

Sistema de Venda de Ingressos

Adson Victor de S. Alves

UEFS – Universidade Estadual de Feira de Santana
Av. Transnordestina, s/n, Novo Horizonte
Feira de Santana – BA, Brasil – 44036-900

21uefs@gmail.com

Abstract. *The Academic Directory's Computer Engineering from Universidade Estadual de Feira de Santana wants to do an event for funds collection and for this, asked a system to help on selling tickets and that could be reused in the future. The system makes all the tickets vending processing, from ticket type selection, documents verification, registration and statistics update and the requested calculations. The elaboration of the program was used the Python language with Visual Studio Code. The program is loop based that will register all the sales until the end of the tickets or the vending period ends, processing all the input information and updating data. Finally, accomplish the final calculations and show collected statistics from vending period. The software answers the requested problem and the requirement of could be reused.*

Resumo. *O Diretório Acadêmico do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana deseja realizar um evento para arrecadação de fundos e para tal, solicitou um sistema para auxiliar nas vendas dos ingressos e que pudesse também ser reutilizado futuramente. O sistema realiza todo o processamento da venda, a partir da seleção do tipo de ingresso, comprovação de documentos, registros e atualizações estatísticas e os cálculos solicitados. Para elaboração do programa foi utilizado a linguagem Python a partir do Visual Studio Code. O programa é baseado em um loop que registrará todas as vendas até que os ingressos se esgotem ou o período para venda termine, processando todas as informações de entrada e atualizando os dados. Finalmente, realiza os cálculos finais e exibe as estatísticas coletadas a partir do período de venda. O software atende ao problema solicitado e também à necessidade de reutilizações futuras.*

1. Introdução

Controle e organização são conceitos importantes e a partir desses conceitos, quando pensamos em gestão de eventos, logo temos a ideia do quão pode ser difícil administrar todos dados, informações e logística por trás desses eventos. Pensando nisso, o Diretório Acadêmico (DA) de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS), ao elaborar um evento para arrecadação a partir da venda de ingressos para shows que seriam organizados também pelo DA, pensou também em formular um sistema que auxiliasse justamente com essa questão de controle e organização desse evento.

Para a gestão do evento, o sistema formulado deve indicar quantos ingressos foram vendidos e quantos ainda faltam vender, verificar o tipo de ingresso a ser vendido

e sua disponibilidade, registrar os devidos comprovantes caso sejam ingressos de meia entrada ou outro requisito, conferir a documentação e valores a partir de vendedores comissionados, que são vendedores de outros cursos que auxiliarão com a venda dos ingressos e em troca receberão cortesia que também deverão ser contabilizadas. Além disso, também devem ser registrados a quantidade de ingressos vendidos por tipo, a quantidade de ingressos vendidos por cada vendedor comissionado, o dinheiro arrecadado por tipo de ingresso, o dinheiro total arrecadado, o tipo de ingresso mais vendido e também a média das idades dos compradores dos ingressos. O ponto importante é que o sistema formulado deveria atender não somente a esse evento, como os futuros eventos que seriam promovidos, então faz-se necessário a possibilidade de ser reutilizado futuramente.

A solução é organizada a partir de um menu, que pode ser acessado por todos, seja o gestor do evento, para a configuração prévia ou adicional do sistema e que também permita que o sistema seja reutilizado futuramente, quanto pelos vendedores comissionados ou por aqueles que desejam, por si mesmo, comprarem seu próprio ingresso. A depender do tipo de acesso será solicitada uma senha para evitar que qualquer pessoa possa desconfigurar ou sabotar o sistema. A partir do acesso selecionado, poderá ser visualizado um outro menu para configuração ou visualização dos dados do evento em tempo real ou um menu para compra ou cadastro de um novo ingresso. Também, a depender do tipo de ingresso selecionado, será solicitado para o usuário as comprovações adequadas, e se tudo for inserido corretamente, o ingresso será devidamente vendido, os comprovantes registrados e os dados atualizados no sistema.

