

# 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 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, venda de camisetas e canecas, sorteios, bingos. 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 uma parceria com alunos dos cursos de Enfermagem e Biologia para auxiliar nas vendas, propondo uma comissão ao realizarem 10 vendas. Mais tarde surgiu uma questão: como gerenciar as vendas dos ingressos? Nesse contexto, os alunos da disciplina MI algoritmos foram convidados a elaborar — individualmente — um sistema de venda de ingressos para gerenciar as vendas e que posteriormente possa ser reutilizado em outros eventos.

### 1.2 Descrição breve da solução

O sistema — desenvolvido em Python — foi projetado para ir executando pequenas partes de cada vez, dividindo uma grande parte em pedaços menores. Ele funciona por meio de variáveis para armazenar e processar todos os dados, condicionais para operar o sistema conforme as escolhas do usuário e loops para gerenciar o fluxo do sistema. Ao iniciar, o usuário define as configurações padrões do sistema — permitindo uma reutilização posterior. Logo após, o sistema roda o loop central e vai registrando as

vendas conforme as opções selecionadas pelo usuário até que o mesmo decida encerrar ou os ingressos disponíveis acabem. Ao finalizar, o sistema realiza os cálculos para as saídas de dados, e exibe 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 cominou em algumas 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 — inteira, meia, engenharia de computação, DA/convidado —, como DA/convidado estava definido como ingresso, e para eles os ingressos seriam doados, o grupo resolveu não conta-lós como ingressos mais vendidos — mas adiciona-lós na média de idade dos compradores. Ademais, decidimos não definir 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, na verdade, isso era um bônus do projeto, e achei viável adicionar. Sob o mesmo viés, foi uma escolha minha, tentar 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**

O sistema deveria atender os seguintes requisitos: ter um menu de configurações iniciais, 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. Além disso, apresentar os dados de saída na tela do usuário: 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. E por fim, o ingresso mais vendido — caso seja empate, apresentar todos — e a média de idade dos compradores.

### **2.3 Descrição de alto nível**

O desenvolvimento desse software seguiu três estágios: entrada de dados, processamento, e saída de dados. Iniciei pela entrada de dados, lendo a descrição do projeto era necessário que o código fosse reutilizado, portanto, antes de rodar as vendas, seria necessária uma tela de configurações iniciais, com os parâmetros necessários para o funcionamento do sistema como, por exemplo, quantidade de ingressos disponíveis, valores, vendedores. Todas as configurações necessárias para a reutilização do sistema.

Após as configurações iniciais, as vendas começam com o laço principal, que opera com uma variável booleana e a quantidade de ingressos disponíveis. O sistema recebe as variáveis básicas: nome e idade. Em seguida, entra na tela para escolher o vendedor. Enquanto o usuário não seleciona uma opção válida, o sistema permanece nessa tela. 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. Seguindo a estrutura anterior, o programa continua na seleção até que seja digitada uma opção válida. Quando uma opção é selecionada, valida 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, conforme a escolha, configura a variável correspondente. Este processo de validação de variáveis foi necessário para manter a lógica de vendas separada da seleção do tipo de ingresso, em outras palavras, foi por questões de organização.

Seguindo, o sistema verifica o estado — 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 sobre esse contexto, se for selecionado “meia-idoso”, a comprovação será a idade — novamente, para confirmar. Depois das devidas comprovações, o sistema segue o fluxo descrito anteriormente — mostra valor, atualiza dados.

Após a venda, como forma de confirmação, o sistema gera um ingresso, ver exemplo na Figura 1, mostrando o tipo selecionado e um código com número do ingresso vendido — a contagem é decrescente. Em seguida, o sistema verifica se tem uma nova cortesia para os vendedores, comparando as cortesias antes e depois da venda atual, se tiver uma nova, ele libera e subtrai dos ingressos disponíveis. Vale ressaltar que o sistema só libera caso ainda tenha ingressos acessíveis.

```
#####  
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, iniciando mais uma venda.

## **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, como já

havia estudado a linguagem antes, não passei muito tempo estudando a sintaxe, apenas revisei alguns aspectos lógicos. Na segunda semana trabalhei na parte do processamento de dados, isto é, a lógica por trás de cada tipo de vendedor, ingresso, as variáveis e tudo relacionado ao processamento. Cabe ressaltar que a segunda semana foi a que mais testei formas de desenvolvimento, visando sempre uma forma mais eficiente de desenvolvimento. Na terceira semana implementei as saídas de dados, comecei configurando as variáveis que seriam usadas após a execução do sistema e logo em seguida configurei as mensagens ao usuário. Por fim, fiquei até o final do prazo testando e corrigindo 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. Pode ser esse executado por uma IDE (Integrated Development Environment), ao instalar selecionar o arquivo e executar, logo após, o sistema irá rodar no terminal. Vale ressaltar que se o usuário fechar a execução, os dados serão perdidos e terá que configurar tudo 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, esta 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, sendo este o padrão aqui no Brasil.

#### **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 desde somente uma venda e todas as 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 descritos pelo problema, 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,