

Sistema de automatização do Dispensário Santana

Ricardo C. Oliveira

Departamento de Ciências Exatas Curso de Engenharia de Computação - Universidade
Estadual de Feira de Santana (UEFS)
Avenida Transnordestina, s/n - Novo Horizonte - CEP 44036-900 - Feira de Santana - Bahia,
Brasil

rickoliver001@hotmail.com

Abstract. *This report aims to describe the construction and development of a donation system designed to automate the process of collecting donations from the Santana Dispensary. In addition, please inform what the requirements of the project were and how all the difficulties encountered in solving the problem were addressed and overcome. Like, describe how the system works properly and without obstacles.*

Resumo. *O presente relatório tem como objetivo descrever a construção e desenvolvimento de um sistema de doações feito para automatizar o processo de arrecadação de donativos do Dispensário Santana. Além disso, relatar quais foram os requisitos do projeto e a forma de como foram sanadas e superadas todas as dificuldades encontradas na resolução da problemática. Assim como, descrever como o sistema funciona de forma correta e sem empecilhos.*

1. Introdução

O Dispensário Santana é uma instituição fundada em 1946, dirigida pelas religiosas da Congregação do Santíssimo Sacramento, que tinha como objetivo ajudar pessoas em condição de pobreza, principalmente idosas, oferecendo-as donativos. Hoje, mais modernizada, a instituição recebe doações de voluntários a causa. Porém, essa arrecadação ocorre de uma forma totalmente manual, o que gera uma dificuldade para os funcionários administrarem todo o material coletado.

No ano de 2019, com o advento da pandemia de Covid 19, o dispensário teve seu contingente de doadores drasticamente reduzido, assim como, em contrapartida, inferiu-se que o número de famílias necessitadas aumentou nesse período devido a grande demanda de donativos. Diante de tal cenário atípico, os donativos ingressaram em um fluxo irregular de entrada no Dispensário, mudando totalmente o método manual de arrecadação tradicionalmente utilizado na instituição, visando uma melhor otimização de todo o processo.

Assim, foi-se criado um programa com o intuito de ajudar os funcionários a padronizar a arrecadação de doações de uma forma tecnológica e intuitiva, melhorando, assim, a forma de recebimento dos donativos e automatizando o processo. Diante disso, o Dispensário Santana tem espaço e competência para sanar as adversidades e contratemplos, referentes às doações, que a pandemia impulsionou até o momento, como a chegada irregular de donativos na instituição.

2. Metodologia

Antes de iniciar as discussões sobre o que o sistema deveria apresentar, foi-se estudado o problema em si. Portanto, na fase inicial, com a leitura da problemática em questão, descobriu-se que o sistema de arrecadação deveria receber entradas de donativos para que fosse possível a contabilização e formação de cestas básicas. No presente caso, essas cestas deveriam conter: 1 kg de açúcar; 4 kg de arroz; 2 kg de café; 2 un de extrato de tomate; 3 un de macarrão; 1 pct de bolacha; 1 L de óleo; 1 kg de farinha de trigo; 4 kg de feijão; 1 kg de sal. Além disso, as cestas poderiam ser acrescidas de um item extra, como donativos que não compõem a cesta básica.

Após o conhecimento dessas questões, foi-se detalhado os requisitos que o sistema deveria conter para adequar-se ao problema. Nesse sentido, os requisitos do programa incluíam contabilizar os donativos e imprimir um relatório contendo: o total de itens recebidos; total de itens (independente do tipo) doados por pessoas físicas e por pessoas jurídicas; quantas cestas básicas poderão ser formadas; quantas cestas básicas receberão um item extra (outros); quantas cestas básicas não receberão um item extra (outros); após a montagem das cestas, quais foram os itens que sobraram.

Para isso, foi preciso definir as entradas que o programa utilizaria para os referidos processos. Diante disso, para otimizar a construção do programa, o problema foi segregado em duas partes: dados do doador e dados das doações.

Utilizando esses dados, o programa deveria registrar as informações do doador, e realizar as contabilizações das doações. Para a área do doador, o sistema receberia como entrada somente nome e tipo de pessoa: física ou jurídica. Nesse contexto, seria possível mensurar o grau de desenvolvimento na construção do código.

Logo depois, para as escolhas do usuário sobre como seria a ordem de execução do código, foi determinado que seria mais viável registrar as informações do doador e na sequência os donativos, para não haver doações vazias (sem um doador). Para isso, o programa contém o “menu de escolhas”, para uma maior otimização e usabilidade. Nele, o usuário escolhe a opção que deseja realizar, como fazer uma doação ou visualizar o relatório.