2. Desenvolvimento

Nesta seção será abordado sobre o desenvolvimento do sistema em relação as questões anteriores e como foram tratadas na solução do problema.

O código do sistema deu início a partir da criação das variáveis que seriam utilizadas posteriormente, como o número de ingressos totais, a precificação inicial dos ingressos, categorização dos tipos de ingresso, somatória das idades que seriam cadastradas, categorização das cortesias e ingressos que seriam vendidos pelos vendedores comissionados, as senhas de acesso, variáveis para inicialização do menu, definição da idade mínima para a compra do ingresso e também a categorização da arrecadação dos ingressos.

O sistema é iniciado a partir de um *loop while* com uma condicional de parada que estará sempre disponível para o usuário caso o mesmo deseje interromper o sistema. A partir desse *loop while* é exibido para o usuário o primeiro menu, no qual é informado as opções para níveis de acesso – gerenciador, que poderá configurar o sistema, vendedor, caso seja um vendedor comissionado, comprador caso o usuário apenas deseje comprar seu próprio ingresso e encerrar, para interromper o sistema.

Para a criação do menu foi utilizando apenas o comando *print()*, juntamente com formatações de *string* como o “\n”. E para confirmar a opção do usuário foi utilizando um *input()* que é temporariamente armazenada em uma variável para a realização das comparações com os itens do menu, a partir de condicionais *if else*. Essa é a base do código do sistema.

```

if manager_password_confirm == manager_password:

    print('\n===== SELECIONE UMA OPÇÃO =====\n')
    print('[1] - Configurar o sistema')
    print('[2] - Visualizar informações')
    print('[3] - Alterar senhas')
    print('[4] - Voltar\n')

    manager_option = str(input('Insira uma opção: '))

```

Figura 1. A figura mostra como foi realizada a organização do código para a criação do menu.

```

===== SELECIONE O TIPO DE ACESSO =====

[1] - Gerenciador
[2] - Vendedor
[3] - Comprador
[4] - Encerrar

Insira uma opção:

```

Figura 2. A figura retrata a visualização do usuário ao iniciar o sistema e ter acesso ao primeiro menu.

Na primeira opção – gerenciador – será solicitado para o usuário, a partir de um *input()*, uma senha de acesso predefinida e que pode ser alterada posteriormente. Após a confirmação da senha, o usuário terá mais opções, que estarão relacionadas principalmente a visualização dos dados do evento em tempo real e configuração do sistema para futuros eventos ou mesmo manutenção e/ou correção de alguma informação. Além disso, também terá a possibilidade de realizar a alteração das senhas e dos códigos necessários para a confirmação de determinados ingressos que serão falados posteriormente.

```

Insira a senha para continuar: y

===== SELECIONE UMA OPÇÃO =====

[1] - Configurar o sistema
[2] - Visualizar informações
[3] - Alterar senhas
[4] - Voltar

Insira uma opção: █

```

Figura 3. A figura mostra a solicitação da senha para o usuário, e caso inserida corretamente, iniciará o menu abaixo.

Todas as solicitações de opções e senhas são a partir de um *input()*, que por vezes serão tipados como *float* ou *int*, quando necessário. As senhas e opções solicitadas para o usuário são temporariamente armazenadas em variáveis para comparação, e caso seja correspondente, terá sequência para a próxima página. Caso contrário, será mostrada uma mensagem de erro e voltará para a tela inicial.

Na segunda opção da figura 2, será tratada a navegação para os usuários que fazem parte dos vendedores comissionados, ou seja, é a partir dessa opção que esses vendedores conseguirão realizar suas vendas, realizando o cadastro de documentos e dados dos compradores, bem como será registrada sua venda para disponibilização de suas cortesias posteriormente. Também será solicitado para o usuário uma senha, que por padrão todas são diferentes, e caso a senha seja correspondente, iniciará o menu que irá solicitar para o mesmo a categoria em que se enquadra. A divisão de categorias foi realizada por cursos, nos quais se enquadram os vendedores.

```
Insira a senha para continuar: x

===== TIPO DE COMISSÃO =====

[1] - Comissão Biologia
[2] - Comissão Enfermagem
[3] - Voltar
```

Figura 4. A figura mostra como é solicitado para o usuário a categoria em que se enquadra para que seja realizado os devidos registros.

