

FACULDADE SENAI FATESG
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LETYCIA SOUZA, JOYCE AQUINO, GABRIEL CESSEL

PROJETO INTEGRADOR
Venda de Ingressos para o Teatro

GOIÂNIA – GO
2021

LETYCIA SOUZA, JOYCE AQUINO, GABRIEL CESSSEL

PROJETO INTEGRADOR

Venda de Ingressos para o teatro

Projeto Integrador apresentado à Faculdade Senai
Fatesg do curso Tecnólogo de Análise e
Desenvolvimento de Sistemas.

Orientadora: Prof. Dllubia Santclair Matias

SUMÁRIO

1. Introdução/Justificativa.....	pág. 4
2. Objetivos.....	pág. 5
2.1 Objetivo Geral.....	pág. 5
2.2 Objetivos Específicos.....	pág. 5
3. Metodologia/Desenvolvimento.....	pág. 5
3.1 Engenharia de Software	pág. 5
3.2 Algoritmo e Programação	pág. 5
3.3 Fundamentos da Arquitetura de Computadores	pág. 15
3.4 Lógica Matemática e Estatística	pág. 16
4. Resultado e Conclusões.....	pág. 18
5. Referências.....	pág. 19
6. Anexo.....	pág. 20

1. Justificativa:

Da mesma forma que utilizamos a tecnologia no dia a dia para facilitar nossa comunicação, hoje ela é uma grande aliada das organizações. Empresas que precisam da automatização, rapidez, precisão e armazenamento que só as máquinas têm...

Dada essa situação, o Senai Fatesg, nos deu a missão para criação do Software de um teatro destinado à venda de ingressos.

Oportunizando a nossa equipe, Java Café, a colocar em prática habilidades adquiridas durante o primeiro semestre do curso de Análise e Desenvolvimento de Sistemas.

O teatro ABC nos contratou então para a construção do Software para a venda de ingressos. Dentre os ganhos da empresa ao contratar nosso sistema, irei listar; o fator econômico, o teatro ABC usando nosso Serviço poderá reduzir seu número de funcionários, e incluir ao mesmo tempo comodidade para os seus clientes como por exemplo evitando filas, para compra das entradas.

2. Objetivos:

2. 1 Objetivo Geral - Criar um Software de venda de bilheteria para o teatro ABC.

2.2 Objetivos Específicos -

- Desenvolver a Lógica de Programação;
- Construir o documento, organizado, com as normas de formatação;
- Elaborar índice de vendas;
- Demonstrar Gráfico de Gantt.

3. DESENVOLVIMENTO: Tendo como base os conhecimentos ministrados em sala de aula, apresentamos a seguir, por disciplina, detalhando cada etapa das atividades desenvolvidas:

3.1 Engenharia de Software:

Gráfico de Gantt para a organização da equipe e inicio do projeto. Utilizando o site: google.gantt.com.

Esse foi feito a fim de registrar as atividades feitas no decorrer do trabalho, dentre elas existem as reuniões, realizadas pelo Google Meet, para análises de requisitos, das funcionalidades realizadas com possíveis feedbacks...

O gráfico de Gantt tem o intuito de demonstrar e organizar quais atividades foram concluídas e quando foram finalizadas. Usa visualmente o gráfico, de modo que evidencia quais tarefas são dependentes de outras e colocam essas atividades por ordem cronológica. Auxiliando cada integrante do grupo com os prazos e objetivos.

3-Algoritmo:

3.1-Levantamento de requisitos:

Após fazermos a reunião de abertura do Projeto e neste termos levantado os três principais requisitos solicitados para o funcionamento do software que consistem na compra do ingresso, na impressão do ingresso após o usuário digitar o CPF e o Relatório de Vendas do Ingresso. Fizemos em outra reunião para o levantamento dos requisitos específicos para conseguimos realizar os requisitos principais, assim, fizemos os seguintes levantamentos de requisitos específicos:

1- Comprar Ingresso:

- A. Na tela irá apresentar uma apresentação do software;
- B. Cliente informe o CPF com 11 dígitos e números inteiros;
- C. Se houver erro na digitação, irá aparecer na tela a mensagem "CPF inválido. Digite novamente!"
- D. Se for aceito, irá aparecer na tela o CPF, e uma mensagem de confirmação do CPF "Confirma";

- E. O usuário irá confirmar o CPF;
- F. Se não confirma o CPF, retornará para parte de informar o CPF;
- G. O cliente irá escolher a data (03/12/2021, 04/12/2021 e 05/12/2021);
- H. Irá aparecer na tela as peças em cartaz, o horário e as datas ;
- I. O cliente irá escolher a peça que deseja assistir;
- J. Irá escolher a sessão (manhã, tarde e noite) desta peça;
- K. Irá escolher a área que deseja assistir (Plateia A, Plateia B, Frisa, Camarote ou Balcão Nobre);
- L. Irá confirmar as informações;
- M. Irá escolher a poltrona que irá ocupar , estas poltronas estarão numeradas, e o sistema deverá mostrar as poltronas livres ou ocupadas;

Área	Qtde Poltronas
Plateia A	25
Plateia B	100
Frisa	5 por frisa
Camarote	10 por camarote
Balcão Nobre	50

- N. Cada poltrona terá um preço relacionado a área que foi escolhida, assim:

Área	Preço
Plateia A	40,00
Plateia B	60,00
Camarote	80,00
Frisa	120,00
Balcão Nobre	250,00

J- O sistema irá reservar o poltrona que ela comprou o ingresso;

H- O sistema irá mostrar na tela o cpf, a data, o nome da peça escolhida, a sessão, a área e numeração da poltrona e preço final;

2- Imprimir Ingresso:

- A. Nesta funcionalidade deverá permitir com que o cliente informe seu CPF ;
- B. Irá imprimir na tela o comprovante do ingresso comprado por ele; informando:

Data

Nome da peça

Horário da sessão

Área (Plateia A, Plateia B, Frisa, Camarote ou Balcão Nobre)

local da poltrona- numero dela

Preço

3. Estatística de Vendas- Nesta funcionalidade irá apresentar em tela:

O usuário assim que solicitar esta funcionalidade, irá visualizar em tela as estatísticas de vendas, que mostrará;

A. A peça que teve mais ingressos vendidos;

Neste de cada peça, iremos somar os ingressos vendidos de todas as sessões (manhã, tarde e noite) e de todos os dias. Depois iremos comparar a soma dos ingressos de cada peça e mostrar a que teve maior número de ingressos vendidos;

B. Mostrará a peça com menos ingressos vendidos.

Do resultado da soma dos ingressos da operação anterior, iremos mostrar a que teve o menor número de ingressos vendidos.

