

UNIVERSIDADE PAULISTA - EaD
PROJETO INTEGRADO MULTIDISCIPLINAR IV
RAFAELA SUMIE ANDRADE KURUMOTO

DESENVOLVIMENTO DE SISTEMA EM LINGUAGEM C
Vendas de Ingressos de Teatro

MARÍLIA - SP
2018

UNIVERSIDADE PAULISTA - EaD
PROJETO INTEGRADO MULTIDISCIPLINAR IV
RAFAELA SUMIE ANDRADE KURUMOTO

DESENVOLVIMENTO DE SISTEMA EM LINGUAGEM C

Vendas de Ingressos de Teatro

Projeto Integrado Multidisciplinar IV do
curso Superior de Tecnologia em
Análise e Desenvolvimento de Sistemas
Aluna: Rafaela Sumie Andrade Kurumoto
RA: 0500384

Orientadora: Vanessa Lessa

MARÍLIA - SP

2018

RESUMO

O Projeto Integrado Multidisciplinar IV do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas tem como objetivo pôr em prática os conhecimentos adquiridos através das disciplinas Linguagem e Técnicas de Programação e Engenharia de *Software* I.

Este projeto exemplifica o processo de desenvolvimento de um *software* em Linguagem C, para gerenciar as vendas de ingressos em uma bilheteria de teatro. A ferramenta utilizada para a criação do *software* foi a DEV C++. O projeto aborda temas como o modelo de processos de *software* cascata, incluindo análise de requisitos, algoritmos, fluxograma, além de técnicas utilizadas para o desenvolvimento do *software* em questão.

Palavras-chave: engenharia de *software*, linguagem C, desenvolvimento.

ABSTRACT

The main purpose of the Integrated Multidisciplinary Project IV from the System Analysis and Design course is to put into action knowledge acquired through the subjects Programming Languages and Techniques & Software Engineering I.

This project exemplifies a software development process in C programming language, with the purpose of managing tickets sales for a theater box office. The tool used to create this software was DEV C++. This project addresses topics such as the Waterfall software process model, including requirements analysis, algorithms, flowcharts, and techniques used to develop the software in question.

Key-words: software engineering, C programming language, design.

SUMÁRIO

INTRODUÇÃO	5
1 CONCEITOS DE SOFTWARE	6
1.1 Engenharia de Software	6
2 PROCESSO DE SOFTWARE	7
2.1 Modelos de Ciclo de Vida de Software	7
3 MODELO CASCATA	9
3.1 Vantagens e Desvantagens do Modelo Cascata	10
3.2 Justificativa da Escolha do Modelo Cascata	10
4 REQUERIMENTOS	11
5 PROJETO	12
5.1 Algoritmos	12
5.1.1 Descrição Narrativa	12
5.1.2 Fluxogramas	14
6 IMPLEMENTAÇÃO E VERIFICAÇÃO	16
6.1 Implementação	16
6.2 Verificação	16
7 MANUTENÇÃO	17
CONCLUSÃO	18
REFERÊNCIAS	19
APÊNDICE A	20
APÊNDICE B	21

INTRODUÇÃO

O projeto a seguir promove uma breve explanação sobre os conceitos da engenharia de *software* e sua importância no mundo atual. Um sistema para vendas de ingressos de teatro, foi desenvolvido para a aplicação de conhecimentos adquiridos nesse bimestre. O *software* foi codificado em linguagem C, utilizando um modelo de processo de *software*, detalhando todo seu desenvolvimento, desde levantamento de requisitos, design, testes, técnicas e tudo o que envolve a criação de um *software*.

1 CONCEITOS DE SOFTWARE

Existem diversos conceitos de *software*. Muitos dizem que são um conjunto de instruções executadas por computadores para resolver um determinado problema, mais especificamente chamados de programas de computadores. Porém, um *software* engloba também o conjunto de dados, arquivos de configuração, documentos que descrevem a operação e o uso do programa, tudo o que integra o desenvolvimento de um *software*. Sendo assim, de acordo com Pfleeger (2004, p.1): “O conceito mais geral de *software* compreende todo o conjunto de programas, procedimentos, dados e documentação associados a um sistema de computador, e não somente ao programa em si.”

Um *software* é desenvolvido e não fabricado, não se desgasta, mas evolui. O *software* está cada vez mais presente em nosso dia-a-dia, automatizando atividades, simplificando rotinas, tornando nossa vida mais efetiva.

1.1 Engenharia de Software

A Engenharia de *Software* é uma disciplina de engenharia que surgiu em meados de 1970, e engloba todos os aspectos da produção de *software*, abrangendo métodos, práticas e processos, visando organização e produtividade para se desenvolver um *software* de qualidade.

A Engenharia de *Software* é dividida em camadas, sendo elas:

- Qualidade - é a camada fundamental da Engenharia de *Software* que tem como foco o aperfeiçoamento contínuo dos processos;
- Processos - é a camada responsável por definir os métodos a serem aplicados, controlar a qualidade, estabelecer prazos, coordenar mudanças;
- Métodos - é a camada que abrange as tarefas a serem realizadas, como análise de requisitos, modelagem de projeto, testes e suporte;
- Ferramentas - é a camada que fornece suporte automatizado aos processos e métodos.

2 PROCESSO DE *SOFTWARE*

O Processo de Desenvolvimento de *Software* é o conjunto de atividades realizadas que levam à produção de um *software*.

As atividades fundamentais de um Processo de *Software* incluem:

- Concepção - especifica o que o sistema deve realizar, requisitos do usuário;
- Elaboração - especifica como o sistema será construído, elaboração da arquitetura;
- Construção - etapa de codificação e testes;
- Transição - implanta pequenas partes do sistema, realiza testes de homologação pelo cliente;
- Operação - depois de terminado e homologado, o sistema é colocado em produção;
- Manutenção - corrige problemas, adiciona novos atributos, adequações;
- Retirada - troca ou cancelamento do produto de *software*.