Após escolher sua categoria, terá início o menu para a venda de ingressos, a partir do qual o vendedor fará o registro de suas vendas, a partir do ingresso que será vendido e suas devidas comprovações. Caso o vendedor confirme todos os dados solicitados pelo sistema corretamente, a compra do ingresso será computada e o vendedor também já terá a venda registrada internamente.

```
Insira uma das opções: 1

===== TIPO DE INGRESSO =====

[1] - Ingresso Inteira - R$30.00
[2] - Ingresso Estudante - R$15.00
[3] - Ingresso Idoso - R$15.00
[4] - Ingresso Promocional - R$10.00
[5] - Ingresso VIP
[6] - Voltar a tela anterior

Selecione o tipo de ingresso: █
```

Figura 5. A figura mostra a tela que se sucede após selecionar a categoria de vendedor ou após entrar como comprador.

Na terceira opção da figura 2 não será solicitado para o usuário nenhuma senha, pois trata-se da navegação para o caso do comprador optar por realizar por si mesmo a compra do ingresso. Após confirmar a opção, a visualização do usuário será também o

menu que é exibido na figura 5. O fluxo é o mesmo, porém sem registrar vendas para qualquer vendedor comissionado. É importante ressaltar que para essa navegação também serão solicitados os documentos e as devidas comprovações para a compra de cada tipo de ingresso.

2.1. Registros e comprovações

A seguir, será abordado como foram realizados os registros de ingressos adquiridos bem como todo o processo de comprovação dos dados solicitados.

Para todos os tipos de ingressos é solicitado para o usuário a idade do comprador, é sempre o primeiro dado solicitado, e caso não atenda aos requisitos de idade de cada tipo de ingresso ou ao mínimo do evento, a compra não será autorizada e o usuário será redirecionado ao menu inicial. É de extrema importância registrar as idades dos compradores pois serão necessárias para cálculos estatísticos posteriormente.

Para o ingresso do tipo Inteira não é solicitado nenhum outro dado além da idade do comprador. Caso a idade inserida seja maior ou igual ao mínimo do evento, a compra é autorizada.

Para o ingresso do tipo Estudante, além da idade, é solicitado também o número do seu documento ou carteira estudantil, que deve conter 09 dígitos e não pode ser incluídos letras ou caracteres. Para realizar a comprovação desse documento foi utilizada a função *len()* que contabilizará quantos caracteres tem uma *string*, e caso seja maior ou menor que 9, a compra não será autorizada. Para verificar se foram inseridos apenas dígitos, foi utilizado a função *isnumeric()*, que retorna o valor booleano *False* caso a *string* não contenha apenas números ou *True*, se de fato possuir apenas números.

Para o ingresso do tipo Idoso também não é solicitado nenhum outro dado além da idade. Para este tipo de ingresso a idade inserida deve ser maior ou igual a 60 anos.

Quanto para os ingressos do tipo VIP e Promocional, em ambos os casos, são solicitados para o usuário uma senha. As senhas solicitadas, a princípio, são diferentes para cada um dos tipos de ingresso. Sendo assim, é solicitado primeiramente as idades e posteriormente é solicitado essas senhas ou códigos para a confirmação da compra do ingresso.

As idades solicitadas são temporariamente armazenadas em uma variável e caso a compra do ingresso seja autorizada, essa idade é somada a uma outra variável que conterà, também, a somatória de todas as idades de compradores que tiveram ingressos autorizados até o momento. Todas as verificações, seja da idade, códigos, senhas ou documentos são feitas também a partir de uma condicional *if else*. Então, a partir do momento que os dados são inseridos corretamente, a compra é autorizada, é somado em 1 a quantidade de ingressos vendidos e atualizada a quantidade de ingressos disponíveis, que é contabilizado, a partir da subtração da quantidade de ingressos totais definidos pela quantidade de ingressos vendidos, contidos em variáveis definidas anteriormente. Conforme o tipo de ingresso selecionado, será somado em 1 também a quantidade daquele tipo de ingresso, novamente armazenada em uma variável específica definida anteriormente. Caso a venda ocorra a partir de um vendedor comissionado, a depender da categoria escolhida anteriormente, é somado em 1 também a quantidade de vendas por aquela categoria e mais uma vez armazenada em uma outra variável específica para contabilização posteriormente.