C. A sessão teve maior ocupação de poltronas:

Neste iremos somar os ingressos vendidos de cada sessão, separados por dia e por horário, e iremos comparar a venda dos ingressos de cada sessão e de cada dia, e mostraremos a sessão que teve mais ocupação, referente a qual dia e qual horário.

D. A sessão com menor ocupação de poltronas;

Conforme o cálculo anterior sobre a sessão com maior ocupação de poltronas, neste iremos comparar a somatória de ocupação de cada sessão, e retornar a sessão que teve menor ocupação de poltronas;

E. A peça mais lucrativa;

Neste, iremos somar os valores dos ingressos por peça, de todos os dias e comparamos estes valores e mostraremos a peça que teve mais lucro. Considerando que o valor do ingresso é obtido através da quantidade de ingressos por área e multiplicados pelo valor da área específica.

F. A Peça menos lucrativa;

Com o resultado da questão anterior, compare os valores e retorne a sessão menos lucrativa.

G. O Lucro médio do teatro com todas as áreas por peça;

Neste, iremos primeiramente somar os valores dos ingressos por peça de todos os dias, e considerando que para termos o valor de um ingresso unitário temos que multiplicar a quantidade de ingressos pela área que foi escolhida, e depois somaremos os valores dos ingressos de todas as peças e de todos os dias e dividiremos pela quantidade de peça para obter o valor do lucro médio do teatro com as peças.

3.2-A codificação do software de Vendas de ingressos:

A codificação do software só foi iniciada após os levantamentos de requisitos do software de “Venda de ingressos”, da definição do software para codificação e de levantamento de protótipos semelhantes na internet, como o site Ingresso Digital, que consolidaram as informações do que deveria ser feito no software.

Depois disto, realizamos a codificação no software Netbeans 8.2, utilizando a linguagem Java para programação. Como vimos que o software necessitava de muitas “ações” serem executadas, passamos a escrever o código baseado na estrutura conhecida como métodos em Java, que são blocos de código pertencentes a uma classe que realiza uma função específica, e está pode ser referenciada por outra estrutura. Desta forma, esta estrutura permite uma codificação mais organizada e uma manutenção mais fácil ao corrigir uma parte específica do código. E, com relação ao código, utilizamos: “estruturas de seleção” (if e else); “laços de repetição” (while, do-while e for) ; estruturas homogêneas (vetores e matrizes); switch e variáveis inteiras, double, booleanas, string e char, que serão tratadas no item a seguir, em “Princípios em Java” .

O código foi organizado em cinco arquivos denominados: “ComprarIngresso.java”; “Input.java”; “MapaDeDados.java”; “RelatórioDeEstatistica.java” e “Main.java”. No “Main.java” é onde o código será executado, assim, nele está o código que será inscrito na tela do usuário, a tela de inicialização contendo as opções se a caso o usuário queira, “comprar o ingresso”; “Imprimir o ingresso”, ou seja o usuário irá digitar o CPF que já comprou ingresso (s) e logo verá na tela ingresso comprado do CPF respectivo; ver o “Relatório de vendas” ou sair do programa. Para isto, nele consta o “chamamento” de blocos de código, dentro do switch, que ao digitar o número referente do que se deseja executar, o programa irá executar o código referente. Neste arquivo, também foi escrito uma estrutura de vetor que guardará os valores do ingresso referente a cada área (em : static double[] carregarValoresIngressos()), outra estrutura de vetor que guarda a quantidade poltronas referente a cada área (em: private static int[] carregarTotalLugaresTeatro()) e um inteiro “int disponíveis”, em forma de matriz que receberá o valor da quantidade de poltronas vagas referente a matriz de data, sessão e peça (em: private static int[][] carregarLugaresDisponiveis(int[] totalLugaresTeatro)).

No arquivo “Input.java” possuem funções que irá armazenar os valores das opções escolhida ou digitada pelo usuário ao comprar um ingresso, em forma sequenciada, o usuário irá digitar o CPF (neste será verificado se possui 11 números), irá escolher a data, a peça, a área, e nesta será mostrado o valor do ingresso em todas as áreas e a quantidade de poltronas disponíveis (vagas), de cada uma delas, e a após escolher a área, o usuário irá digitar a quantidade de ingressos que deseja comprar referente a área escolhida e todas estas ações terá uma função booleana para confirmar a escolha.

No arquivo “ComprarIngresso.java” é onde que, após o usuário escolher lá no sistema de repetição do arquivo Main.java, a opção para comprar ingresso, será executado as ações referentes a compra do ingresso, que para isto irá ter neste “chamamentos” as estruturas de bloco de códigos dos arquivos “Input.java” e do “MapaDeDados”, para que sejam feito neste arquivo a escolha da data, sessão e área. E, ao finalizar a compra do ingresso, neste arquivo, possui o método que irá mostrar as informações referente ao ingresso comprado como o CPF, o valor total dos ingressos, a peça e data escolhida.

No arquivo “MapaDeDados.java” colocamos as conversões de variáveis, de uma variável em outra, ou seja, de uma variável string em variável inteira ou virse versa, objetivando uma facilidade para interagir com as posições e valores das matrizes. Então, temos os valores lá convertidos de: string data para um inteiro que é o índice ou seja será

tratado em outros códigos como a posição da matriz (em: `int stringParaIndiceData`); o seu contrário, ou seja de inteiro data para a string data (em: `String intDataParaString`); de string sessão para inteiro sessão, onde será utilizado como índice da matriz referente a peça (em: `int stringParaIndiceSessao`); de string área para inteiro área, que da mesma forma será utilizado este como o índice da matriz referente a área. (em: `int stringParaIndiceArea`); de inteiro sessão para string sessão, no qual será utilizado esse valor no formato string (em: `String intSessaoParaString`) e inteiro área para string área (em: `String intAreaParaString`). Neste, arquivo também, colocamos além destas conversões, um bloco de código que retorna da matriz a quantidade de poltronas ocupadas (em: `int[][][] disponiveisParaOcupados`) .

E por último, no arquivo “RelatorioDeEstatistica.java” fizemos a codificação que atendesse a funcionalidade de “Relatório de Vendas” que se trata de informações estatísticas das vendas dos ingressos do teatro. Assim, codificamos para a atender a funcionalidade referente a peça com ingressos mais e menos vendidos, neste somamos a quantidade de ingressos vendidos de todos os dias e separados este valor por peças, ou seja, consiste, por exemplo, no total de ingressos vendidos durante o tempo de exposição da “peça 1” , e depois fizemos a comparação destes valores que mostraram o que tiveram maior e menor ingressos vendidos (em: `static void totalDeIngressos`); da sessão mais e menos ocupada, fizemos o somatório de ocupação de cada sessão por dia, e depois comparamos estes valores com todas as sessões e com todos os dias, expondo assim, a sessão/peça que teve mais ocupação, e está é referente a uma peça de uma data (em: `static void sessaoMaiorOcupacao`) e os seus respectivos índices (posições) da matriz; a sessão mais e menos lucrativa, na qual utilizou-se da informação da ocupação total por peça dos dias de exposição e multiplicou pelo o valor do ingresso da área referente (em: `static void sessaoLucroOcupacao`) e a funcionalidade referente ao lucro médio do teatro pelas peças, que utilizamos o somatório da ocupação de todas as peças e de todos os dias e multiplicamos pelo valor do ingresso referente a cada área de ingressos vendidos e dividimos este valor pela quantidade de peças (em: `static void lucroMedioTotalIngressos`) .

