
305: printf("Data e sessao escolhidas: Quinta-feira as 13:00 - Como
respirar em dias de chuva\n\n");
306: printf("Seu bilhete foi adquirido com sucesso\n");
307: printf("Tenha um otimo espetaculo!\n");
308: }
309: else if(pagar == 2)
310: {
311: valorIngresso = 10;
312: printf("Forma de Pagamento: Meia\n\n");
313: printf("Data e sessao escolhidas: Quinta-feira as 13:00 - Como
respirar em dias de chuva\n\n");
314: printf("Seu bilhete foi adquirido com sucesso\n");
315: printf("Tenha um otimo espetaculo!\n");
316: }
317: }
318: else
319: {
320: printf("Nao ha mais bilhetes disponiveis para esta peca.\n");
321: }
322: }
323: if(horario == 3)
324: {
325: strcpy(horarioDescricao, "Sabado as 14:00");
326: printf("\nInforme a quantidade de bilhetes que deseja comprar: \n");
327: scanf("%i", &ingresso);
328:
329: if(ingresso <= 20)

```

```

330: {
331: printf("-----\n");
332: printf("Escolha o tipo de ingresso:\n\n");
333: printf("[1] Inteira: R$ 20.00\n");
334: printf("[2] Meia: R$ 10.00\n");
335: scanf("%i", &pagar);
336: printf("-----\n");
337:
338: if(pagar == 1)
339: {
340: valorIngresso = 20;
341: printf("Forma de Pagamento: Inteira\n\n");
342: printf("Data e sessao escolhidas: Sabado as 14:00 - Como
respirar em dias de chuva\n\n");
343: printf("Seu bilhete foi adquirido com sucesso\n");
344: printf("Tenha um otimo espetaculo!\n");
345: }
346: else if(pagar == 2)
347: {
348: valorIngresso = 10;
349: printf("Forma de Pagamento: Meia\n\n");
350: printf("Data e sessao escolhidas: Sabado as 14:00 - Como
respirar em dias de chuva\n\n");
351: printf("Seu bilhete foi adquirido com sucesso\n");
352: printf("Tenha um otimo espetaculo!\n");
353: }
354: }
355: else
356: {
357: printf("Nao ha mais bilhetes disponiveis para esta peca.\n");
358: }
359: }
360: }
361: else
362: if(peca == 0)
363: {
364: system("cls");
365: }
366: }
367:
368: void escolherPoltrona()
369: {
370: while (poltronaOptionPicked != 5)
371: {
372: system("cls");
373: printf("\n\n");
374: printf("\n=== BEM VINDO AO TEATRO DOS SONHOS ===\n");
375: printf("\n\n");
376: printf("*** ESCOLHA DE POLTRONAS ***\n\n");
377: printf("*****\n");
378: printf("[1] Selecionar poltrona\n");
379: printf("[2] Poltronas disponiveis\n");
380: printf("[3] Poltronas selecionadas\n");
381: printf("[4] Cancelar poltrona\n");
382: printf("[5] Sair\n");
383: printf("*****\n");
384: printf("\nSelecione uma opcao: ");
385: scanf("%d", & poltronaOptionPicked);
386:

```

```

387: switch(poltronaOptionPicked)
388: {
389: // CASE 1: SELECIONAR POLTRONA
390: case 1:
391: printf("Digite o numero da poltrona desejada: ");
392: scanf("%d", &pickedSeat);
393: while (checkSeatNumber(pickedSeat) == 0)
394: {
395: printf("Digite o numero da poltrona desejada: ");
396: scanf("%d", &pickedSeat);
397: }
398: containerPoltrona[pickedSeat - 1] = pickedSeat;
399: printf("\nPoltrona %d reservada!\n", pickedSeat);
400: printf("\nPressione qualquer tecla para voltar ao menu
principal.\n");
401: getch();
402: break;
403:
404: // CASE 2: VISUALIZAR POLTRONAS DISPONIVEIS
405: case 2:
406: printf("Poltronas disponiveis:\n\n");
407: for(iterator =0 ; iterator <20; iterator ++)
408: {
409: if (containerPoltrona[iterator] == 0)
410: {
411: printf("\tPoltrona: %d \n",iterator + 1);
412: }
413: }
414: printf("\nPressione qualquer tecla para voltar ao menu
principal.\n");
415: getch();
416: break;
417:
418: // CASE 3: VISUALIZAR POLTRONAS SELECIONADAS
419: case 3:
420: printf("Poltronas selecionadas:\n\n");
421: for(iterator =0 ; iterator <20; iterator ++)
422: {
423: if (containerPoltrona[iterator] != 0)
424: {
425: printf("\tPoltrona: %d \n",iterator + 1);
426: poltronaWasSelected = 1;
427: }
428: }
429: if (poltronaWasSelected == 0)
430: {
431: printf("Nenhuma poltrona foi selecionada ate o momento.\n");
432: }
433: printf("\nPressione qualquer tecla para voltar ao menu
principal.\n");
434: getch();
435: break;
436:
437: // CASE 4: CANCELAR POLTRONA (UMA OU TODAS)
438: case 4:
439: printf("[1] Cancelar uma poltrona\n");
440: printf("[2] Cancelar todas as poltronas\n");
441: printf("\nSelecione uma opcao: ");
442: scanf("%d", &cancelar);

```

