

Festa na UEFS: Um Sistema em Python para Venda de Ingressos

Pedro Lucas F. de Souza

¹UEFS – Universidade Estadual de Feira de Santana, Av. Transnordestina, s/n,
Novo Horizonte, Feira de Santana – BA, Brasil – 44036-900

pedrofernandeseng.comp@gmail.com

Abstract. *This article describes and reports on the operation and development of software in Python language to calculate tickets sales for an event promoted by Academic Directory of the Computer Engineering course at Universidade Estadual de Feira de Santana (UEFS). The software computes the statistics relevant to the event, such as tickets sales, revenue, comission and average age. The user can configure the software quantity of tickets and the price of ticket types to be sold.*

Resumo. *Este artigo objetiva descrever e relatar o funcionamento e desenvolvimento de um software em linguagem Python para contabilizar as vendas de ingressos para um evento promovido pelo Diretório Acadêmico do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS). O software computa as estatísticas pertinentes ao evento, tal como venda de ingressos, arrecadação, comissão e média de idade. O usuário pode configurar o software por quantidade de ingressos e o preço dos tipos de ingressos a serem vendidos.*

1. Introdução

Na história da humanidade sempre houve a necessidade de contar e gerenciar objetos e recursos para os mais diversos fins. Ao longo do tempo, com o desenvolvimento das civilizações e formação dos centros urbanos, houve a exigência de elaborar métodos, procedimentos e algoritmos para contabilizar e administrar os recursos e ações dos cidadãos e da sociedade. Nesse sentido, os eventos sociais, como shows e festas, são exemplos desse tipo de demanda, pois o gerenciamento de espaço, convites, características dos participantes, comissão para vendedores e arrecadação demandam esforços diretamente proporcionais às magnitudes do evento, isto é, o que pode ser facilmente computado e gerenciado para um evento de menor proporção torna-se difícil para eventos de maior magnitude, com maior quantidade de pessoas de perfis diversos. Este relatório aborda a solução do problema tal como seu funcionamento e processamento dos dados pertinentes ao evento.

A solução da problemática centra-se de modo a responder questões essenciais relacionadas ao problema: como solicitar ao usuário as preferências, tal como número de ingressos e preços? Como manter a repetição até o fim das vendas ou esgotamento dos ingressos? Como armazenar esses dados para computá-los? Como processar a arrecadação, a quantidade de vendas totais e por tipo de ingresso, a comissão dos vendedores e quantidade de suas vendas e a média das idades? Como determinar qual tipo de ingresso foi o

mais vendido? Como imprimir os dados tratados e computados para o usuário? A solução encontrada é estruturada da seguinte maneira:

1. Variável Booleana é inicializada para ser falseada em caso de negação do período de vendas ativo;
2. Variáveis gerais de contabilização são declaradas e inicializadas;
3. Solicitação para o usuário de preço por tipo de ingresso, quantidade a ser vendida e cortesias;
4. Um loop principal é criado para ficar ativo até fechamento das vendas, que é determinado pelo usuário, ou esgotamento de ingressos, que são atualizados;
5. No loop principal pergunta-se se a venda foi comissionada ou não:
 - (a) Em caso afirmativo, pergunta-se qual curso do vendedor comissionado;
 - (b) Do contrário, a venda é direta, não comissionada.
6. Em ambas as modalidades de venda, pergunta-se a quantidade vendida;
7. Inicia-se um loop que contém perguntas sobre o tipo de ingresso vendido, que se encerra quando corresponde à quantidade vendida;
8. Após o loop, são computados e processados os dados obtidos e armazenados nas variáveis gerais;
9. Encerrado o loop principal, o software computa o tipo de ingresso mais vendido através de condicionais, calcula a média das idades dos compradores e soma-se às variáveis gerais de contabilização;
10. O programa imprime os dados pertinentes ao evento.

2. Desenvolvimento

Esta seção se dedica a explicar as questões abordadas na introdução e relacionar com a solução do problema.

Como solicitar ao usuário as informações pertinentes ao evento e armazená-las? A organização do evento, ou usuário, precisa inserir as informações que serão processadas e computadas. Em caso de shows e eventos, geralmente se trata de informações relacionadas aos preços e quantidade de ingressos a serem vendidos. Nesse sentido, perguntas devem ser exibidas na tela, e as respostas devem ser entradas e armazenadas.

Na linguagem interpretativa Python, o comando de entrada que permite exibir uma mensagem na tela e solicitar uma entrada é *input*. A entrada, entretanto, inicialmente é uma *string*, que é transformada em número inteiro ou decimal através de comandos de conversão, como "*int(input("Digite um valor x"))*" e "*float(input("Digite um valor y"))*", respectivamente. As entradas são armazenadas em variáveis, que são valores manipulados pelo software conforme sua execução [Menezes 2019].