E durante todo o processo de codificação do software buscando uma qualidade no código e atender as funcionalidades requeridas, fazíamos assim que terminamos a codificação de uma funcionalidade um teste unitário referente a ela, e se acaso desse erro no teste este não passasse no teste, corrigimos o seu código e fazíamos o teste novamente até novamente até o código ser aprovado, se fosse, seguíamos com a codificação das próximas funcionalidades.

3.2.1- Princípios em Java:

3.2.1.1-Métodos em Java:

Escolhemos utilizar a técnica da modularização que consiste em dividir o problemas em partes, ao reduzir um problema ao conjunto de tarefas (módulos) que desenvolve algoritmo de forma independente, e que pode ser “chamado” ou requisitado em outras partes do programa. E, o gerenciamento das tarefas (módulos) é feito pelo algoritmo principal, ou módulo principal, que “chama” os módulos em um algoritmo principal em meio de funções ou procedimentos (Puga, pág. 101). Portanto, de acordo com o site Wikilivros , os métodos são :

“blocos de código que pertencem a uma classe e tem por finalidade realizar uma tarefa (...). Numa classe para realizar cálculos matemáticos poderia ser *somar*, *subtrair*, *dividir*, etc. Além disso, um papel fundamental dos mesmos é ter de evitar reescrever código para uma mesma função sempre que se deseja chamá-la.”Wikilivros. Disponível em: <<https://pt.wikibooks.org/wiki/Java/M%C3%A9todos>>. Acesso no dia 29/11/2021.

As vantagens de criar métodos está na redução do tamanho de código de um sistema, a sua modularização, ou seja, cada trecho de código realiza uma tarefa específica e a facilidade e agilidade na manutenção, alterando uma pequena parte do código. Assim, temos métodos que não possuem retorno (void), sendo suas informações utilizadas apenas em métodos do mesmo tipo durante o código, que se utiliza a seguinte sintaxe para este método:

```
qualificador void nomeDoMetodo ( <parâmetros>){ código }
```

Ele tem um qualificador é o método da classe, que pode ser um static ou void, e quando é static, ele é uma método da classe. O void executa a funcionalidade mas, não retorna nada para quem o chamou. Nome do método, a seguir, e entre parênteses tem os parâmetros separados por vírgula, os parâmetros conforme o site Wikibooks:

“representa um valor a ser informado para que seja processado pelo método. Um método pode não possuir parâmetros ou possuir vários parâmetros. Cada parâmetro deve ser acompanhado de seu respectivo tipo. Observando o método main(), podemos perceber que ele possui um parâmetro chamado args, que é um vetor do tipo String: **public static void main**(String[] args){ }.” Wikibooks. Disponível em: <<https://pt.wikibooks.org/wiki/Java/M%C3%A9todos>>. Acesso no dia 29/11/2021.

Já o método Static, de acordo com Thiago (2012):

"Os métodos static ou métodos da classe são funções que não dependem de nenhuma variável de instância, quando invocados executam uma função sem a dependência do conteúdo de um objeto ou a execução da instância de uma classe, conseguindo chamar direto qualquer método da classe e também manipulando alguns campos da classe.

Os métodos static tem um relacionamento com uma classe como um todo (...)"¹

¹ Trabalhando com métodos em Java. Disponível:

//<<https://www.devmedia.com.br/trabalhando-com-metodos-em-java/25917>>.

Estas possibilidades encontradas pelo uso de métodos, foram essenciais para a estruturação e organização do código, e sendo de grande importância para futuras manutenções e incrementos nele.

3.2.1.2-Switch:

Utilizamos a estrutura do switch durante o código quando precisávamos da interação de escolha do usuário para seguir o fluxo dos processos do código. Assim, o switch permite ao usuário ver em tela as opções que são os (case) e escolher qual instrução que deseja executar. Puga (2016, pág. 52) explica a estrutura do case ao afirmar que :

```
Java:
1.  switch (variavel)
2.  {
3.      case valor_1:
4.          <conjunto de instruções A>;
5.          break;
6.      case valor_2:
7.          <conjunto de instruções B>;
8.          break;
9.      case valor_3:
10.         <conjunto de instruções C>;
11.         break;
12.     default:
13.         <conjunto de instruções D>;
14. }
```

“Na linha 1, o comando switch é responsável pelo início do conjunto de opções para escolha, delimitado pela primeira chave, aberta na linha 2 e fechada na linha 14. Cada opção é apresentada na sequência, com a instrução case, nas linhas 3, 6 e 9, seguida pelo valor que será comparado com o armazenado na variável informada (variável). Quando uma comparação resulta verdadeiro, o conjunto de instruções associado é realizado, e este pode ter uma sequência de comandos delimitados por chaves. Observe que, para cada case, há um break, instrução responsável por provocar o encerramento do switch, desviando o fluxo do processamento para o final do laço de escolha (linha 14). Caso nenhuma comparação resulte verdadeiro, o conjunto de operações da opção default é executado. A palavra reservada break é utilizada na linguagem Java para garantir que apenas a instrução selecionada seja executada.”

Essa estrutura foi essencial no código para que o usuário pudesse ter uma interação com software e obter realização da ação desejada no software relacionadas a compra do ingresso, podendo assim escolher a peça, a data e a área desejada.

3.2.1.3-Vetor:

Assim, como também a estrutura do Vetor foi essencial para guardarmos informações e podemos relacioná-la com outras informações no código. O vetor é uma variável homogênea (dados do mesmo tipo) unidimensional (única dimensão), e que são conjuntos de variáveis do mesmo tipo, que possuem o mesmo identificador (nome) e são alocadas sequencialmente na memória (Puga, pág. 74, 2016) .

O vetor é representado por um identificador que é nome do vetor, o par de colchetes ([]) que indica que ele é um vetor, o tamanho dele que é uma quantidade de “elementos” (dados) guardados e o tipo do dado que pode ser um int, float, double, string ou char.

Sintaxe:

tipo identificador[]= new tipo[dimensão];

E, podemos manipular ou acessar os valores alocados nos espaços do vetor, através dos índices do vetor. Então, o índice indica a posição do dado no vetor, dada em forma de números (int), sequencial que começa sempre pelo número 0, ou seja, começa pela posição

zero. Por exemplo, imagem abaixo, com o nome do vetor temp, de tamanho de dez espaços, que na posição de número 0, irá receber um valor 18 a ser colocado neste espaço, e assim, sucessivamente.