```

443: if(cancelar == 1)
444: {
445: printf("Digite o numero da poltrona para cancelar: ");
446: scanf("%d", &numeroPoltrona);
447: for(iterator = 0; iterator < 20; iterator++)
448: {
449: if (containerPoltrona[iterator] == numeroPoltrona)
450: {
451: containerPoltrona[numeroPoltrona - 1] = 0;
452: poltronaWasDeleted = 1;
453: printf("Poltrona %d cancelada com sucesso.\n",
numeroPoltrona);
454: printf("\nPressione qualquer tecla para voltar ao menu
principal.\n");
455: getch();
456: }
457: }
458: if (poltronaWasDeleted == 0)
459: {
460: printf("Nao ha nenhuma poltrona %d para efetuar
cancelamento.\n", numeroPoltrona);
461: printf("\nPressione qualquer tecla para voltar ao menu
principal.\n");
462: getch();
463: }
464: }
465: else if (cancelar == 2)
466: {
467: for(iterator = 0; iterator < 20; iterator++)
468: {
469: if (containerPoltrona[iterator] != 0)
470: {
471: printf("Poltrona %d cancelada com sucesso.\n",
numeroPoltrona);
472: containerPoltrona[iterator] = 0;
473: poltronaWasDeleted = 1;
474: }
475: }
476: if (poltronaWasDeleted == 0)
477: {
478: printf("Nao ha poltronas para efetuar cancelamento.\n");
479: }
480: printf("\nPressione qualquer tecla para voltar ao menu
principal.\n");
481: getch();
482: }
483: else
484: {
485: printf("Opcao incorreta.\n");
486: printf("\nPressione qualquer tecla para voltar ao menu
principal.\n");
487: getch();
488: }
489: break;
490:
491: // CASE 5: RETORNAR AO MENU PRINCIPAL
492: case 5:
493: break;
494: default:

```

```

495: printf("Opcao incorreta\n");
496: printf("\nPressione qualquer tecla para voltar ao menu
principal.\n");
497: getch();
498: break;
499: }
500: }
501:
502: poltronaOptionPicked = 0;
503: }
504:
505: void imprimirIngresso()
506: {
507:
508: system("cls");
509: printf("\n\n");
510: printf("\n=== BEM VINDO AO TEATRO DOS SONHOS ===\n");
511: printf("\n\n");
512: printf("-----\n");
513: printf("IMPRIMINDO INGRESSO\n");
514: printf("-----\n");
515:
516: printf("Nome: %d \n ", nome);
517: printf("Idade: %d \n", idade);
518: printf("CPF: %d \n", cpf);
519:
520: printf("\n Peca: %d %s \n ",peca,pecaDescricao);
521:
522: for(iterator =0 ; iterator <20; iterator ++ )
523: {
524: if (containerPoltrona[iterator] != 0)
525: {
526: printf("\n Assento: %d \n",iterator + 1);
527: poltronaWasSelected = 1;
528: }
529: }
530: printf("\n Ingresso: %d",ingresso);
531: printf("\n Horario: %d %s",horario,horarioDescricao);
532: printf("\n Valor: %d %s",pagar,valorDescricao);
533: getch();
534: }
535:
536: void faturamento()
537: {
538: system("cls");
539: pagarTotal = ingresso * valorIngresso;
540: printf("\n\n");
541: printf("\n=== BEM VINDO AO TEATRO DOS SONHOS ===\n");
542: printf("FATURAMENTO\n");
543: printf("\nTotal recebido no dia: R$ %d", pagarTotal);
544: }
545: int checkSeatNumber()
546: {
547: int ret = 1;
548: if(pickedSeat < 1 || pickedSeat > 20)
549: {
550: printf("Numero de poltrona invalido\n");
551: ret = 0;
552: }

```