Na sequência, o código segue caminho para a área de doações, onde as mesmas são contabilizadas e estruturadas em cestas e nas demais categorias citadas que o problema requisita.

Antes de apresentar o detalhamento do código, algumas questões importantes valem a pena serem ressaltadas. Os itens extras, por exemplo, podem ser quaisquer objetos que queiram ser doados, independente dele ser alimento ou não. No desenvolvimento das questões, não foi admitido que seria doado dinheiro por esse sistema, assim, o programa opera somente com unidades do donativo, como: 1 unidade de item extra.

Diante do exposto, a ordem de codificação do programa seguiu nessa ordem, onde utilizou-se laços de repetição e estruturas condicionais em sua maior parte.

Assim, seguindo a ordem de desenvolvimento do código, é importante pontuar como e quais ferramentas foram utilizadas na construção do sistema. Inicialmente,

houve a declaração de variáveis que seriam utilizadas no dado código. Em sua maior parte, são variáveis de contagem que servem para o armazenamento da quantidade de donativos que estão alimentando o sistema, elas serão responsáveis pela formação do relatório parcial e final do programa.

Logo após, o programa adentra em um laço de repetição *while* que é responsável por fazer a manutenção do funcionamento do sistema, que só ocorre por ter sido declarada uma variável de nome “sistema_ON”, assim, o código vai manter-se em atividade até a mudança do valor desta variável. O programa finalmente é iniciado para o usuário quando lhe é apresentado o “menu de escolhas”, para que possa navegar no sistema. Nessas escolhas, utilizou-se variáveis sem tipagem numérica (como int e float) para facilitar a validação de decisões do sistema, já que as mesmas são realizadas através de números atribuídos a cada uma delas.

Após ser apresentado ao usuário o “menu de escolhas”, a construção do código seguiu-se com o registro do doador (caso ele tenha escolhido realizar uma doação). Nessa parte, ocorre a entrada do nome do doador para mais adiante ser usada em uma mensagem de confirmação de doação. Logo após, a entrada do tipo de pessoa (física ou jurídica) deve ser lido. Nesse ponto do código, ele é dividido em dois blocos de comandos, um para cada tipo de pessoa (Figura 1).

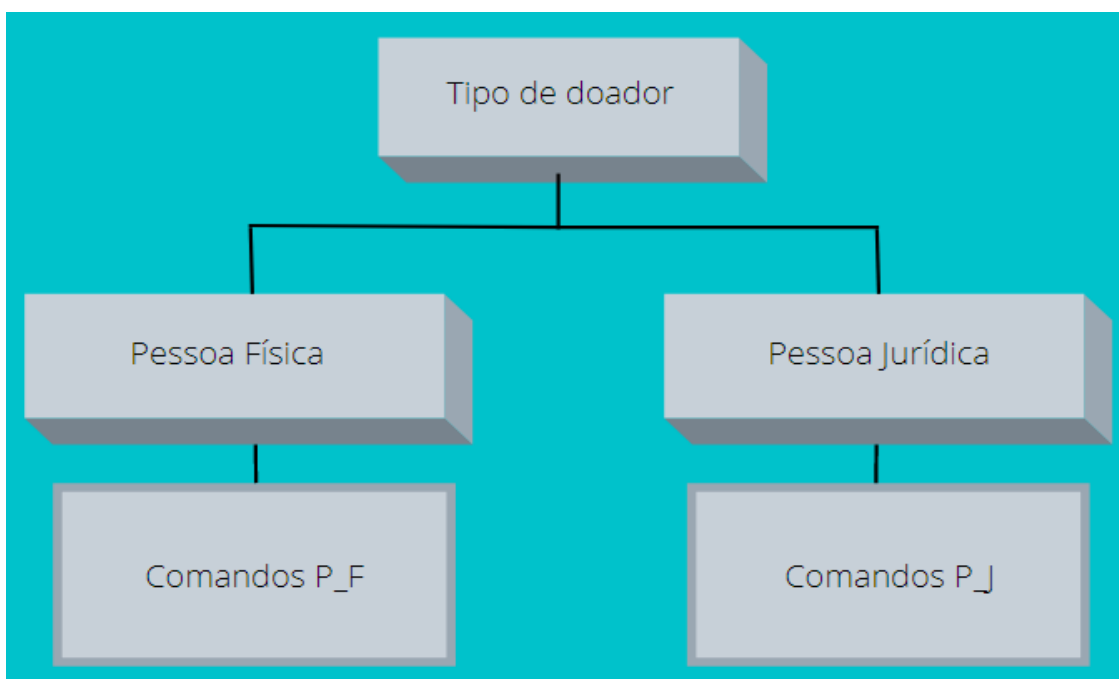


Figura 1 - Estrutura condicional para tipo de doador

Seguindo adiante, o programa apresenta o “menu de escolha das doações”, onde são feitas as escolhas do tipo de alimento e a quantidade que irá ser doada. Para cada tipo de alimento é atribuído um número para a escolha, por exemplo: 0 é a escolha para doar açúcar; 1 para café; etc. Assim, as estruturas condicionais vão incrementando as