índice:	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
nome: temp	18	17	20	26	32	29	15	12	16	21

nome- índice- valor:

```
temp[0]= 18;
temp[1]=17;
temp[2]=20;
(...).
```

E, para colocarmos valores dentro de cada espaço indicado pelo índice, podemos utilizar uma estrutura de repetição, como por exemplo o “**for**”, que possui o seu valor inicial, que aplica o código a cada medida que for solicitado por um contador (i++ ou i--), na condição e tamanho definido. Então, por exemplo no caso anterior o for é da seguinte forma:

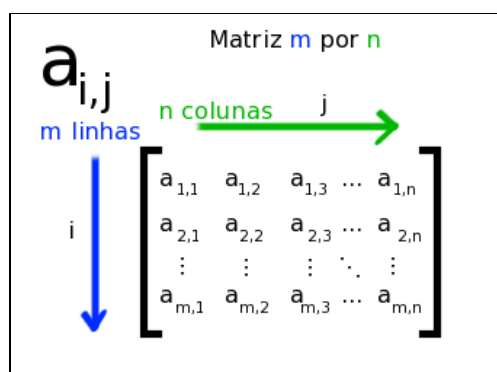
```
for (i=0;i<9;i++) {
temp[i]=entrada.nextInt();
System.out.println(temp);
}
```

Então, vimos as estruturas principais de um vetor e como podemos acessá-lo ou acrescentar informações, através do índice. Que utilizamos muito ao longo do código para manipular os dados de informação. E, associado aos vetores também utilizamos o sistema de matrizes.

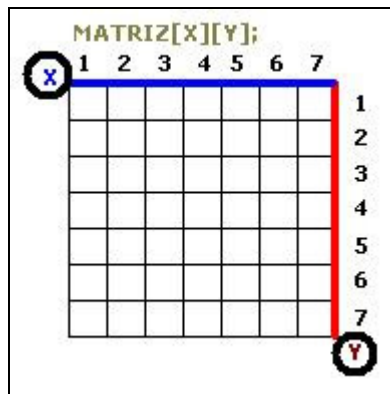
3.2.1.4-Matrizes:

As matrizes assim como os vetores foram fundamentais no código pois pudemos relacionar dados (data, área, peça e valor do ingresso) e obter novas informações (quantidade poltronas ocupadas, disponíveis, valor total dos ingressos, entre outros) e atribuindo a estas informações funções relativas a estatística de vendas dos ingressos do teatro.

Uma matriz, portanto, possui um conjunto de variáveis de um mesmo tipo, alocadas em uma sequência de memória, e que necessitam de mais de um índice para identificar um de seus elementos na sua estrutura, e que não há limite de índices, ficando a cargo do programador utilizar a quantidade de índices que considera necessário (Pugi, pág. 90, 2016).



Posições da matriz.



Índices da matriz.

A declaração de uma matriz é semelhante de um vetor, pois o vetor é uma matriz de uma dimensão. Assim, ao declarar por exemplo, uma matriz de duas dimensões na linguagem de programação, o tamanho dela é declarada da seguinte forma, por exemplo:

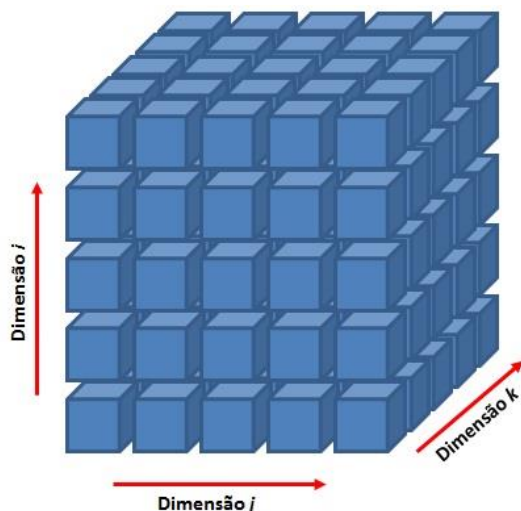
```
int[ ][ ] sessaoOcupacaoEntreOsDias = new int[3][3];
```

Neste caso, ela é uma matriz inteira, com o nome chamado "sessaoOcupacaoEntreOsDias" que tem o tamanho de três linhas e três colunas, onde o primeiro colchete se refere a quantidade de linha e o segundo a quantidade de colunas. E, quando queremos declarar uma matriz de três dimensões, acrescentamos mais um colchete que vai representar a profundidade desta matriz, então, ficaria desta forma, por exemplo:

```
int[ ][ ][ ] disponiveis = new int[3][3][14];
```

Uma matriz de tipo inteiro, com o nome "disponível", tendo três linhas, três colunas e 14 espaços de profundidade.

Matrizes de três dimensões:



Quando queremos atribuir valor dentro da matriz, utilizamos de índices, que são referências de localização dentro da estrutura da matriz, assim, se uma matriz tiver duas dimensões, ela necessitará de dois índices e se tiver três dimensões, três índices, para acessar a “célula” específica. E, para colocarmos um valor dentro desta célula de uma posição específica, utilizamos a seguinte denominação, por exemplo:

y [0] [3] [1]= 12.

Neste exemplo, a matriz y de três dimensões, recebeu o valor 12 na posição identificada pelos índices 0 (1ºlinha), 3 (4ºcoluna) e 1 (2º profundidade).

E ao preencher os valores em uma matriz, podemos utilizar uma estrutura de repetição (For) para preencher cada dimensão da matriz. Assim, podemos fazer a seguinte estrutura como exemplo de declaração, no qual no primeiro for irá preencher os dados da linha, no segundo for irá preencher os dados na coluna e dentro do bloco de códigos das interações, irá preencher a célula:

```
int x[ ][ ]= new int[7][3];
Scanner e=new Scanner (System.in);
for (i= 0; i<7; i++) { // preenche a linha.
    for (j= 0; j<3; j++) { // preenche a coluna
        x[i][j]=e.nextInt(); // preenche a célula
    }
}
```

Assim, a estrutura de matrizes nos permitiu ter uma maior facilidade para relacionarmos os dados necessários às funções determinadas do software e atingirmos os requisitos do software.