```

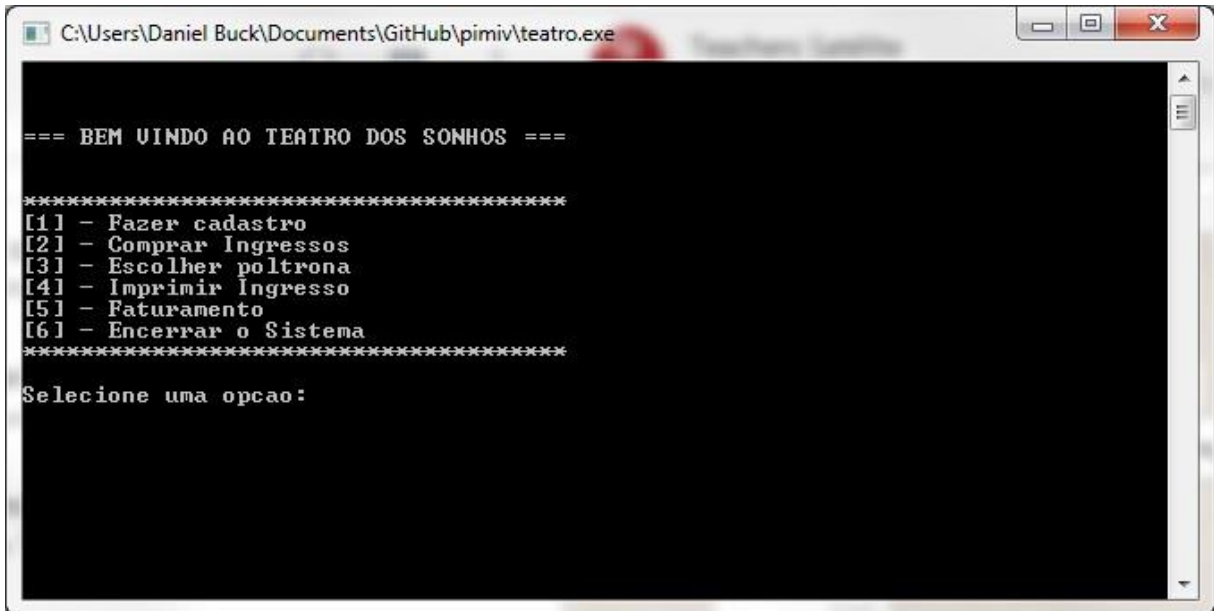
553: for(iterator = 0 ; iterator < 20; iterator ++)
554: {
555: if (containerPoltrona[iterator] == pickedSeat)
556: {
557: printf("Numero de poltrona indisponivel.\n");
558: ret = 0;
559: }
560: }
561: return ret;
562: }
563:
564:
565: int main (int argc, char *argv[])
566: {
567: while(mainMenuOptionPicked != 6)
568: {
569: mainMenu();
570:
571: switch(mainMenuOptionPicked)
572: {
573: case 1://Fazer cadastro
574: fazerCadastroMenu();
575: break;
576: case 2://Comprar Ingressos
577: compraIngressoMenu();
578: break;
579: case 3://Escolher poltrona
580: escolherPoltrona();
581: break;
582: case 4://Imprimir Ingresso
583: imprimirIngresso();
584: break;
585: case 5: //FATURAMENTO
586: faturamento();
587: break;
588: case 6://ENCERRAMENTO DO SISTEMA
589: system("exit");
590: printf("\nFim da Sessao\n");
591: break;
592:
593: default:
594: printf("* OPCAO INVALIDA *\n");
595: system("pause");
596: break;
597:
598: }
599: }
600: system("pause");
601: return 0;
602: }
603:

```

6. Como o Sistema Funciona

- **Menu**

Ao iniciar o sistema existe 6 opções, o menu da tela contém as opções onde o usuário pode clicar a opção que pretende interagir.

A screenshot of a Windows application window titled 'C:\Users\Daniel Buck\Documents\GitHub\pimiv\teatro.exe'. The window has a black background with white text. At the top, it says '=== BEM VINDO AO TEATRO DOS SONHOS ==='. Below this, a list of options is displayed, separated by asterisks: [1] - Fazer cadastro, [2] - Comprar Ingressos, [3] - Escolher poltrona, [4] - Imprimir Ingresso, [5] - Faturamento, and [6] - Encerrar o Sistema. At the bottom, it prompts 'Selecione uma opcao:'.

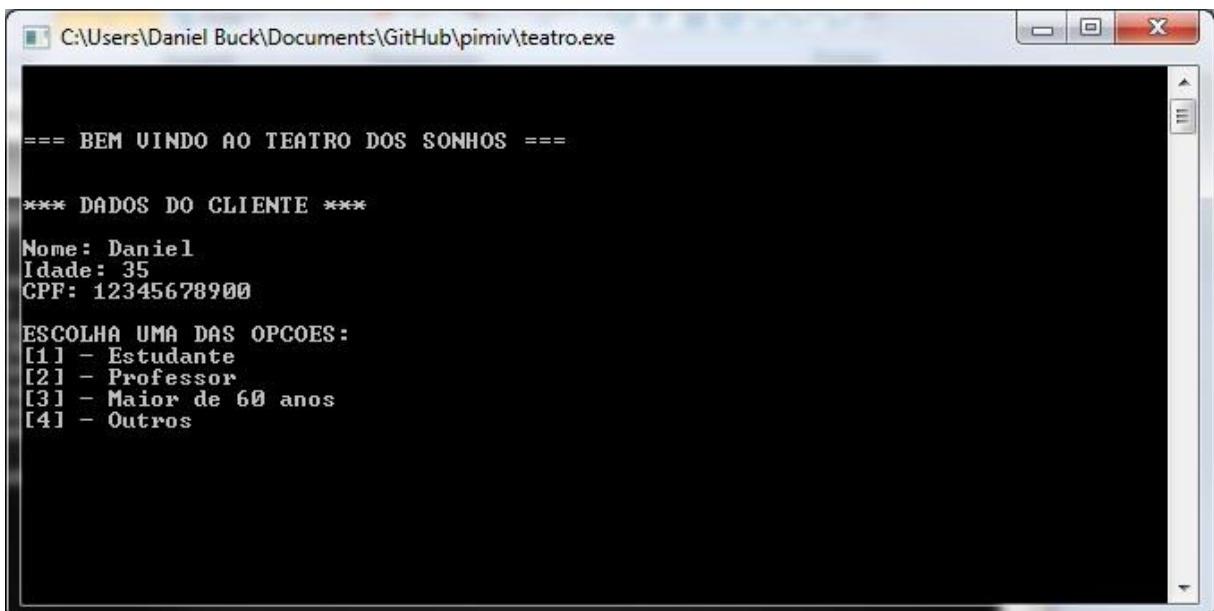
```
=== BEM VINDO AO TEATRO DOS SONHOS ===

