

# Sistema de Venda de Ingressos

Lucas Oliveira da Silva

Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS)  
Av. Transnordestina, S/N. Feira de Santana, Novo Horizonte – BA, Brasil – 44036-900.  
lucasoliveiradaailva@gmail.com

**Resumo.** *Na organização de um evento musical tem sido difícil para o Diretório Acadêmico de Engenharia de Computação gerenciar o controle de venda dos ingressos. Tendo em vista essa problemática, os alunos da disciplina MI algoritmos, foram solicitados para elaborar um sistema de venda de ingressos, que possa ser reutilizado e que gerencie os dados de vendas automaticamente. Portanto, este relatório traz uma breve explicação do processo de elaboração desse sistema: metodologia adotada, dificuldades ao longo da elaboração, bem como os resultados e possíveis melhorias.*

## 1. Introdução

### 1.1 Resumo do problema

Na busca por recursos financeiros, os Diretórios Acadêmicos exploram abordagens como vendas de camisetas, sorteios, bingos etc. Apesar dos esforços, apenas uma forma de evento demonstra ser mais lucrativa em relação as outras: a realização de eventos musicais. A previsão de faturamento em eventos no Brasil para 2024, de acordo com dados da Associação Brasileira dos Promotores de Eventos (Abrape), é expressiva: 75 bilhões de reais. Sabendo disso, o Diretório Acadêmico de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS), planeja realizar um evento musical no bairro Feira VI.

Durante a organização do evento, o Diretório firmou parcerias para auxiliar nas vendas. No entanto, seria necessário um sistema para gerenciar as vendas dos ingressos.

Nesse contexto, os alunos da disciplina MI algoritmos foram convidados para elaborar, individualmente, um sistema de venda de ingressos para gerenciar as vendas e que posteriormente possa ser reutilizado em outros eventos. O software deveria atender os seguintes requisitos básicos:

- 1.º. Possui uma tela de configurações iniciais;
- 2.º. Realizar as vendas dos diferentes tipos de ingressos;
- 3.º. Apresentar as estatísticas das vendas ao finalizar o sistema;

## **1.2 Descrição breve da solução**

O sistema, desenvolvido na linguagem Python, vai ser projetado para executar pequenas partes de cada vez, dividindo uma grande parte em pedaços menores. O sistema terá como base:

- Variáveis para armazenar e processar todos os dados;
- Condicionais para operar o sistema conforme as escolhas do usuário;
- Estruturas de repetição para gerenciar o fluxo do sistema;

Ao iniciar o sistema, o usuário vai definir as configurações padrões do sistema, permitindo uma reutilização posterior. Logo após, o sistema irá rodar o loop central e registrar a venda conforme a preferência do usuário. Ao finalizar, o sistema deve realizar os cálculos para as saídas de dados, e exibir as estatísticas das vendas que ocorreram ao longo de sua execução no sistema. Nos próximos tópicos deste relatório serão abordadas os detalhes da elaboração do sistema, além dos resultados obtidos, limitações e possíveis melhorias.

## **2. Metodologia**

### **2.1 Construção de conhecimento nas sessões tutoriais**

O processo de elaboração do sistema teve escolhas de projeto, como a decisão de só haver três tipos de vendedores, sendo um padrão e dois convidados, estes recebendo as cortesias por vendas. Além de deixar por padrão somente quatro tipos de ingressos:

1. Inteira;
2. Meia entrada;
3. Alunos de Engenharia de Computação;
4. Diretório acadêmico ou Convidados;

Tendo em vista que os ingressos seriam doados para DA/convidados, o grupo resolveu não contá-los como ingressos mais vendidos, mas adicioná-los na média de idade dos compradores. Ademais, decidimos não definir uma idade mínima para entrar no evento, logo, o acesso seria livre para todos os públicos. Por fim, decidimos adicionar um botão de finalizar o sistema no lugar do limite de dias.

Foi escolha individual, o usuário definir com quantas vendas cada vendedor ganhará uma cortesia. Sob o mesmo viés, optei seguir uma estrutura de código com trechos curtos e bem definidos, isso porque facilitaria a manutenção e a leitura posterior. Além disso, para comprovar a compra do ingresso, elaborei uma mensagem ao final de cada compra, seguido de um “código do ingresso”.

### **2.2 Definição de requisitos**

Para atender os requisitos básicos foi necessário seguir algumas determinações

#### **2.2.1 Primeiro requisito básico:**

- Um menu de configurações iniciais;

### **2.2.2 Segundo requisito básico:**

- Apresentar quantos ingressos continuam disponíveis e quantos foram vendidos;
- Leitura do nome e idade;
- Caso o tipo for meia entrada, apresentar as devidas comprovações;
- Contar as cortesias para vendedores convidados;
- Encerrar as vendas quando os ingressos acabarem ou finalizar os dias de venda;

### **2.2.3 Terceiro requisito básico**

- Quantos ingressos foram e não foram emitidos;
- Quantos ingressos foram vendidos para cada tipo, bem como o valor arrecadado por cada um e o valor total;
- Mostrar a quantidade de cortesias e quantos ingressos foram vendidos por cada vendedor;
- Mostrar ingresso(s) mais vendido(s);
- Média de idade dos compradores;

## **2.3 Descrição de alto nível**

Primeiramente, foi necessário desenvolver o menu de configuração iniciais, onde o usuário define quais serão os vendedores, valores dos ingressos ect. Todas essas informações iram ser utilizadas quando iniciar as vendas.

Após as configurações iniciais, começa o laço principal, que opera enquanto o usuário não fechar as vendas e os ingressos não esgotarem. O sistema recebe as variáveis básicas: nome e idade. Em seguida, entra na tela para escolher o vendedor, ao selecionar, o sistema atualiza os dados do vendedor correspondente e continua o fluxo.