3.3 Fundamentos da Arquitetura de Computadores:
Levantamento dos Softwares e Hardwares utilizados para construção do programa:

Tabela de custo do Hardware			
Nome do Aluno	Gabriel Cessel	Joyce	Letycia
Sistema Operacional	Windows 10 R\$64,90	Windows 10 R\$64,90	Windows 10 R\$64,90
Processador	Intel core i5-10400F: R\$1.219,00	Ryzen 5 Ryzen 5 3500u: R\$1.933,78	Intel core 2 quad q8400: R\$150,00
Memória	8GB: R\$329,90	8GB: R\$R\$ 321,30	8GB R\$220,00
Placa mãe	Micro-Star International H310M PRO-VDH PLUS R\$409,9	Lenovo LNVNB161216: R\$ 800,00	PEGATRON CORPORATION MODELO: 2AAB R\$300,00
HD	R\$ 365,00	R\$ 312,48	R\$ 125,00
Placa de video	Galax GeForce GTX 1660 Super: R\$3.499,00	AMD Radeon™ Vega 8: R\$ 4.800,00	Rx550: R\$500,00
Valor Total	R\$ 5.822,80	R\$ 8.167,56	R\$ 1.295,00

Fonte: Letycia Souza,2021

Tabela de Custo de Software				
Nome do Aluno	Nome do Software	Nome do Software	Comunicação	Valor Total
Gabriel Cessel	NetBeans_R\$00,00	Windows 10 R\$64,90	Goole Meet R\$00,00	R\$ 64,90
Joyce	NetBeans_R\$00,00	Windows 10 R\$64,90	Goole Meet R\$00,00	R\$ 64,90
Letycia	NetBeans_R\$00,00	Pacote Office R\$50,00	Goole Meet R\$00,00	R\$ 50,00

Fonte: Letycia Souza,2021

3.4 Lógica Matemática e Estatística:

Na codificação do software, tivemos que recorrer várias vezes a conceitos matemáticos como os conectivos lógicos, igualdade de conjuntos, matrizes e matemática estatística.

Assim, utilizamos de conectivo lógico de conjunção, associados ao conceito de igualdade de conjuntos quando queríamos colocar uma estrutura de confirmação (booleana-verdadeiro ou falso) de um conjunto ser semelhante ao outro .

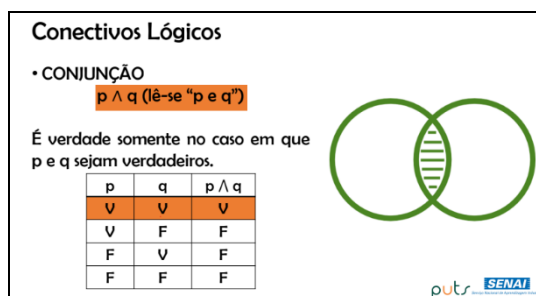


Imagem do slide da aula sobre "Noções Lógicas" do professor Ujeverson- SENAI/FATESG.

código:

```
static boolean confirmar() {  
  
    Scanner leia = new Scanner(System.in);  
    System.out.println("Confirma:");  
    System.out.println("1-Sim");  
    System.out.println("2-Não");  
  
    int codigo = leia.nextInt();  
    if (codigo == 1) {  
        return true;  
    }  
    return false;  
}
```

```
static String obterData() {
```

```
    // Primeiro, irá mostrar na tela as opções de datas. Que usuário irá digitar  
    //o número referente a data desejada.  
    do {  
        System.out.println("-----");  
        System.out.println("Escolha a data:");  
        System.out.println(" ");  
        System.out.println("Opção - Data:");
```



```

System.out.println("1    - 03/12/2021");
System.out.println("2    - 04/12/2021");
System.out.println("3    - 05/12/2021");
Scanner leia = new Scanner(System.in);
int opcao = leia.nextInt();

// Se o número digitado for igual ao condição "opcao" dos case (1, 2 ou
//3). Irá retornar a opção escolhida em forma de string e depois irá
//solicitar que confirme a informação escolhida pela booleana (confirmar ()).

// MATEMÁTICA: Estrutura de igualdade de conjuntos, onde dois conjuntos são
comparados
// se possuem o mesmo valor referente e conectivos lógicos de conjunção.
// em que só será verdade quando os elementos relativos forem verdade.
//Ou seja, se as duas opções referentes de data são iguais (Verdadeiro ) e a
//confirmação for Verdadeiro então é Verdade, o dado é armazenado e segue para
//outras partes do código. Se as datas forem iguais (verdadeiro) e se a confirmação
//for falso, então será falso. Retorna novamente ao loop de repetição.
//para sair da condição de loop (while) é se ~(0<opcao<=3 ^ confirmar()).

if (confirmar()) {
    switch (opcao) {
        case 1:
            return "03/12/2021";
        case 2:
            return "04/12/2021";
        case 3:
            return "05/12/2021";
    }
}

} while (true);

}

```

Também, utilizamos de matrizes para efetuarmos os cálculos estatísticos, pois tinha muitos elementos que se relacionavam entre si, as vezes estavam separados por categorias de outros elementos, então para relacionarmos e até obtermos novos elementos, tivemos que relacionar as informações de CPF, dia, peça, área, quantidade de ingressos, quantidade de poltronas ocupadas e disponíveis em matrizes.

Assim, em estruturas de matrizes para obtermos os valores de ingresso (multiplicando o valor do ingresso da área referente pela quantidade de ingressos comprados); para calcular a peça que teve mais ingressos vendidos (somando os ingressos vendidos, de todos os dias e todas as sessões, de cada peça e, retornando a peça que teve mais ingressos e menos ingressos vendidos do total de ingressos vendidos de cada peça); a peça que teve maior e menor ocupação de poltronas e de que dia (ao somarmos a quantidade de ingressos vendidos de cada sessão, por dia e por horário); a peça mais e menos lucrativa (ao somarmos os valores dos ingressos - quantidade de ingressos multiplicado pelo o valor da área específica- de cada peça, de todos os dias e compararmos e mostrarmos a que teve mais e menos lucro); e no

lucro médio do teatro das peças, ao somarmos o preço de todos os ingressos vendidos, de todas as peças e de todos os dias e dividimos pela quantidade de peças.

4. Resultados e Conclusão:

Os Resultados alcançados com este projeto foram, a elaboração de um software de venda para ingressos para o teatro, atingindo os objetivos de construir a função de compra do ingresso, de mostrar esse ingresso comprado na tela do usuário, e o relatório de vendas dos ingressos, todos arquitetados dentro do software. Assim, destacamos como vantagens deste projeto, a prática de todos os conhecimentos aprendidos durante o curso, e com um produto que nos capacita para o mercado de trabalho. Assim, como também a prática do trabalho em um grupo, no qual se buscou simular um ambiente de trabalho, em que foram definidas funções, estabelecidas atividades e atribuídas, entre os membros. Dificuldades Encontradas: Durante a elaboração do projeto encontramos dificuldade para organização da equipe quanto às reuniões e encontros presenciais. Ensinaamentos Colhidos: Um impacto científico que colhemos foi a melhoria na qualidade de vida que nosso software pode ofertar para as pessoas e organizações, trazendo por exemplo, otimização do tempo.