*****
[1] - Fazer cadastro
[2] - Comprar Ingressos
[3] - Escolher poltrona
[4] - Imprimir Ingresso
[5] - Faturamento
[6] - Encerrar o Sistema
*****

Selecione uma opcao:
```

- **Cadastro de Clientes**

O usuário faz o cadastro, cada etapa faz com que o usuário digite cada um dos questionários como nome, idade, cpf.

A screenshot of the same application window, now showing the registration screen. It displays the entered information: 'Nome: Daniel', 'Idade: 35', and 'CPF: 12345678900'. Below this, it asks the user to choose an option from a list: [1] - Estudante, [2] - Professor, [3] - Maior de 60 anos, and [4] - Outros.

```
=== BEM VINDO AO TEATRO DOS SONHOS ===

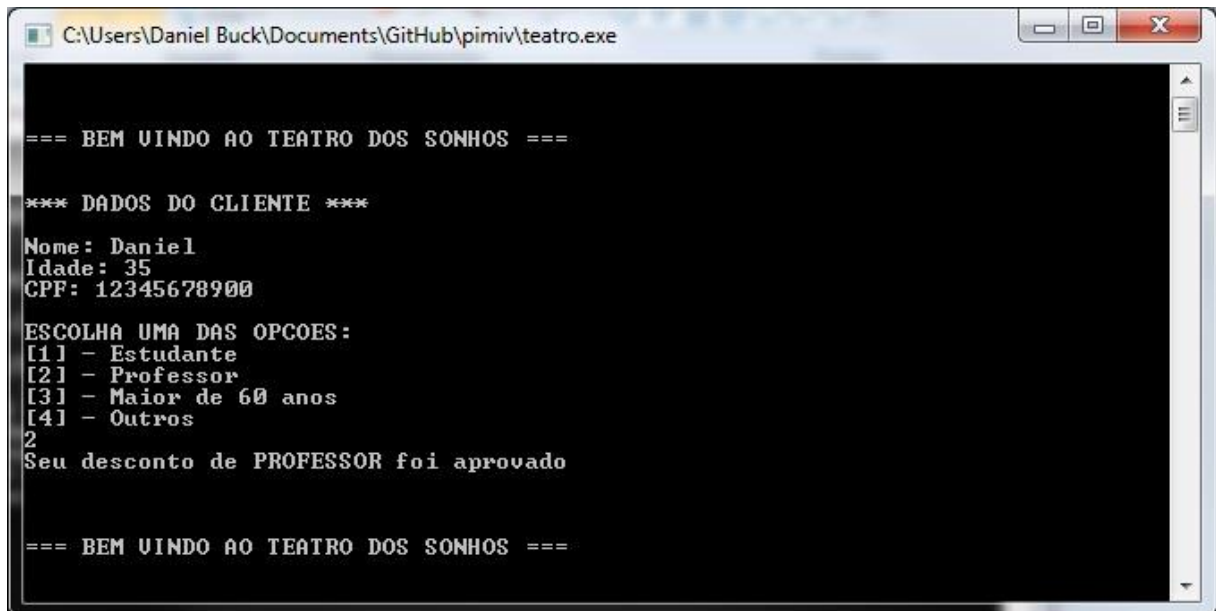
*** DADOS DO CLIENTE ***

Nome: Daniel
Idade: 35
CPF: 12345678900

ESCOLHA UMA DAS OPCOES:
[1] - Estudante
[2] - Professor
[3] - Maior de 60 anos
[4] - Outros
```

- **Escolha de Peças Teatrais**

Em seguida na opção 2 o usuário escolhe a peça que gostaria de assistir, em seguida escolhe o dia e horário, a quantidade de ingressos que irá comprar e faz o pagamento.



```
=== BEM VINDO AO TEATRO DOS SONHOS ===

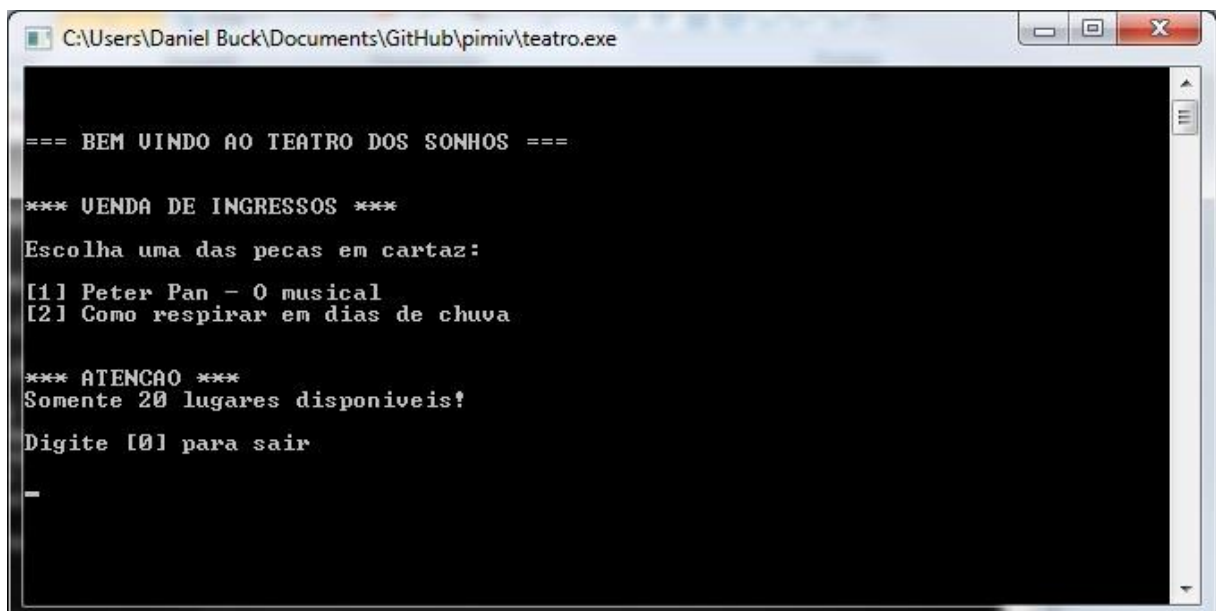
*** DADOS DO CLIENTE ***
Nome: Daniel
Idade: 35
CPF: 12345678900