No programa em questão, tais comandos são utilizados para realizar perguntas referentes aos valores numéricos, como total de ingressos a serem vendidos, preços por tipo de ingresso, as cortesias para membros do Diretório Acadêmico e convidados, quantidade de ingressos vendidos, qual tipo de ingresso foi vendido. Além disso, é utilizado para se referenciar a códigos nominais, como 1 e 2 para os cursos de Biologia e Enfermagem, respectivamente, e afirmação e negação do período de vendas.

Como manter a repetição até o fim das vendas ou esgotamento dos ingressos? O software precisa computar os dados e imprimir as saídas ao fim das vendas ou esgotamento dos ingressos. Por isso, faz-se necessário uma estrutura de repetição que encerre o processamento dos dados ao fim do período de vendas ou esgotamento dos ingressos.

Em Python, existem duas estruturas básicas de repetição: o comando *for*, que geralmente é empregado quando se conhece o número de vezes que um processo se repetirá; e o *while*, quando não se conhece quantas repetições serão feitas [Ramalho 2015].

No programa tratado, não se conhece o número de repetições necessárias, pois não se sabe quantas vendas serão feitas, tampouco se todos os ingressos serão vendidos dentro do período de vendas. Por isso, a estrutura empregada foi *while*, que é estruturado em um *loop* principal que executa o processamento de dados somente se houver disponibilidade de ingressos e se ainda está no período de vendas, que verifica a validade, através de operação Booleana, e encerra o *loop*.

Como armazenar esses dados para computá-los? Como supracitado, os dados são inseridos por meio de comandos de entrada e são armazenados em variáveis, que são modificados conforme a execução do programa. Em caso de eventos, computar arrecadação é de suma importância, por exemplo; e o desafio é acumular os dados em variáveis a cada venda feita.

Para solucionar tal desafio, foram empregados quatro grupos de variáveis: as variáveis gerais de acréscimo e decréscimo, as variáveis de alteração por venda, as variáveis fixas e as variáveis auxiliares. As variáveis gerais são responsáveis por guardar os dados de todas as vendas feitas enquanto o *loop* principal estava em execução; as variáveis de alteração por venda, por outro lado, são responsáveis por armazenar os dados de cada venda e depois adicionar os novos dados às variáveis gerais, modificando-se a cada venda feita. As variáveis fixas armazenam valores pertinentes às características inalteráveis do evento, como preço dos ingressos, e são usadas para computar o valor de arrecadação total e por tipo de ingresso para cada venda. As variáveis auxiliares são recursos para computar especificidades dos dados, como comissão e cortesia para fins de cálculo estatístico.

No programa, essas variáveis são numéricas e relacionam-se por meio de operações aritméticas. No caso, as variáveis de alteração por venda são responsáveis por se somar às variáveis gerais após computadas, o comando que as relaciona é "+=", em caso de acréscimo de valores, e "-=", no caso de decréscimo. As variáveis fixas são relacionadas com a quantidade de ingressos vendidos por tipo a cada venda, com comando de multiplicação "*", posto entre as variáveis [Menezes 2019]. As variáveis auxiliares são usadas para computar especificidades como comissão para vendedores comissionados e cortesias, a fim de evitar computação incorreta dos dados, como excedente de ingressos de comissão e cortesias. As variáveis de acréscimo são referentes à soma das idades, ingressos emitidos, arrecadação, quantidade de venda comissionada, quantidade de ingressos vendidos por tipo; a de decréscimo, refere-se ao total de ingressos disponível para venda, que é decrescida a cada *loop*.

Como processar a arrecadação, quantidade de vendas totais e por tipo de ingresso, a comissão dos vendedores e quantidade de suas vendas e a média das idades? As características dos eventos é determinada por seus organizadores. No software em questão, o usuário pode determinar a quantidade de ingressos disponíveis tal como o valor dos ingressos por tipo.