Outro impacto positivo foi em relação ao meio ambiente evitando o desperdício de recursos econômicos com gastos em papel impresso, que também promove a poluição. Vantagem de cunho Social em decréscimo nas horas trabalhadas e até mesmo a redução de funcionários. Acrescentando a precisão nos relatórios e fluxo de caixa para melhor controle financeiro do Teatro.

Encontramos no fim do projeto uma desvantagem, que ao darmos um upgrade no software poderá ser solucionada. Pois não demos a possibilidade do usuário escolher a numeração da sua cadeira.

A vantagem, é que, da forma que o programa foi logicamente pensado, e que o código foi escrito, será possível encontrar com fácil acesso a localização da alteração a ser efetuada para o upgrade do código.

E que quanto ao mercado para nós, desenvolvedores, enquanto continuarmos realizando bons projetos, que, auxiliam e facilitam a vida de organizações e também das pessoas, nós estaremos sempre a um passo à frente, e rumo ao futuro.

Pensando em conjunto foi sugerido pela equipe Java Café, um projeto futuro voltado à gerencia de um Centro de Estética onde será possível a organização da agenda, e horários dos clientes.

5. Referências

Gráfico de GANTT. Disponível em: google.ganttter.com Acesso em: 22/11/2021

Puga, Sandra; Rissetti, Gerson. Lógica de programação e estruturas de dados, com aplicações em Java.3. ed. - São Paulo : Pearson Education do Brasil, 2016.

Sites:

Ingresso Digital. Disponível em :

<https://ingressodigital.com/evento/3615/Afonso_Padilha_Show_Solo>. Acesso em 28/11/2021.

Trabalhando com métodos em Java.

Disponível:<<https://www.devmedia.com.br/trabalhando-com-metodos-em-java/25917>>.

Métodos em Java. Disponível em: <<https://pt.wikibooks.org/wiki/Java/M%C3%A9todos>>. Acesso no dia 29/11/2021.

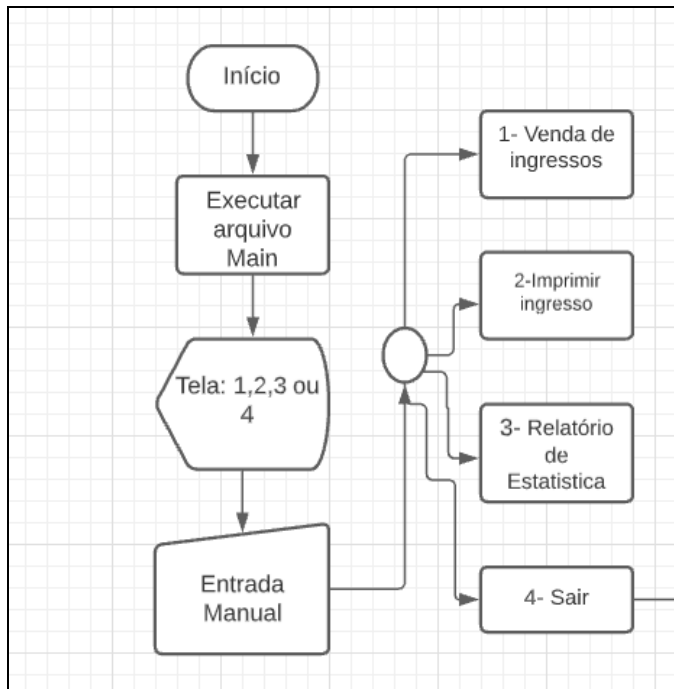
Lucidapp- site usado para o protótipo- Disponível em : <https://lucid.app/lucidchart/947da7ee-32d3-45ce-b6be-af2231157ffc/edit?from_docslist=true&invitationId=inv_a4c025f5-1b6a-4141-9221-c6833bd7c2c5&page=0_0#>. Acesso em 01/12/2021.

6. Anexos:

Protótipo:

O nosso software seguiu uma ordem condizente com os requerimentos solicitados pelo requerente do projeto buscando atingir todos os objetivos solicitados por este. Desta forma buscamos utilizar um fluxograma do site Lucid App para explicar cada funcionalidade de cada tela do software.

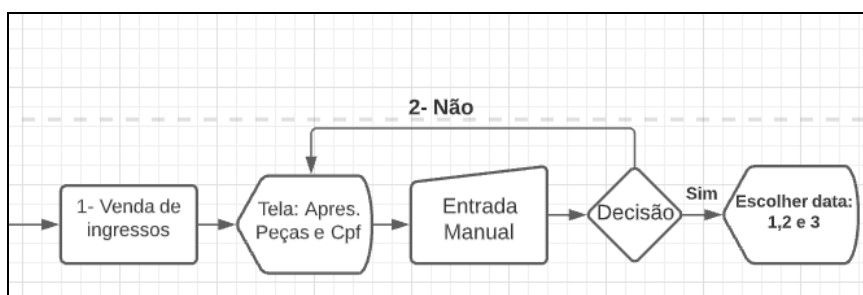
1º- Tela :



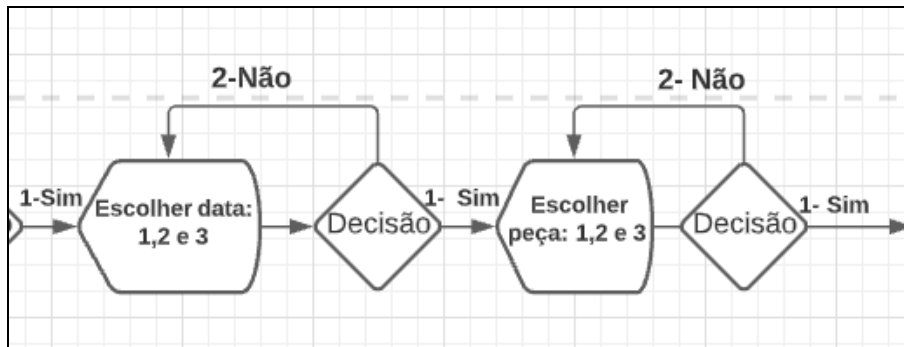
Protótipo- Tela inicial (1º tela). Fonte: Joyce, 2021

Nesta primeira tela, o programa irá executar o arquivo inicial, o Main.java, que mostrará na tela do usuário uma apresentação com o nome do teatro e as opções que o usuário gostaria que o programa executasse, que são : opção 1- Venda de ingressos; opção 2- Imprimir ingresso; opção 3- Relatório de Estatística e opção 4- Sair. Nesta última opção, o programa irá encerrar, e todos os dados de ingressos comprados serão zerados.