ESCOLHA UMA DAS OPCOES:
[1] - Estudante
[2] - Professor
[3] - Maior de 60 anos
[4] - Outros
2
Seu desconto de PROFESSOR foi aprovado

=== BEM VINDO AO TEATRO DOS SONHOS ===
```

- **Opção 3**

Este é a reserva de poltrona, o usuário pode escolher visualizar quais poltronas estão disponíveis, realizar a escolha da poltrona, se o usuário não quiser mais aquele lugar ele pode fazer o cancelamento e realizar outra escolha.



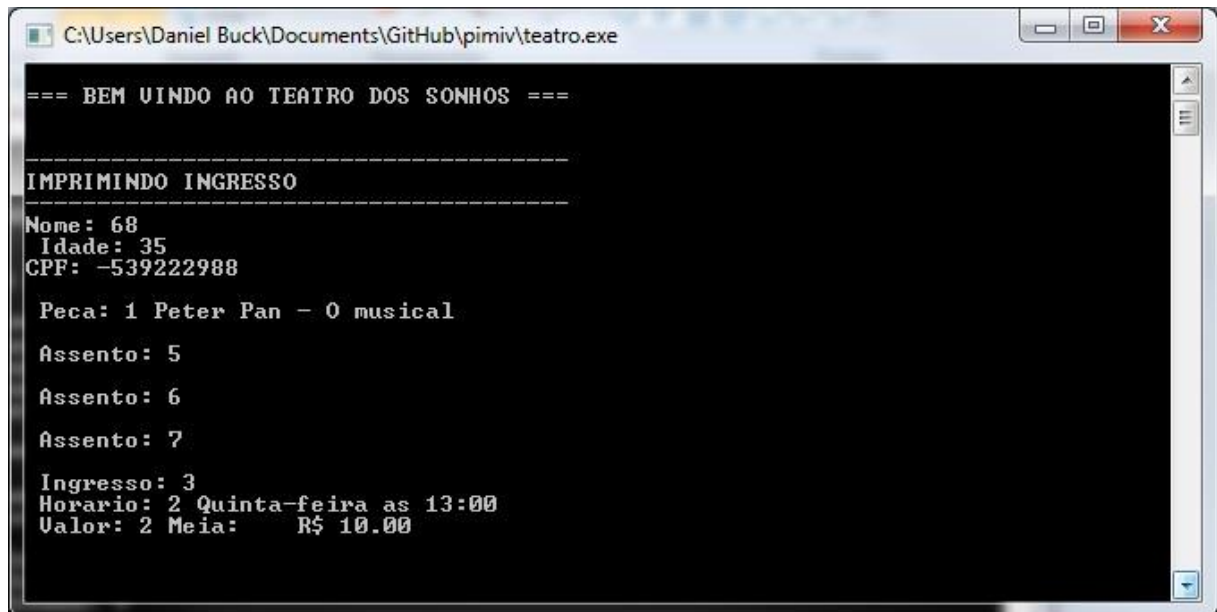
```
=== BEM VINDO AO TEATRO DOS SONHOS ===

*** VENDA DE INGRESSOS ***
Escolha uma das pecas em cartaz:
[1] Peter Pan - O musical
[2] Como respirar em dias de chuva

*** ATENCAO ***
Somente 20 lugares disponiveis!
Digite [0] para sair
-
```

- **Comprovante do Ingresso**

A opção em que o usuário imprime o seu ingresso, onde ele visualiza o cadastro que fez, e a compra do ingresso da peça que escolheu, e o número da poltrona.