No programa, após as entradas de ingressos a vender e seus valores, há uma estrutura de laço, que é responsável pela comunicação de processos [Ramalho 2015]. No *loop*

principal, há duas estruturas principais: a responsável pela venda direta, não comissionada, e a responsável pela venda comissionada. A estrutura de venda direta se organiza baseada em perguntas e respostas, como a quantidade de ingressos vendidos no total e por tipo, e são processadas. A venda não comissionada diferencia-se por dois pontos principais: qual é o curso do vendedor e o cálculo da comissão. São oferecidas duas opções de curso: Biologia e Enfermagem, os quais têm suas variáveis gerais e de arrecadação de venda que são computadas conforme a opção digitada pelo usuário. O processamento das vendas para ambos assemelha-se com o da venda direta, tal como seu processamento; a diferença, entretanto, é o cálculo da comissão dos vendedores. A computação da comissão verifica, através de operações aritméticas e variáveis auxiliares, se há quantidade suficiente de ingressos para comissionar o vendedor. Em caso afirmativo, há subtração do valor total de ingressos disponíveis; em caso negativo, o vendedor é comissionado com o maior valor possível.

Como determinar qual tipo de ingresso foi o mais vendido? A determinação de qual tipo de ingresso foi o mais vendido demanda de estruturas de comparação dentro de uma série de valores computados. Para isso, foi essencial o uso de operadores de comparação e lógicos. Em Python, os operadores de comparação, tal como maior, menor, igual, maior ou igual, menor ou igual assemelham-se aos utilizadas na matemática [Menezes 2019].

1. >: maior
2. >=: maior ou igual
3. <: menor
4. <=: menor ou igual
5. ==: igual

Os operadores lógicos condicionais, em Python, *if*, *elif*, *else* [Ramalho 2015] são responsáveis por fazer verificações conforme as comparações feitas entre valores ou variáveis, executando caso valores booleanos sejam verdadeiros, sendo essenciais para tomadas de decisão.

No programa, os operadores lógicos e de comparação relacionam-se com as variáveis gerais responsáveis por contabilizar a quantidade de ingressos por tipo, sendo comparadas entre si para verificar qual foi a mais vendida ou se pelo menos duas foram igualmente vendidas.

Como imprimir os dados tratados e computados para o usuário? O programa precisa apresentar as informações após o processamento. Para isso, as variáveis responsáveis pela contabilização dos dados devem ser impressas na tela. Em Python, o comando *print* é responsável por fazer tal impressão, imprimindo o valor da variável. Por exemplo, *print(variavel)*. No programa, uma série de comandos *print* foi adicionada para imprimir as variáveis gerais, que são os dados processados e armazenados. [Menezes 2019]

2.1. Manual de instrução para o uso do programa

1. Digite a quantidade de ingressos a ser vendida;
2. Digite o preço de cada tipo de ingresso;
3. Digite a cortesia do D.A. e dos convidados;

4. Digite se está ou não no período de vendas conforme o código numérico informado solicitação de entrada;
5. Digite se a venda é comissionada ou não conforme o código numérico informado na solicitação de entrada
6. Execute a venda com entrada de dados referente à quantidade de ingressos vendidos e a soma das idades. O código permanecerá em *loop* em caso de erro de entrada;
7. Os processos anteriores serão executados repetidamente até o usuário encerrar o período de vendas ou os ingressos esgotarem.

3. Conclusão

Este relatório apresentou a solução para computação de um sistema de vendas de ingressos em linguagem interpretativa Python, sendo dados solicitados ao usuário, computados e impressos em interface de linhas de comando. A solução resolve o problema proposto através de várias estruturas pertinentes ao Python e às linguagens de programação, como comandos de entrada e saída de dados, estruturas de repetição, variáveis gerais, que incrementam e decrementam; variáveis de alteração por venda, que armazenam os dados a cada venda para incrementar ou decrementar; variáveis fixas, para armazenar dados inalteráveis; e variáveis auxiliares para auxílio de cálculos matemáticos; além de operadores lógicos e de comparação.

Várias questões foram abordadas e estruturadas de modo a explicar a constituição básica do programa, tendo suas etapas solucionadas através do uso de variáveis de inicialização, estruturas de repetição, operadores lógicos, de comparação e Booleanos.

O programa resolve o problema proposto. Entretanto, possui a limitação de armazenar dados na memória, tendo que executar todo o processo antes de encerrar o computador. Além disso, apenas as configurações dadas pelo usuário são a quantidade de ingressos a vender, os preços por tipo de ingresso e as cortesias do Diretório Acadêmico e seus convidados; portanto não se pode configurar o curso dos vendedores comissionados, nem alterar as regras de comissão, ou especificar quais cursos têm desconto.

Este relatório objetivou explicar os métodos utilizados para solução do problema de modo estruturado, conciso e direto.

Referências

Menezes, N. N. C. (2019). *Introdução à programação com Python*. Novatec, 3ª edition.
Ramalho, L. (2015). *Python Fluente*. Novatec, 1ª edition.

[Menezes 2019] [Ramalho 2015]