2.2. Resultados e visualização das estatísticas

Nesta seção busca-se abordar como são calculados e gerados os resultados, como é realizada a visualização dessas estatísticas no sistema, bem como essas informações estão organizadas.

As informações são sempre atualizadas a cada compra de ingresso que tenha sido autorizada. Essas informações iniciais são a quantidade de ingressos disponíveis, quantidade de ingressos vendidos até o momento e é feita a somatória da idade inserida ao total de idades que já haviam sido inseridas, o que permite que seja realizado o cálculo da média de idades do evento. Ou seja, todas as idades são somadas a cada compra e são divididas pelo total de termos que é representado pela quantidade de ingressos vendidos.

As cortesias são oferecidas aos vendedores comissionados a cada 10 ingressos vendidos por eles, então para esse cálculo, foi utilizado uma divisão inteira a partir do operador aritmético “/” da quantidade de ingressos vendidos por cada curso pelo valor inteiro, 10.

A fim de calcular a arrecadação por tipo, foi necessário apenas multiplicar a quantidade de cada tipo de ingresso por seu respectivo valor, o que permite também que possa ser calculado a arrecadação total apenas somando todos os valores arrecadados pelo tipo específico.

Para definir qual foi o ingresso mais vendido, foi utilizado a função *max()* que retorna o maior valor dentre uma lista, tupla ou uma base de valores fornecidos e um sistema de condicionais para determinar a qual tipo de ingresso o valor retornado pela função corresponde.

A visualização das estatísticas pode ser feita ao ser encerrado o sistema, bem como pode ser realizada em tempo real também durante o período de vendas.

2.3. Extras

Pensando em obter um maior controle das informações ou mesmo outras funcionalidades que retornassem mais praticidade, foram implementadas algumas outras funções.

A primeira funcionalidade extra é justamente a visualização das informações em tempo real, o que permite que a gestão do evento tenha um maior controle das vendas que estão ocorrendo, em qualquer momento que deseje.

Para a produção do sistema foi requerido que pudesse ser reutilizado em outros eventos futuramente, e pensando nisso, a partir da navegação de Gerenciador (pode ser visualizado nas figuras 2 e 3), é possível realizar alterações de informações como idade mínima, valor de cada tipo de ingresso e o número total de ingressos do evento, permitindo na apenas reutilizações futuras, mas manutenção durante qualquer evento que já esteja ocorrendo.

Outro ponto foi a verificação de um documento para comprovação da compra de determinado tipo de ingresso, a este nível, fictício, porém que já traz uma base para aprimoramentos futuros do sistema e para que seja, de fato, realizado o processo de checagem adequado.

2.4. Problemas, soluções e alternativas

É comum que durante o desenvolvimento de códigos e sistemas sejam encontrados alguns problemas e/ou falhas ao longo de todo o projeto. Nesse caso, não foi diferente. Nesta seção será tratado desses problemas, as soluções encontradas e possíveis alternativas.

No início do desenvolvimento do sistema, as informações de número total de ingressos e quantidade de ingressos disponíveis não eram previamente inseridos, pois a ideia era que fossem inseridos a partir da navegação do Gerenciador, no entanto, ao iniciar o sistema a partir da opção de Comprador ou Vendedor, o sistema parava, pois não haviam sido definidos a quantidade de ingressos, que era importante para o funcionamento do sistema. Para resolução do problema, atribuí previamente valores para as variáveis já no início do código.