Se a opção escolhida for a 1- Venda de ingressos, abrirá a 2ª tela, que apresentará as peças em cartaz, com os horários e datas, e solicitando para o usuário digitar o CPF e confirmar. Para confirmação, que estará em todas as etapas do software, consiste do usuário certificar que está correta a informação digitada ou escolhida pelo usuário. Então, para confirmar que sim, ele irá digitar o número 1, e para confirmar que não, ele irá digitar o número 2, que neste caso, irá retornar a opção anterior de escolha na tela e novamente irá apresentar a confirmação. Assim, ao confirmar que o CPF está correto, ele irá para a 3ª tela deste fluxo, que é a tela referente a escolha da data da peça.



Protótipo- Segunda Tela do fluxo de comprar o ingresso, na parte digitar e confirmar o CPF. Fonte: Joyce, 2021



Protótipo- Terceira tela do fluxo de compra do ingresso, na parte de escolher a data, e a quarta tela para escolher a peça. Fonte: Joyce, 2021

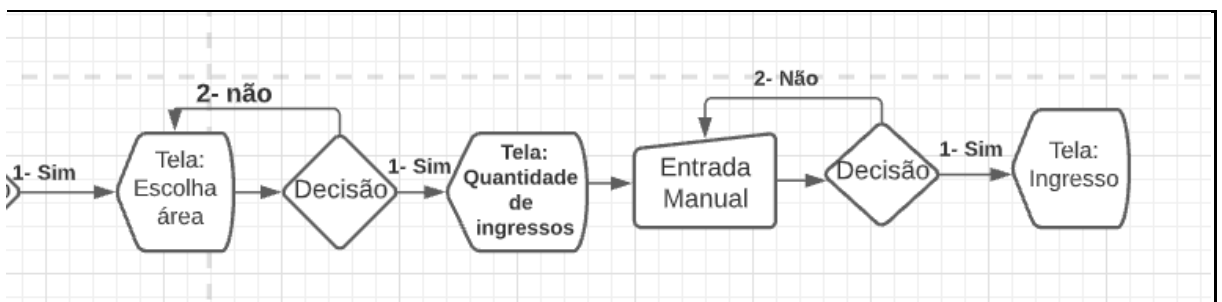
Que ao confirmar a data da peça escolhida, seguirá para a quarta tela para a escolha da peça teatral, que o usuário irá digitar o número referente a opção da peça escolhida, e confirmar. Após, isto, irá para a quinta tela para escolher a área do teatro, referente a localização da poltrona que deseja assistir a peça, nesta tela, constam informações sobre quais áreas, o preço unitário de cada área e informações sobre a quantidade de poltronas vagas. Então, após digitar o número referente a área que deseja comprar o ingresso, e confirmar.

```

-----
Escolha a área:

Opção - Área          - Valor      - Poltronas vagas:
1      - Plateia A     - R$ 40,00 - 25
2      - Plateia B     - R$ 60,00 - 100
3      - Frisa 01      - R$ 120,00 - 5
4      - Frisa 02      - R$ 120,00 - 5
5      - Frisa 03      - R$ 120,00 - 5
6      - Frisa 04      - R$ 120,00 - 5
7      - Frisa 05      - R$ 120,00 - 5
8      - Frisa 06      - R$ 120,00 - 5
9      - Camarote 01   - R$ 80,00 - 10
10     - Camarote 02   - R$ 80,00 - 10
11     - Camarote 03   - R$ 80,00 - 10
12     - Camarote 04   - R$ 80,00 - 10
13     - Camarote 05   - R$ 80,00 - 10
14     - Balcão Nobre  - R$ 250,00 - 50
14
Confirma:
1-Sim
2-Não
1
-----
  
```

Imagem da quinta tela do software, referente a escolha da área. Fonte: Joyce, 2021.



Protótipo- Quinta tela do fluxo de compra do ingresso, na parte de escolher a área, e

a sexta tela para digitar a quantidade de ingressos, e sétima tela, mostrando o ingresso comprado. Fonte: Protótipo, Joyce, 2021.

Após isso, abrirá a sexta tela para que o usuário possa digitar a quantidade de ingressos que deseja comprar para aquela área escolhida, e ao confirmar este valor, irá aparecer a última tela, a sétima, que aparecerá a imagem do ingresso comprado, constando todas as informações de: data, peça, horário, área, quantidade de ingressos, o valor total do ingresso, o nome do teatro e a nossa empresa que fez o software (Java-café), finalizando o fluxo do item de comprar ingresso.

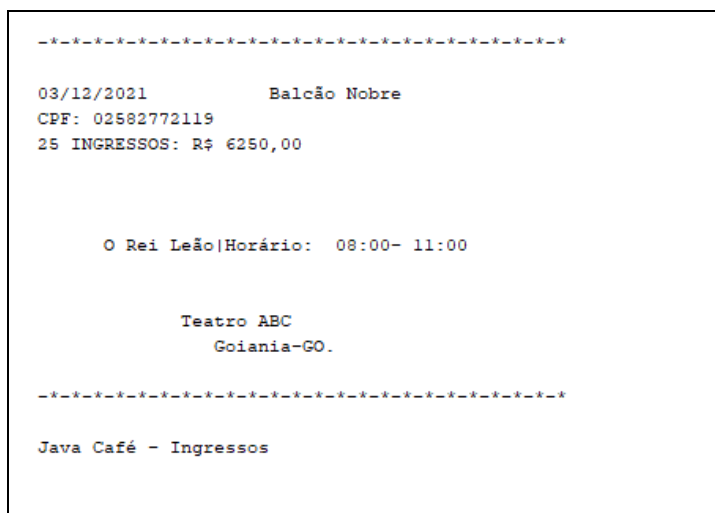
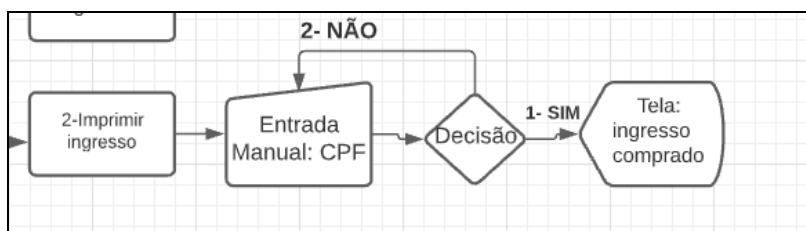


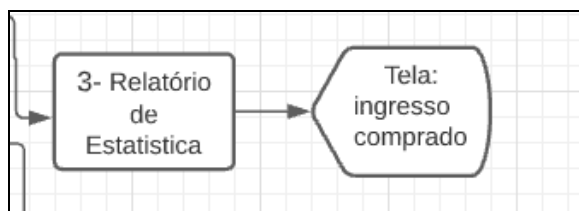
Imagem do ingresso comprado na tela do software. Fonte: Joyce, 2021

Já na opção para imprimir o ingresso, o usuário terá que digitar o número do CPF que foi comprado o ingresso, e após isto, abrirá uma tela com as informações do ingresso comprado.