2.1 Modelos de Ciclo de Vida de *Software*

Os Modelos de Ciclo de Vida de *Software* são uma representação simplificada de como será construído o *software*. É um conjunto de atividades, ações, tarefas e produtos de trabalho necessários para se desenvolver um *software* de qualidade. Devem ser utilizados de acordo com as características do projeto, equipe, prazo, entre outros. Para equipes menores e com menos experiência, o ideal é um processo mais simples. Já para equipes numerosas, sistemas complexos, com diversos elementos ou usuários, um processo simples não será suficiente para assegurar a qualidade do produto de *software*.

Abaixo, estão listados alguns dos Modelos de Ciclo de Vida de *Software* mais populares e suas descrições:

- Modelo Cascata - foi um dos primeiros modelos de ciclo de vida que surgiu e é a base de muitos outros ciclos usados nos dias atuais. Consiste de um modelo linear, em que deve-se realizar cada passo antes de prosseguir para o próximo.

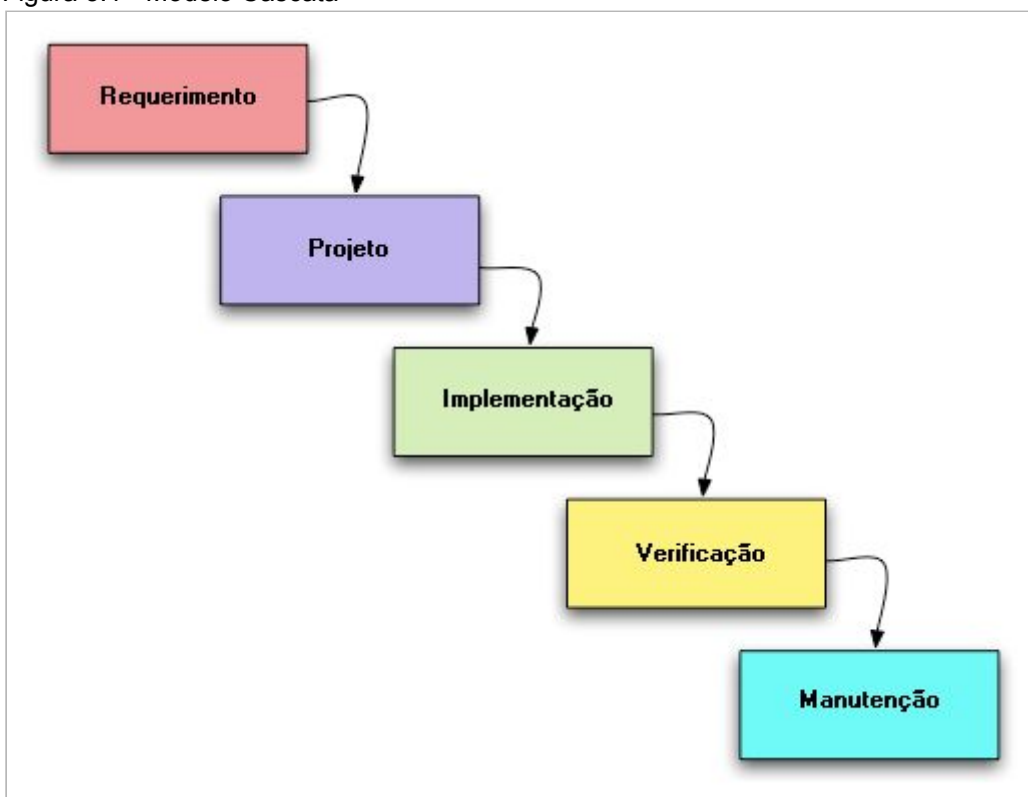
- Modelo Incremental - segue a mesma linha de idéia do Modelo Cascata, porém, sendo possível dividir o *software* em partes entregáveis, de modo que o cliente possa ver resultados de pequenos pedaços do *software*. Em cada incremento contém um processo em cascata.
- Modelo Orientado a Reuso - constitui-se de componentes de sistemas já existentes que são utilizados em um desenvolvimento de *software*, em vez de começar um *software* do zero.

Com base nos modelos citados acima, o Modelo de Ciclo de Vida mais adequado para o desenvolvimento do *software* proposto pela orientadora do Projeto Integrado Multidisciplinar IV é o Modelo Cascata.

3 MODELO CASCATA

O Modelo Cascata (Figura 3.1) foi um dos primeiros modelos de ciclo de vida a surgir e serve de base para muitos outros modelos utilizados nos dias atuais. Os estágios do modelo em Cascata devem ser executados em sequência, devendo concluir um estágio antes de iniciar o próximo.

Figura 3.1 - Modelo Cascata



Fonte: Wikipedia, 2007.

Os estágios do Modelo Cascata constituem-se de:

- **Requerimento** - refere-se a etapa em que o cliente especifica seus requisitos, os serviços, restrições e metas do sistema a ser desenvolvido. Também conhecido como Análise ou Definição de Requisitos.
- **Projeto** - etapa em que define-se a arquitetura do sistema, realizando o planejamento, transformando em documentos capazes de serem interpretados pelos programadores.

- Implementação - é nesta etapa que o projeto será codificado e cada parte do programa será testada para se assegurar de que atendem suas especificações.
- Verificação - depois de finalizado, as partes individuais do programa serão integradas e testadas. Após a validação, o sistema de *software* é entregue ao cliente.
- Manutenção - é a etapa em que se corrige os erros, ou atualiza-se o *software* para que ele se adeque às descobertas de novos requisitos.

3.1 Vantagens e Desvantagens do Modelo Cascata

O Modelo Cascata é um modelo de ciclo de vida simples, facilitando a compreensão do que deve ser realizado pelos desenvolvedores. A documentação produzida em cada fase do ciclo, possibilita o monitoramento do progresso aos gerentes.

Um dos problemas do Modelo Cascata é que o projeto só é visível ao final de todo o ciclo, sendo assim, o cliente tem que ter muita paciência. Outro problema que ocorre é que alguns membros da equipe necessitam aguardar a conclusão de tarefas de outros membros para que eles possam dar sequência ao trabalho.