```
C:\Users\Daniel Buck\Documents\GitHub\pimiv\teatro.exe

=== BEM VINDO AO TEATRO DOS SONHOS ===

-----
IMPRIMINDO INGRESSO
-----
Nome: 68
Idade: 35
CPF: -539222988

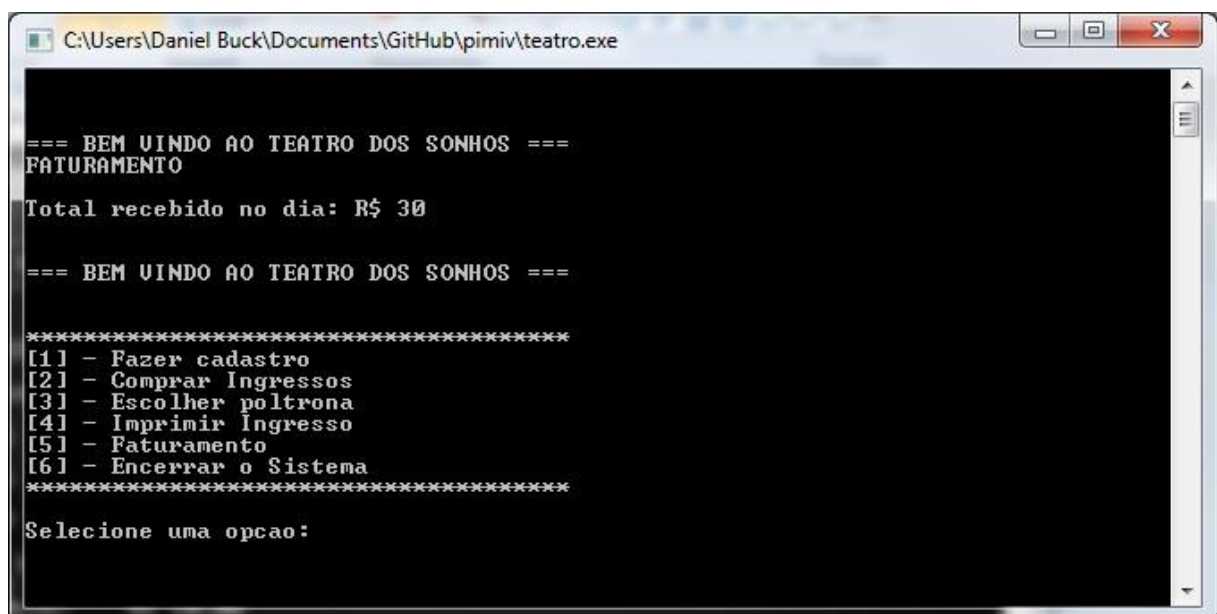
Peca: 1 Peter Pan - O musical

Assento: 5
Assento: 6
Assento: 7

Ingresso: 3
Horario: 2 Quinta-feira as 13:00
Valor: 2 Meia: R$ 10.00
```

- **Fechamento do Caixa**

Para o gestor aqui está a contabilização de venda de ingressos, e qual o faturamento do dia.