Protótipo-Opção de "Imprimir o ingresso", terá uma tela para digitar e confirmar o CPF, e aparecerá a próxima tela com informações do ingresso comprado. Fonte: Joyce, 2021

E, na penúltima opção, que é se trata de mostrar o "Relatório de Estatística", abrirá uma tela com as informações referentes a estatística de vendas de ingresso no Teatro.



Protótipo- Fluxo de solicitar o Relatório de Estatísticas de vendas dos ingressos. Fonte: Joyce, 2021

```
-----:
ESTATÍSTICA DE VENDAS:
-----

Peça com ingressos mais vendidos e menos vendidos:

Peça com ingressos mais vendidos: Rei Leão - 25 ingressos vendidos
Peça com ingressos menos vendidos: Stand Up- Os parças! - 0 ingressos vendidos
-----

Sessão mais e menos ocupada:

Sessão mais ocupada: Rei Leão 03/12/2021 - 25 poltronas ocupadas.
Sessão menos ocupada: Stand Up- Os parças! 03/12/2021 - 0 poltronas ocupadas.
-----

Sessão mais e menos lucrativa:

Sessão mais lucrativa: Rei Leão - 03/12/2021 - R$ 6250,00 reais.
Sessão menos lucrativa: Stand Up- Os parças! - 03/12/2021 - R$ 0,00 reais.
-----

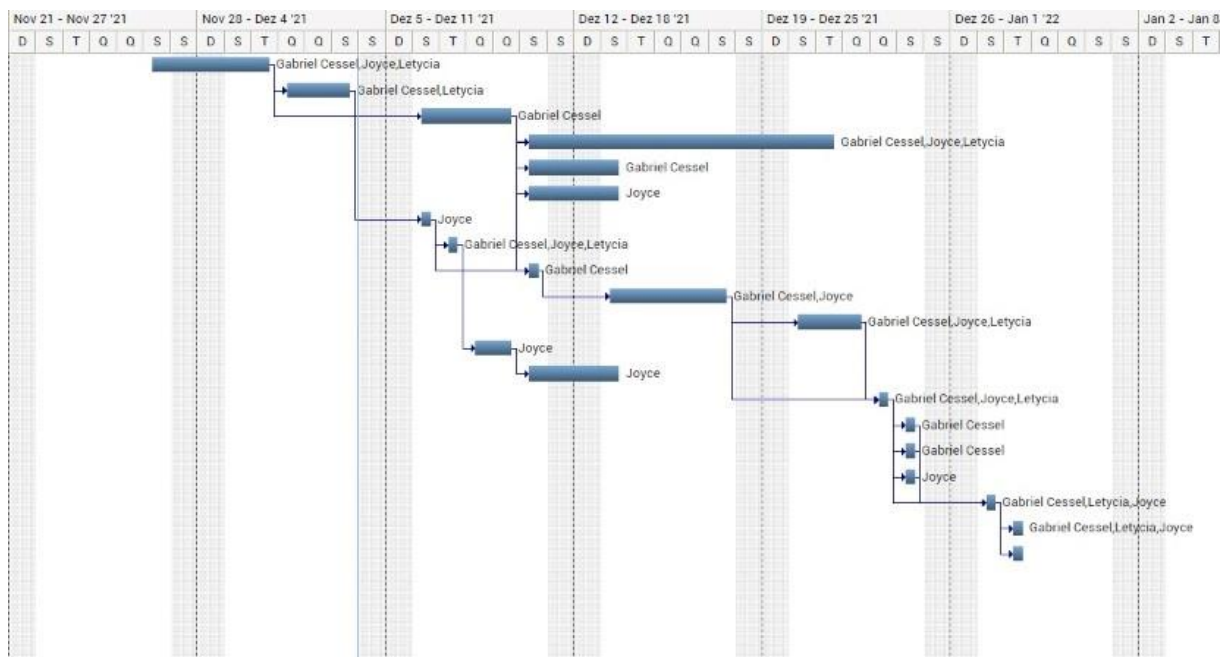
Lucro Médio do Teatro:
```

Imagem do software do Relatório de Estatística de vendas. Fonte: Joyce, 2021

6.1 -Gráfico Gantt

		Nome	Duração	Início	Fim	Predecessores	Recursos	Custom 1	Custom 2
1		Reunião	3 dias	11/26/2021	11/30/2021		Gabriel Cessel, Joy		
2		Levantamento de requisitos	3 dias	12/01/2021	12/03/2021	1	Gabriel Cessel, Let		
3		Escolha da Metodologia ágil a ser usada	4 dias	12/06/2021	12/09/2021	1, 2	Gabriel Cessel		
4		Gráfico de Gantt	8 dias	12/10/2021	12/21/2021	3	Gabriel Cessel, Joy		
5		Relatório descritivo do Software e Hardware usados	2 dias	12/10/2021	12/13/2021	3	Gabriel Cessel		
6		Termo de abertura PI	2 dias	12/10/2021	12/13/2021	3	Joyce		
7		Funcionalidade 1 (Validação do CPF)	1 dia	12/06/2021	12/06/2021	2	Joyce		
8		Funcionalidade 2 (Escolher data/sessão)	1 dia	12/07/2021	12/07/2021	7	Gabriel Cessel, Joy		
9		Reunião	1 dia	12/10/2021	12/10/2021	3, 7, 8	Gabriel Cessel		
10		Teste das funcionalidades	5 dias	12/13/2021	12/17/2021	9	Gabriel Cessel, Joy		
11		Reunião	3 dias	12/20/2021	12/22/2021	10	Gabriel Cessel, Joy		
12		Funcionamento 3 (Escolher Área e imprimir ingresso)	2 dias	12/08/2021	12/09/2021	8	Joyce		
13		Funcionalidade 4 (Estatística de vendas)	2 dias	12/10/2021	12/13/2021	12	Joyce		
14		Reunião	1 dia	12/23/2021	12/23/2021	10, 11	Gabriel Cessel, Joy		
15		Ajustes no termo de abertura	1 dia	12/24/2021	12/24/2021	14	Gabriel Cessel		
16		Ajustes no gráfico de Gantt	1 dia	12/24/2021	12/24/2021	14	Gabriel Cessel		
17		Explicação das funcionalidades do Software	1 dia	12/24/2021	12/24/2021	14	Joyce		
18		Reunião	1 dia	12/27/2021	12/27/2021	14, 15, 16, 17	Gabriel Cessel, Let		
19		Teste das funcionalidades	1 dia	12/28/2021	12/28/2021	18	Gabriel Cessel, Let		
20		Alterações no relatório descritivo do Software e Hardware usados	1 dia	12/28/2021	12/28/2021	18			

Fonte: print de tela de Gabriel Cessel, 2021



Fonte: print de tela de Gabriel Cessel, 2021