O sistema agora avança para a seleção do tipo de ingresso. Quando uma opção é selecionada, atribui “True” em uma variável para o tipo de ingresso escolhido. Se for meia, o programa entra em uma tela para selecionar o tipo de meia correspondente e configura sua variável correspondente. A mesma lógica se aplica ao selecionar “DA” e “Convidados”. O sistema pergunta se é “DA” ou “Convidado” e atribui a variável correspondente. Este processo de validação de variáveis foi necessário para manter a lógica de vendas separada da seleção do tipo de ingresso.

Seguindo, o sistema verifica o valor, verdadeiro ou falso, das variáveis e entra no bloco de código do tipo selecionado pelo usuário. Nessa parte ele apresenta qual será o valor do ingresso e atualiza os dados para o respectivo tipo de ingresso. Caso seja selecionado “meia-estudante”, o sistema pedirá o número de matrícula como forma de comprovação. Ainda sobe esse contexto, se for selecionado “meia-idoso”, a comprovação será a idade. Depois das devidas comprovações, o sistema segue o fluxo descrito anteriormente.

Após a venda, como forma de confirmação, o sistema gera um ingresso, conforme exemplo na Figura 1, mostrando o tipo selecionado e um código com número do ingresso vendido. Em seguida, ocorre a verificação de cortesia para vendedores convidados.

```
#####  
Venda realizada  
Tipo ingresso: Inteira  
Código: ECOMP-FEST10  
#####
```

**Figura 1. Ingresso gerado**

Por fim, o usuário é perguntado se quer finalizar as vendas, caso queira o sistema calcula as variáveis de saída e apresenta as estatísticas das vendas. Caso não queira, o sistema repete o loop principal.

## **2.4 Ordem de codificação**

O processo de desenvolvimento desse sistema durou cerca de três semanas. Na primeira semana, desenvolvi as entradas de dados e como ia funcionar o loop principal. Na segunda semana trabalhei na parte do processamento de dados, ou seja, a lógica por trás de cada tipo de vendedor, ingresso, as variáveis e tudo relacionado ao processamento. Na terceira semana implementei as saídas de dados, comecei desenvolvendo as variáveis que seriam usadas após a execução do sistema e depois desenvolvi as mensagens ao usuário. Por fim, testei e corrigir eventuais erros e bugs.

O sistema foi desenvolvido no sistema operacional Linux Mint. Com o interpretador Python 3.10.12, na IDE (Integrated Development Environment) Visual Studio Code Utilizando apenas funções nativas do Python.

## **3. Resultados e Discussões**

### **3.1 “Manual” de uso**

Para utilizar o Sistema de Venda de Ingressos, é necessário ter um interpretador do Python na versão mais recente para rodar o código. O arquivo pode ser executado por uma IDE (Integrated Development Environment). Vale ressaltar que, se o usuário fechar a execução, os dados serão perdidos e ele terá que configurar novamente.

### **3.2 Dados de entrada**

O sistema foi desenvolvido visando o “usuário perfeito”, isto é, ele não vai mentir e preencher todos os dados da forma correta, até porque o sistema não possui validação de entradas, caso algo seja digitado errado, o interpretador apresentará um erro.

De modo geral, a maioria das entradas funciona com menus enumerados para selecionar as opções, essa forma deixa o sistema menos vulnerável a entradas inadequadas, e opera de modo simples e intuitivo.

### **3.3 Dados de saídas**

O sistema apresenta todas as saídas de dados corretamente, alguns dados só podem ser definidos após a execução do programa, outros são obtidos ao longo do fluxo. Para as saídas serem exibidas, é necessário que o fluxo do sistema se encerre corretamente. Além disso, os valores estão formatados para a notação com vírgulas.

### **3.4 Testes**

Ao longo do desenvolvimento, diversos testes foram realizados com intuito de reconhecer limitações e erros do Software. Testes também foram realizados para conferir as saídas de dados, simulando diversos cenários de vendas. Além do teste das cortesias, se o sistema estava calculando corretamente quanto cada vendedor deveria ganhar.

### **3.5 Erros**

Como citado anteriormente, o sistema foi desenvolvido visando o usuário perfeito, logo, não possui o tratamento de erros — qualquer entrada digitada incorretamente o sistema apontará um erro, e consequentemente irá interromper a execução.

## **4. Conclusão**

Em síntese, o sistema funciona como uma venda real, onde ele vai perguntando e seguindo o fluxo passo a passo. Seu código é dividido em pequenas partes com uma tarefa bem definida, facilitando a organização. O sistema possui todos os requisitos básicos e específicos, além do extra de definir com quantas vendas cada vendedor irá ganhar uma cortesia.

A melhoria que pode acontecer é a utilização de outras estruturas de dados, tendo em vista a quantidade alarmante de variáveis utilizadas no sistema. Além disso, o programa poderia ter verificações para tratar os “usuários imperfeitos”, isto é, o tratamento para entradas inesperadas. E por fim, a refatoração do código em funções para melhorar a manutenção.

## **5. Referências**

- Os bilhões que o mercado de eventos deve faturar em 2024. VEJA, 2024. Disponível em: <<https://veja.abril.com.br/economia/os-bilhoes-que-o-mercado-de-eventos-deve-faturar-em-2024/>> Acesso em 19 de Abril de 2024.
- WAZLAWICK, R. S. (2018). Introdução a Algoritmos e Programação com Python. Elsevier.
- MANZANO, J. A. N. G.; OLIVEIRA, J. F. (1996). Algoritmos: Lógica para Desenvolvimento de Programação. Érica,