Em geral, o Modelo Cascata é mais indicado quando as especificações dos requisitos são bem claras e quando o projeto é simples e pequeno.

3.2 Justificativa da Escolha do Modelo Cascata

De acordo com as especificações do *software* a ser desenvolvido, a escolha do Modelo Cascata se mostra a mais adequada para uso. Os requisitos das funcionalidades do *software* estão bem especificadas e não haverá mudanças, o sistema a ser desenvolvido é simples, o prazo para realizá-lo é curto, o *software* só será entregue após a conclusão do mesmo.

Nos capítulos seguintes, seguem as etapas do Modelo Cascata, com detalhamento das atividades realizadas para o desenvolvimento do sistema de *software* proposto pelo Projeto Integrado Multidisciplinar IV.

4 REQUERIMENTOS

O projeto proposto foi o desenvolvimento de um sistema para venda de ingressos de teatro, que deverá ser desenvolvido em linguagem de programação C e conter os seguintes critérios:

1. Levar em consideração uma tabela de valores para os ingressos;
2. Contemplar meia entrada (50% de desconto) a:
 - a. estudantes,
 - b. crianças de 02 a 12 anos,
 - c. adultos a partir de 60 anos,
 - d. professores da rede pública de ensino.
3. Contemplar 100% de desconto para crianças carentes da rede pública de ensino às terças-feiras¹;
4. Emitir um ticket (na tela) informando data, horário, nome da peça teatral e número da poltrona;
5. Não ser possível a venda de um ingresso que contém o mesmo número de poltrona, para a mesma peça, no mesmo horário de outro ingresso já emitido;
6. Informar quando os ingressos de determinada peça estiverem esgotados, não permitindo mais a venda do mesmo;
7. Fazer a gestão do caixa, informando:
 - a. Todas as movimentações do dia,
 - b. Saldo do fechamento.
8. O código-fonte deve ser organizado, acurado, contendo comentários explicativos para melhor entendimento.

¹ No sistema desenvolvido, se o dia da semana não for terça-feira, haverá um desconto de 50% por serem estudantes.

5 PROJETO

Nesta etapa, são realizadas as documentações de especificações de como o sistema de *software* deve ser desenvolvido, definindo a arquitetura de modo que seja facilmente compreendida ao ser passada para a etapa de Implementação (codificação).

A linguagem de programação a ser utilizada será C, desenvolvida com a ferramenta DEV C++. Sendo uma linguagem estruturada, tem ênfase no uso de laços de repetição, condicionais, sub rotinas e estruturas em bloco.

5.1 Algoritmos

Algoritmo é uma sequência de instruções finitas, bem definidas, que tem como objetivo a resolução de um problema. Um programa de computador é um algoritmo escrito em uma linguagem de programação.

Existem três formas de algoritmos que podem ser utilizados na programação:

- Descrição Narrativa - utiliza a linguagem natural do dia-a-dia para descrever as instruções. Ela é a mais simples porém sujeita a ambiguidades.
- Fluxograma - é a representação por meio de símbolos que contêm os passos para a resolução do problema. É de fácil entendimento, porém exige entendimento dos símbolos e é de difícil edição.
- Pseudocódigo/Portugol - é a estrutura lógica, próxima à linguagem de programação, mas escrita em português. É mais fácil de ser passada para uma linguagem de programação, porém é necessário ter conhecimento de lógica de programação.

Com base nas definições acima, foram desenvolvidos uma descrição narrativa e um fluxograma para o desenvolvimento do sistema de *software* proposto.

5.1.1 Descrição Narrativa

Ao iniciar o sistema de *software*, na tela apresentará uma nota de boas vindas ao programa, com um menu.

O usuário do sistema, no caso o(a) vendedor(a) de ingressos, deve escolher uma das opções: “iniciar uma venda”, “fechamento do caixa” ou “encerrar programa”.

Caso a escolha seja “iniciar uma venda”, o sistema irá apresentar um menu para escolha do da peça teatral.

Após a escolha da peça teatral, um outro menu aparecerá, mostrando as datas e horários disponíveis para a peça teatral escolhida.

Em seguida, o sistema apresenta um mapa com as poltronas disponíveis para a sessão escolhida, onde o comprador poderá escolher quantas poltronas desejar.

Se a poltrona selecionada estiver ocupada, informar na tela que não será possível continuar a compra, e escolher novamente.

Se a poltrona selecionada estiver disponível, perguntar se o comprador deseja escolher mais uma poltrona, se sim, repita, se não, prossiga.

Mostrar um novo menu perguntando para cada ingresso se o espectador é “estudante”, ou tem “idade entre 02 a 12 anos”, ou tem “idade a partir de 60 anos”, ou é “professor de rede pública de ensino”, ou é “criança carente da rede pública de ensino”, ou “nenhuma das opções”.

Caso a escolha seja “estudante”, “idade entre 02 a 12 anos”, “idade a partir de 60 anos”, ou “professor de rede pública de ensino”, conceder desconto de 50%.

Caso a escolha seja “criança carente da rede pública de ensino” e o dia da semana seja uma terça-feira, conceder desconto de 100% no ingresso, se não, conceder o desconto de estudante (50%).

Caso a escolha seja “nenhuma das opções”, o espectador deverá pagar o valor integral do ingresso.

Mostrar na tela a soma do valor de todos os ingressos selecionados pelo comprador e perguntar a forma de pagamento

Se a forma de pagamento for “em dinheiro”, entrar com o valor recebido, calcular o troco e finalizar compra.

Se o valor recebido for menor do que o valor devido, ou o troco for maior do que o valor no caixa², exibir uma mensagem de erro.

Caso a forma de pagamento seja “no cartão”, apenas passar o cartão e finalizar a compra.

² O caixa se inicia com R\$100,00 para troco, caso a primeira venda necessite.

Após finalização do pagamento, mostrar o ticket de cada espectador na tela, com nome da peça teatral, data e hora. Voltar à tela inicial com o menu para “início de venda”, “fechamento de caixa”, ou “encerrar programa”.

Se a escolha for “fechamento de caixa”, mostrar na tela todas as movimentações do dia e o saldo final do caixa.