```
C:\Users\Daniel Buck\Documents\GitHub\pimiv\teatro.exe

=== BEM VINDO AO TEATRO DOS SONHOS ===
FATURAMENTO
Total recebido no dia: R$ 30

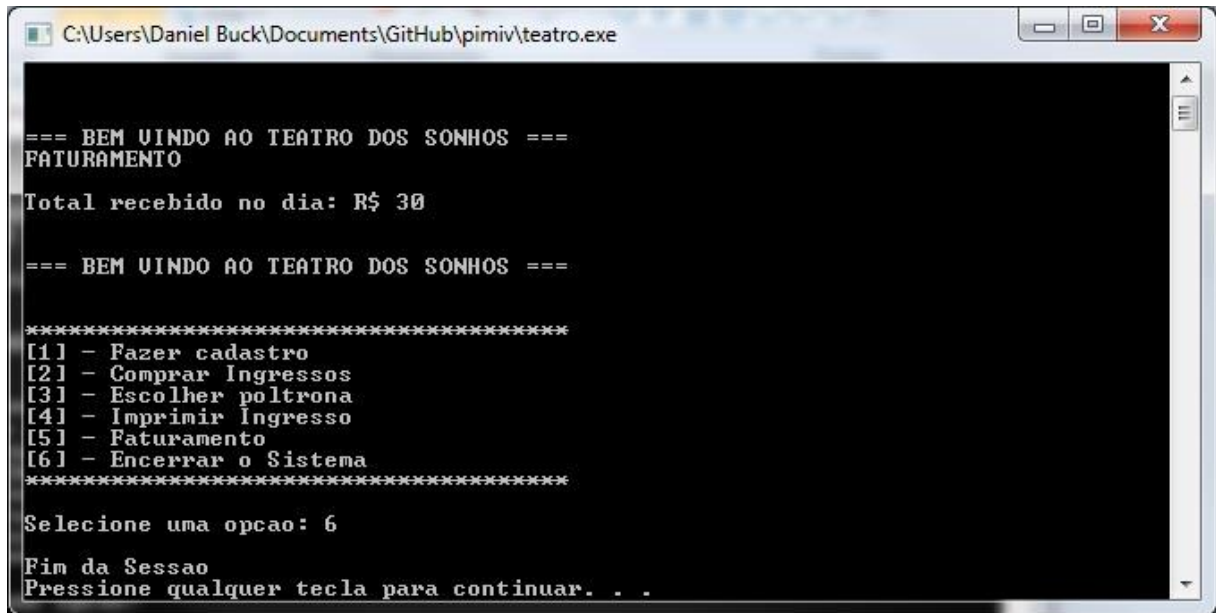
=== BEM VINDO AO TEATRO DOS SONHOS ===

*****
[1] - Fazer cadastro
[2] - Comprar Ingressos
[3] - Escolher poltrona
[4] - Imprimir Ingresso
[5] - Faturamento
[6] - Encerrar o Sistema
*****

Selecione uma opcao:
```

- **Encerramento do sistema**

Tela de encerramento do sistema.



```
C:\Users\Daniel Buck\Documents\GitHub\pimiv\teatro.exe

=== BEM VINDO AO TEATRO DOS SONHOS ===
FATURAMENTO
Total recebido no dia: R$ 30

=== BEM VINDO AO TEATRO DOS SONHOS ===

*****
[1] - Fazer cadastro
[2] - Comprar Ingressos
[3] - Escolher poltrona
[4] - Imprimir Ingresso
[5] - Faturamento
[6] - Encerrar o Sistema
*****

Selecione uma opcao: 6
Fim da Sessao
Pressione qualquer tecla para continuar. . .
```

7. Implantação de Software

Visto que o programa codificado e validado pela equipe de trabalho, consegue dar início ao processo de implantação.

A implantação do sistema se dá quando tudo foi testado ao máximo, para que o produto cumpra com papel para o qual foi desenvolvido.

8. Manutenção se Necessárias

Objetivando viabilizar o suporte e manutenção do software, utilizamos de técnicas de engenharia de software, tanto na coleta de requisitos (para evitar retrabalho) como no processo de desenvolvimento em si.

Para o processo de desenvolvimento foi utilizado o modelo cascata (ou clássico), do qual possui como forte característica, ser orientado à documentação, ou seja, tudo o que é desenvolvido, é devidamente documentado. Esta característica, além de ser indispensável durante o desenvolvimento, torna-se uma base para o suporte e manutenções futuras no software já em produção.

Conclusão

Aprendemos Como fazer um sistema, a entrega do sistema de acordo com o projeto, que conta com cadastro, venda do ingresso, desconto para estudantes e idosos, gratuidade para as crianças na terça feira, fechamento de caixa com a contabilidade de venda do dia.

Além disso prestar atenção na indentação do código, para a compilação ser prestativa, Fizemos com que tudo fosse interativo de acordo com que o cliente fosse necessitar, e para o caixa fazer a contabilidade no sistema.

Referências

https://homepages.dcc.ufmg.br/~rodolfo/aedsi-2-10/printf_scanf/printfscanf.html

<http://linguagemc.com.br/programando-em-c/>

<https://www.codingame.com/playgrounds/24988/programacao-c/funcoes-em-c>