quantidades que foram digitadas pelo usuário às variáveis de contagem declaradas no início do código. Dessa forma, conforme o sistema é alimentado com os dados dessas quantidades doadas, as variáveis vão somando o total de cada item recebido. Esse bloco de comandos se repete tanto para pessoas físicas quanto para jurídicas, onde, para cada um desses, há variáveis para a contagem separada, para os dois tipos.

É importante ressaltar que para cada escolha feita pelo usuário, há uma validação para tal decisão. Essas validações servem para a manutenção do funcionamento do sistema, assim como, para inviabilizar a perda de informações no decorrer de seu uso, como o acontecimento de um erro, o qual faz o sistema parar completamente. Nesse sentido, as validações foram feitas com laços de repetição *while*, sendo utilizadas escolhas diversas, como condições, para a prevenção de erros no programa.

No decorrer das doações, sempre é apresentado ao usuário se o mesmo deseja realizar uma nova doação, isso somente é possível por influência de um laço *while* que faz essa manutenção de escolha para a volta à doação.

Além do bloco de comandos das doações, há também o bloco de comandos para a visualização do relatório, sendo ele parcial ou final, dependendo de quando o sistema for encerrado. Assim como o laço *while* repete a pergunta das doações, ele também mantém a visualização do relatório no mesmo processo.

Desse modo, o relatório das doações é feito seguindo os requisitos do problema. Para a soma total de todos os itens, o sistema faz uma soma de todas as variáveis acumuladoras, assim como, no segundo requisito (total de itens, independente do tipo, doados por pessoas físicas e por pessoas jurídicas), ele soma as variáveis separadamente, de acordo com o tipo da pessoa. Para a formação das cestas, utilizou-se um laço *while* com as condições de formação de cesta (como 4 kg de feijão, 4 kg de arroz) usando o operador lógico *and* para associar todos esses elementos em uma só condição. Desse modo, se a condição for aceita, a cesta básica será formada, ao passo que, as variáveis contadoras vão sendo decrementadas ao longo desse processo de acordo com a quantidade necessária na cesta, pois elas agrupam um quantitativo de alimentos, como um conjunto.

Para a formação de cestas básicas com ou sem itens, foi feita uma lógica para que não fossem feitas operações com números negativos. Assim, as cestas com ou sem itens extras foram contadas com a quantidade desses itens como parâmetro. Se a quantidade de extras fosse maior que o número de cestas, todas as cestas teriam um item extra, e assim por diante. Se a quantidade de itens extras for menor, as cestas com item extra serão o número de itens extras. E no último caso, se as quantidades forem iguais, as cestas com item extra serão iguais ao número de itens extras. Consequentemente, as cestas sem extra seriam o resto de todas as operações citadas anteriormente.

Já na parte final do código, resta o último requisito do problema, que se refere aos itens que sobraram após a montagem das cestas. Para isso, foi necessário somente imprimir na tela por meio de *print's* as variáveis utilizadas para a soma das quantidades, que a essa altura, já estão com os restos das doações que não adentraram na formação das cestas, ou seja, a parte que não foi decrementada na estrutura condicional de formação das cestas.

Enfim, o laço de repetição *while* do “menu de escolhas” se conclui com a escolha “SAIR” para encerrar o programa. Para isso, o valor da variável “sistema_ON” citada no início, é modificada para “sistema encerrado”, encerrando o programa se essa for a escolha do usuário (Figura 2).

```
if escolha == '2':  
    print('\nSistema de doações encerrado. Volte sempre!')  
    sistema_ON == 'Sistema Encerrado' #Nessa ocasião, é atribuído à variável um valor diferente da inicialização, para encerrá-lo.  
    escolha = 'Sistema Encerrado' #A variável escolha é mudada também para não se iniciar um loop infinito de print's.
```

Figura 2 - Encerramento do programa

O programa foi desenvolvido em linguagem Python, e em sistema operacional Windows, o que tornou a experiência mais prazerosa e menos complicada para o seu desenvolvimento, visto que outras linguagens de certo modo são mais “complicadas” aos olhos de muitos, e além disso, o Windows é mais visual que o sistema operacional Linux por exemplo. No python, foi utilizado basicamente laços de repetição e estruturas condicionais compostas, como o *if-elif*, para a resolução da maioria dos requisitos do sistema.