Se a escolha for “encerrar programa”, finalizar o sistema.

5.1.2 Fluxogramas

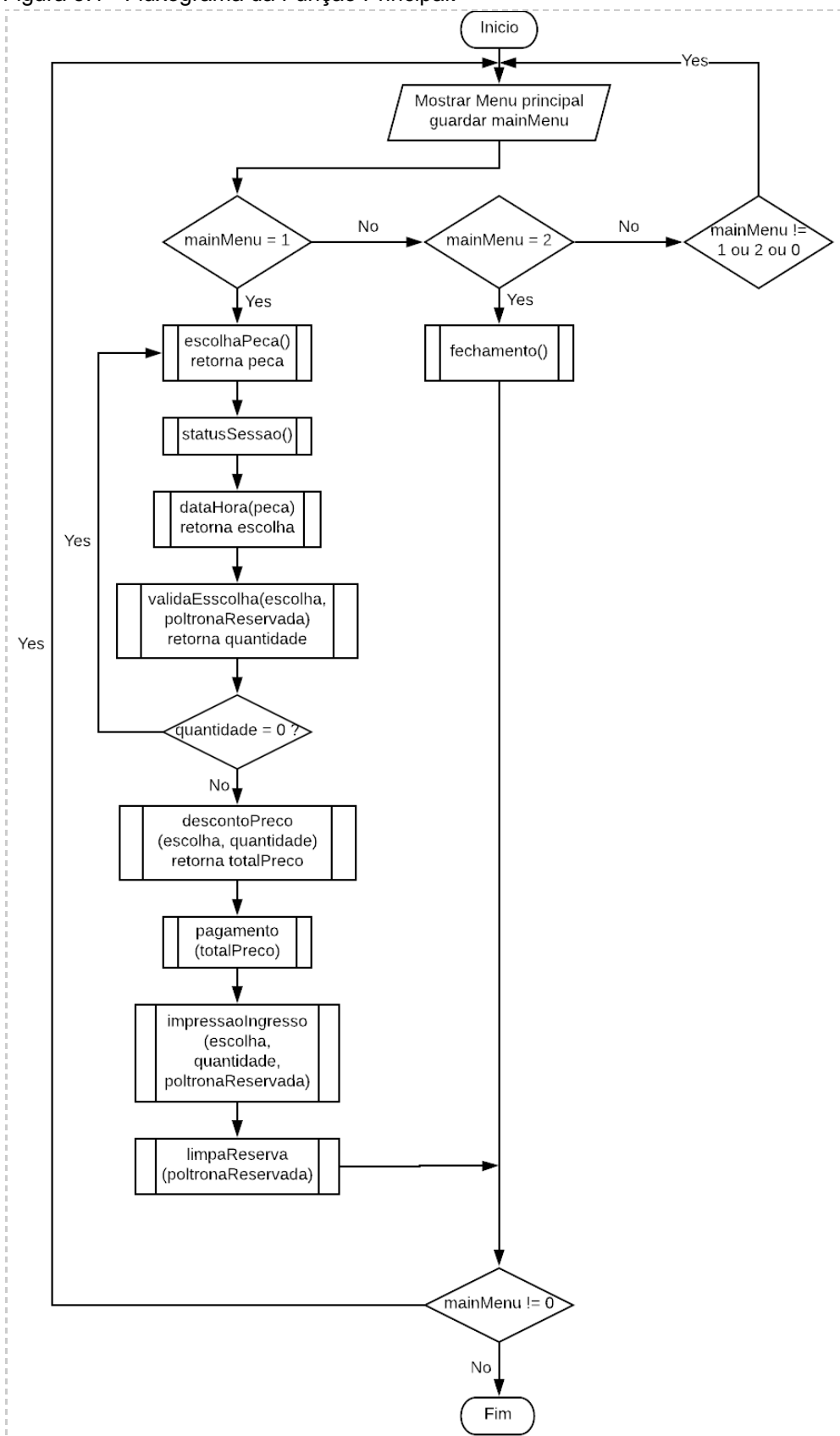
De acordo com a descrição narrativa acima, foi desenvolvido o fluxograma da função principal (Figura 5.1), contendo as chamadas das funções em que o programa foi dividido.

A função principal constitui-se de um grande loop (primeiro) com validação apenas ao final do mesmo. Dentro do loop, há um outro loop (segundo), que possui dentro a tela inicial com a apresentação do menu com as opções “início de vendas”, “fechamento de caixa” e “sair do programa”. Ao escolher a opção de início de vendas, segue-se o fluxo para a função de “escolha da peça”, que retorna um valor inteiro. Chama-se o procedimento para “atualizar o status da sessão”, e o valor retornado na primeira função é passado para a terceira função, “escolher a data e hora”, que retorna um valor inteiro. A quarta função a ser chamada checa a escolha, chamando uma outra função dentro dela (não detalhada no fluxograma) para a escolha de poltronas e retornando uma quantidade de ingressos reservados.

Se a quantidade de ingressos for maior do que ‘0’, o loop (segundo) é finalizado e chama-se a função que “descontos”, calculando o valor de acordo com as regras. Esta função retorna o valor total a ser pago que é enviado para a função “pagamento”. Após finalizar o pagamento, é chamada a função de “exibição dos ingressos na tela”, em seguida é chamado o procedimento de “limpar a reserva”, preparando o programa para a próxima venda.

Retornando para o início, o usuário pode escolher novamente entre “iniciar de vendas”, “fechamento de caixa” ou “sair do programa”. Ao optar por “fechamento de caixa”, é chamada a função que “exibe o relatório do caixa”. Ao optar por “sair do programa”, o sistema sai do loop (primeiro) e é finalizado.

Figura 5.1 - Fluxograma da Função Principal.



Fonte: Rafaela Kurumoto, 2018.

6 IMPLEMENTAÇÃO E VERIFICAÇÃO

6.1 Implementação

A fase de Implementação é onde o sistema é escrito em uma linguagem de programação, ou seja, é a codificação. Quanto mais detalhada for a fase de Projeto, menos complicada se torna a codificação. Durante a codificação, o sistema é testado diversas vezes para validar sua funcionalidade enquanto ele é desenvolvido.