Outro problema foi solicitaram as opções do usuário declarando a variável em que seria armazenada a opção como do tipo *int*, então caso o usuário digitasse qualquer outra coisa além de um valor inteiro, o sistema também logo era interrompido. Para correção, só alterei o tipo da variável de armazenamento para *string*, então qualquer valor inserido era válido, no entanto, caso não fossem correspondentes as opções disponíveis, era exibida uma mensagem de erro. Uma possível alternativa nesse caso seria a partir dos comandos *try* e *except*, a fim de testar primeira o valor e inserido e caso divergisse do tipo válido, seria também exibida uma mensagem de erro, porém para essa situação em específica não seria tão prático dado o grau de desenvolvimento que o sistema já apresentava.

Todavia, essa solução foi utilizada logo mais a frente, quando era solicitado para o usuário a idade. Então tem-se o mesmo problema, porém não era possível de seguir com a opção utilizada antes pois o valor inserido deveria ser somado, então precisaria ser um valor inteiro.

Durante a fase de testes do sistema, foi observado que os resultados da quantidade de ingressos emitidos, não emitidos, o valor da arrecadação de cada ingresso e arrecadação total não estavam sendo atualizados. O erro tratava-se apenas de falha semântica em que as variáveis não estavam declaradas dentro do bloco responsável por gerar e atualizar as estatísticas.

Quando o usuário solicitava visualizar as informações antes de qualquer ingresso ter sido vendido, o sistema era interrompido pois não conseguia calcular a média de idades, dado que seria uma divisão por zero, e para a resolução foi necessária apenas uma condicional que fez com que a divisão não ocorresse e ao invés, seria exibida apenas uma mensagem informando que não havia sido vendido ingressos e portanto não era possível calcular a média das idades.

Para informar o tipo de ingresso mais vendido, inicialmente foi utilizada uma série de condicionais, porém tinha um problema: quando 2 ou mais tipos de ingressos eram da mesma quantidade, o sistema não informava nada. A correção foi tratada a partir da utilização da função *max()* informada anteriormente.

3. Conclusão

Este relatório apresentou uma solução de software para venda de ingressos e coleta e análise de dados obtidos a partir de sua utilização em eventos. O sistema foi elaborado a partir do Visual Studio Code (VSCode), na versão 1.88.1 utilizando a linguagem

imperativa Python na versão 3.12.3, com interface a partir do próprio terminal e sendo os dados armazenados em memória. A solução compreende o problema proposto e algumas funcionalidades a mais que tornam sua utilização mais prática a partir da utilização de linhas de comandos imperativas estruturadas, como definição de variáveis, comandos de entrada e saída, estruturas de repetição, estruturas de condicionais, contadores e acumuladores e processamentos a partir de cálculos e atribuições.

Ao longo do relatório foram explicitados diversos pequenos problemas e suas respectivas soluções e possíveis alternativas. A solução centra-se em uma estrutura de repetição que registrará cada venda, prontamente com a atualização dos dados em caso do processamento da venda ser bem sucedido, com navegação a partir de comandos de entrada e saída formando uma interface e possibilidade a comunicação do usuário com o sistema e uma etapa final de cálculos adicionais e apresentação dos resultados e estatísticas calculadas.

O programa atende o problema proposto e às especificações solicitadas, bem como sua necessidade de reutilização em outras situações distintas que tratem de vendas de ingressos. Entretanto, apresenta uma grande limitação pelo fato de armazenar os dados em memória e pelo fato do desenvolvimento ser previamente limitado a partir de uma base de comandos específicos. Futuras versões podem adicionar mais flexibilidade e maior capacidade de reutilização, como a implementação de banco de dados para armazenamento de dados e arquivos e/ou cadastro de usuários também a partir de um banco de dados, tal como a utilização de uma interface gráfica mais elaborada, em um programa dedicado.

Este relatório apresenta robustez em sua estruturação dada a organização detalhada do problema em etapas, problemas, soluções e alternativas. Também é válido informar que pode haver problemas de redação que reduzam a clareza e a concisão, além de citações repetitivas no texto.

Referências

Fabício Bueno (2021) “Introdução a Programação com Python”.

John V. Guttag. Introduction to Computation and Programming Using Python. Revised and expanded edition. MIT Press, 2013.