O sistema de bilheteria foi codificado em linguagem C, utilizando estruturas para a manipulação de dados como: estruturas básicas (variáveis e instruções); estruturas de controle (if, else, switch, for, while); vetores e operações aritméticas.

No sistema foram incluídas verificações de entrada utilizando técnicas de programação, assim, se o usuário entrar uma opção inválida em qualquer um dos menus, uma mensagem de erro é exibida na tela, ou não será possível continuar até que entre com um valor válido.

Foram realizados diversos testes ao longo de sua codificação para identificar erros de lógica ou sintaxe.

6.2 Verificação

A fase de Verificação ocorre depois que a etapa de Implementação está concluída. Durante esta etapa, o sistema é testado para verificar sua funcionalidade externa, a lógica interna e se todos os requisitos determinados anteriormente foram atendidos. Também são realizados testes de entrada para determinar se o sistema está produzindo resultados eficientes.

Para a etapa de Verificação do sistema de bilheteria, primeiramente foram transcritos em um papel os resultados esperados em cada situação. Em seguida, os mesmos dados de entrada foram usados no sistema, verificando se as saídas eram as mesmas esperadas. Foram verificadas também as funcionalidades do sistema, bem como vendas de bilhetes de diferentes peças de teatro, reserva de poltronas, esgotamento de sessão, fechamento de caixa, impressão de ingressos e entradas inválidas. Obtendo êxito em todos os testes realizados, o sistema está pronto para ser entregue.

7 MANUTENÇÃO

A Manutenção é a última fase do Modelo Cascata e ocorre depois que o produto de *software* foi entregue ao cliente. É uma fase de melhorias e otimização feitos no *software* para garantir seu bom desempenho. Podem ocorrer para reparar erros, por adaptação à mudanças, por acréscimos de funcionalidades ou para resolver problemas de segurança.

A fase de Manutenção é a mais longa do ciclo de vida de um *software*. A principal razão disso, é o tempo em que o *software* permanece em funcionamento até que alcance o final de seu ciclo de vida. Um *software* alcança seu fim quando se torna muito ultrapassado, ou quando os gastos para sua manutenção são demasiados.

CONCLUSÃO

Ao fim do Projeto Integrado Multidisciplinar, concluí que a Engenharia de *Software* é indispensável e é importante ter conhecimento sobre todos os estágios de um processo de desenvolvimento de *software*, assim como os vários modelos de ciclo de vida a serem seguidos. Dessa maneira, um desenvolvedor pode aprimorar a qualidade de seu trabalho, escolhendo o modelo de ciclo de vida ideal para o *software* a ser desenvolvido.

O modelo Cascata foi o mais adequado para o desenvolvimento do *software* proposto, porém, durante a fase de Implementação, necessitei voltar para a etapa de Projeto, pois o mesmo possuía erros de lógica. Sendo assim, não segui sua estrutura linear.

A codificação do *software* se tornou mais fácil com o uso de fluxogramas e de descrições narrativas. A linguagem de programação C é simples de ser compreendida, porém necessita muita cautela para não ocorrer erros de sintaxe e lógica. Utilizei as técnicas e estruturas de programação que achei mais adequadas, e as que compreendi melhor. Dividi o programa em funções e procedimentos para o código permanecer organizado.

Testes foram realizados a cada adição de código, facilitando a correção de erros. Do mesmo modo, foi realizado um teste final para validar todas as funcionalidades propostas, entregando um *software* de qualidade, sendo esta a finalidade principal de um desenvolvedor de *software*.

REFERÊNCIAS

GUEDES, Sergio. (Org.) **Lógica de Programação Algorítmica**. São Paulo: Pearson, 2014.

MEDEIROS, Higor. Introdução ao Modelo Cascata. **Devmedia**, 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-modelo-cascata/29843>>. Acesso em: 11 de novembro de 2018.

MODELO DE CICLO DE VIDA. **Wikipedia**, 2017. Disponível em: <https://pt.wikipedia.org/wiki/Modelo_de_ciclo_de_vida>. Acesso em: 11 de novembro de 2018.

MODELO EM CASCATA, **Wikipedia**, 2018. Disponível em: <https://pt.wikipedia.org/wiki/Modelo_em_cascata>. Acesso em: 11 de novembro de 2018.

PFLEEGER, Shari Lawrence. **Engenharia de Software**: Tradução Dino Franklin. São Paulo: Pearson, 2011

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução Kalinka Oliveira; Ivan Bosnic. São Paulo: Pearson, 2011

APÊNDICE A - Testes e Resultados

Testes realizados sem reiniciar o programa, acumulando assim o valor do caixa e contagem de ingressos vendidos para cada peça, levando em consideração a tabela de preços e descontos abaixo:

- O Rei Leão - R\$25,00
- Os Improváveis - R\$15,00
- Sai de Baixo - R\$20,00
- (1) Estudantes - 50% de desconto
- (2) Idade 2 a 12 anos - 50% de desconto
- (3) Idade a partir de 60 anos - 50% de desconto
- (4) Professores da rede pública - 50% de desconto
- (5) Crianças carentes da rede pública às terças-feiras 100%, demais dias 50%
- (6) Nenhuma das alternativas - sem desconto


Teste #	Entrada	Saída esperada	Fundo de caixa	Passou ?
1	<ul style="list-style-type: none"> • Rei Leão x2 • Descontos: (1) e (6) • Pago em dinheiro 	<ul style="list-style-type: none"> • Total: R\$37.50 • Total ingressos: 2 sendo: <ul style="list-style-type: none"> • Rei Leão: 2 • Os Improváveis: 0 • Sai de Baixo: 0 	<ul style="list-style-type: none"> • Dinheiro: R\$37.50 • Cartão: R\$0 • Total: R\$37.50 • inteira - 1 • 50% - 1 • 100% - 0 	✓
2	<ul style="list-style-type: none"> • Sai de Baixo x7 • Descontos: (1), (3), (6), (6), (5), (1), (2) • Pago em cartão 	<ul style="list-style-type: none"> • Total: R\$90.00 • Total ingressos: 9 sendo: <ul style="list-style-type: none"> • Rei Leão: 2 • Os Improváveis: 0 • Sai de Baixo: 7 	<ul style="list-style-type: none"> • Dinheiro: R\$37.50 • Cartão: R\$90.00 • Total: R\$127.50 • inteira - 3 • 50% - 6 • 100% - 0 	✓
3	<ul style="list-style-type: none"> • Improváveis x5 • Descontos: (6), (6), (6), (6), (5) • Pago em cartão 	<ul style="list-style-type: none"> • Total: R\$67.50 • Total ingressos: 14 sendo: <ul style="list-style-type: none"> • Rei Leão: 2 • Os Improváveis: 5 • Sai de Baixo: 7 	<ul style="list-style-type: none"> • Dinheiro: R\$37.50 • Cartão: R\$157.50 • Total: R\$195.00 • inteira - 7 • 50% - 7 • 100% - 0 	✓
4	<ul style="list-style-type: none"> • Sai de Baixo x3 • Descontos: (6), (5), (6) • Pago em dinheiro 	<ul style="list-style-type: none"> • Total: R\$50.00 • Total ingressos: 17 sendo: <ul style="list-style-type: none"> • Rei Leão: 2 • Os Improváveis: 5 • Sai de Baixo: 10 	<ul style="list-style-type: none"> • Dinheiro: R\$87.50 • Cartão: R\$157.50 • Total: R\$245.00 • inteira - 9 • 50% - 8 • 100% - 0 	✓

Obs: Os testes foram realizados em uma sexta-feira, por isso não foi concedido nenhum desconto de 100%.

APÊNDICE B - Screenshots

Os *screenshots* abaixo correspondem à diferentes etapas da venda dos testes realizados.

Imagem B.1 - Teste #1



```

D:\Courses\UNIP\Análise e Desenvolvimento de Sistemas\PIM IV\Bilheteria.exe
*** Fechamento do caixa ***

Ingressos vendidos no total:          2
Total de vendas para [O Rei Leao]:    2
Total de vendas para [Os Improvaveis]: 0
Total de vendas para [Sai de Baixo]:  0

Pagos no valor integral:              1
Descontos de 50 por cento:            1
Descontos de 100 por cento:           0

Total arrecadado no dinheiro:         R$37.50
Total arrecadado no cartao:           R$0.00
Total arrecadado:                     R$37.50

Press any key to continue . . .
  
```

Fonte: Rafaela Kurumoto

Imagem B.2- Teste #2



```

D:\Courses\UNIP\Análise e Desenvolvimento de Sistemas\PIM IV\Bilheteria.exe
Pro Seleção de uma poltrona disponível

-----PALCO-----

[P 1] [P 2] [P 3] [P 4] [P 5]
[P 6] [P 7] [P 8] [P 9] [P 10]
[xxxx] [xxxx] [xxxx] [xxxx] [xxxx]
[xxxx] [xxxx] [P 18] [P 19] [P 20]

Poltrona reservada com sucesso!

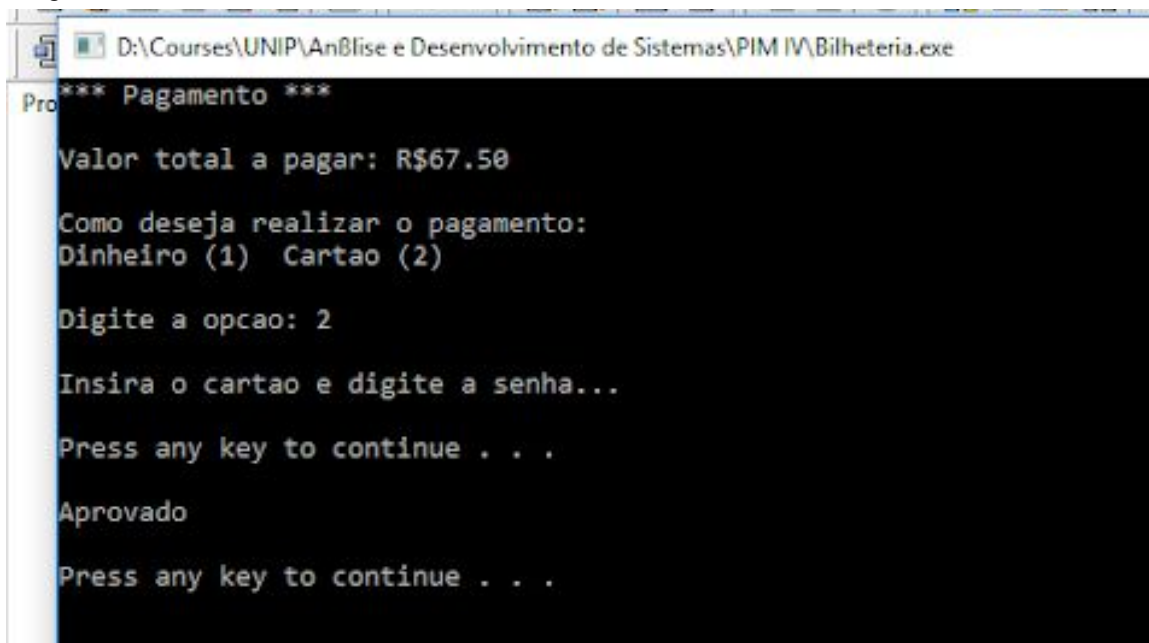
Gostaria de reservar mais um lugar? (1)Sim ou (0)Nao: 0

Quantidade de lugares reservados: 7

Press any key to continue . . .
  
```

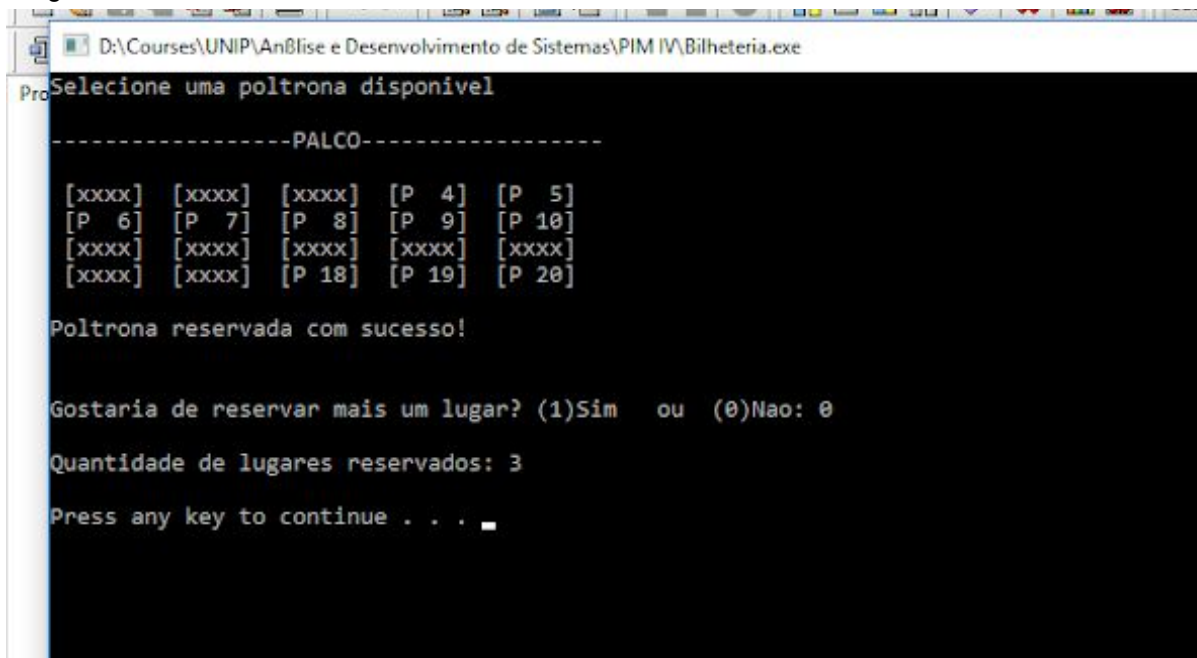
Fonte: Rafaela Kurumoto

Imagem B.3 - Teste #3



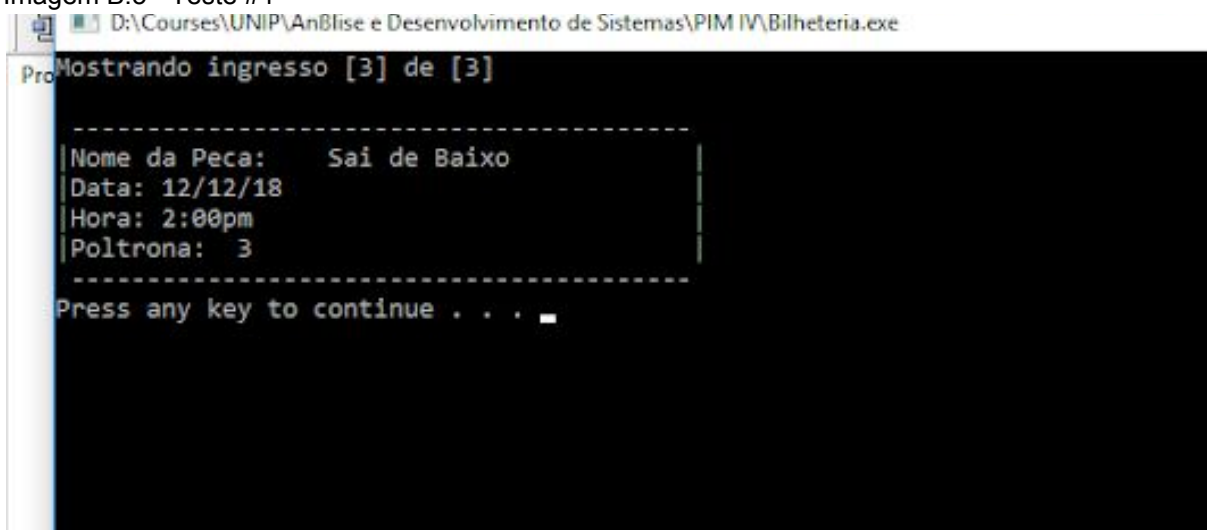
Fonte: Rafaela Kurumoto

Imagem B.4 - Teste #4



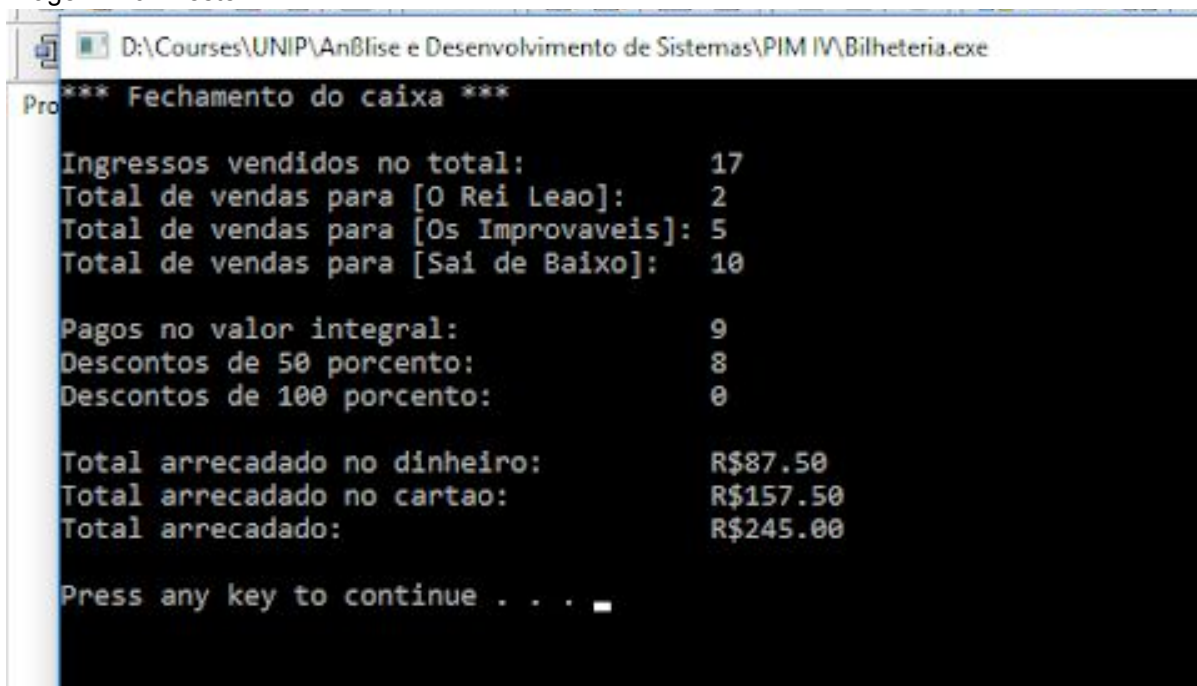
Fonte: Rafaela Kurumoto

Imagem B.5 - Teste #4



Fonte: Rafaela Kurumoto

Imagem B.6 - Teste #4



Fonte: Rafaela Kurumoto

Após mudar a data do computador para um dia que cai na terça-feira e reiniciar o software, obtive o seguinte resultado:


Teste #	Entrada	Saída esperada	Fundo de caixa	Passou ?
1	<ul style="list-style-type: none"> Sai de Baixo x5 Descontos: (5), (3), (6), (6), (5) Pago em cartão 	<ul style="list-style-type: none"> Total: R\$50.00 Total ingressos: 5 sendo: Rei Leão: 0 Os Improváveis: 0 Sai de Baixo: 5 	<ul style="list-style-type: none"> Dinheiro: R\$0.00 Cartão: R\$50.00 Total: R\$50.00 inteira - 2 50% - 1 100% - 2 	

Imagem B.7 - Resultado de Terça-Feira (desconto)

```
D:\Courses\UNIP\Análise e Desenvolvimento de Sistemas\PIM IV\Bilheteria - Rafaela Kurumoto.exe
Informacao sobre o espectador[5] de [5]:
Estudante (1)
Idade entre 2 a 12 anos (2)
Idade a partir de 60 anos (3)
Professor de rede publica de ensino (4)
Crianca carente da rede publica de ensino (5)
Nenhuma das alternativas (6)

Opcao escolhida: 5

Parabens, hoje e terca-feira e voce tera desconto de 100 por cento!!!

Valor do ingresso: R$0.00

Press any key to continue . . .
```

Fonte: Rafaela Kurumoto

Imagem B.8 - Resultado de Terça-Feira (caixa)

```
D:\Courses\UNIP\Análise e Desenvolvimento de Sistemas\PIM IV\Bilheteria - Rafaela Kurumoto.exe
*** Fechamento do caixa ***

Ingressos vendidos no total: 5
Total de vendas para [O Rei Leao]: 0
Total de vendas para [Os Improvaveis]: 0
Total de vendas para [Sai de Baixo]: 5

Pagos no valor integral: 2
Descontos de 50 por cento: 1
Descontos de 100 por cento: 2

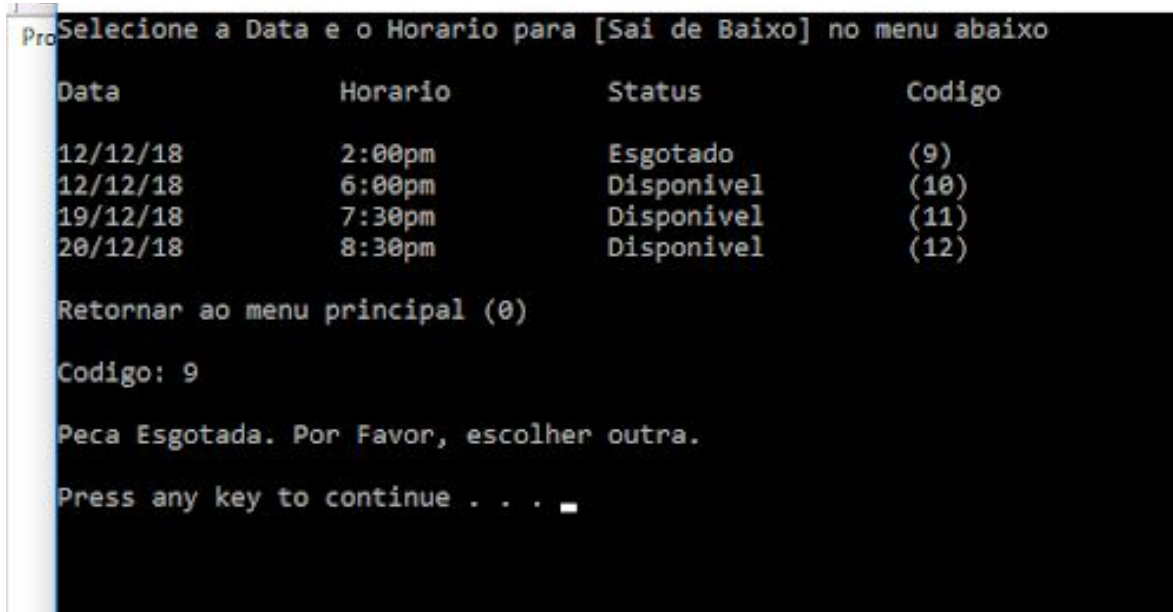
Total arrecadado no dinheiro: R$0.00
Total arrecadado no cartao: R$50.00
Total arrecadado: R$50.00

Press any key to continue . . .
```

Fonte: Rafaela Kurumoto

Abaixo, um exemplo de quando uma das sessões da peça está esgotada, não possibilitando mais a sua venda:

Imagem B.9 - Sessão esgotada



Fonte: Rafaela Kurumoto