3. Resultados e discussões

Para uma melhor utilização do sistema, algumas ações são apropriadas para o pleno funcionamento do mesmo . Ao iniciar o programa para o usuário, o sistema apresenta um “menu de escolhas”, para o bom funcionamento deve-se responder somente as escolhas apresentadas, caso contrário, o sistema irá voltar várias vezes ao menu para uma escolha apropriada. Deve-se também, adicionar o nome do doador para a continuação da doação, caso não haja nome do doador, o programa imprime na tela uma mensagem de erro e o campo para preenchimento novamente.

O mesmo vale para as outras escolhas apresentadas na tela, como o tipo de alimento que irá ser doado. Embora nessa escolha em específico não haja uma validação para a resposta do usuário, ela não influenciará no funcionamento do programa, pois a escolha adversa que foi feita não tem final, consequentemente, a doação não será feita e será perguntado novamente se o usuário deseja realizar uma nova doação.

Logo após a escolha do tipo de alimento a ser doado, o sistema solicita a quantidade que será doada. É importante informar que, o programa possui pleno funcionamento somente com números inteiros, a não ser o campo de nome do doador, então, consequentemente, o sistema apresentará um erro (*ValueError*) na tela do usuário (Figura 3) caso neste ponto em específico do código ele digite um valor além de um número inteiro

```
Traceback (most recent call last):  
  File "c:\Users\Wanas\Documents\Computação\Programação\1º PBL\PBL_DispensarioSantana.py", line 98, in <module>  
    doacao_acucar = int(input('\nQuantos quilos(kg) de açúcar vão ser doados?\nQuantidade: '))  
ValueError: invalid literal for int() with base 10: 'aleatório'
```

Figura 3 - Erro de valor (Value Error)

Esse erro se refere ao valor atribuído na entrada, como a variável que contém essa entrada está tipada em *int*, qualquer valor diferente de um número inteiro apresenta o referido erro na tela.

O usuário poderá realizar quantas doações forem necessárias, após cada doação, é possível visualizar o relatório parcial dos donativos presentes em “estoque”. Assim como, após as dadas doações é possível sair do sistema escolhendo essa opção no menu.

Como saída de dados, o programa apresenta o relatório seguindo a ordem dos requisitos do problema. Assim, é possível visualizar tudo o que já foi realizado até o momento. A diferença entre o relatório parcial e o final está justamente na próxima ação do usuário no sistema, se ele optar por realizar alguma nova doação, consequentemente, o relatório será parcial, já que ele ainda está sendo alimentado. Deste modo, caso o usuário opte por finalizar o programa, aquele último relatório solicitado será o final.

Para tal considerações apresentadas, o programa foi testado de variadas formas. Inicialmente, o primeiro teste realizado foi de validação de escolhas, sendo assim, cada escolha feita pelo usuário foi testada e revisada. Logo após, com a finalização do desenvolvimento da área de doações, foram feitas as testagens para as entradas, assim teve-se conhecimento dos erros de *ValueError* citados acima, onde foram feitos todos os reajustes necessários para a prevenção destes casos. Seguindo com os testes, foram validadas as formações de cestas básicas, onde foram testados variados casos de doações, como: a formação de somente uma única cesta, formação de uma cesta com item extra e outra sem item extra, doar uma grande quantidade de donativos para casos extremos, etc. Em todo teste de formação de cestas realizado, nos primeiros resultados, o valor achado não condizia com o esperado, isso porque, inicialmente, as estruturas condicionais não estavam perfeitamente projetadas e desenvolvidas. Porém, com o avanço do tempo e da construção do código, todas as adversidades foram encontradas e reorganizadas para o pleno funcionamento do sistema.

Assim, o código foi testado e estudado para ter o conhecimento de possíveis erros de sistema que poderiam ocorrer.

4. Conclusão

Dado o presente problema, foi possível desenvolver e aprender variadas ferramentas e funcionalidades da linguagem Python. Todos os requisitos foram cumpridos adequadamente ao decorrer do desenvolvimento do projeto.

As secções que podem ser melhoradas no programa são a parte visual do código, a validação tanto da quantidade de itens a serem doados quanto do nome do doador, já que podem ser digitados numerais no campo onde seria ideal pôr o nome da pessoa.

Porém, algumas funcionalidades a mais que não foram requisitadas no problema foram implementados, como a validação das escolhas do usuário nos menus que se repetem no sistema, tornando assim o código mais fortificado e bem preparado para as variadas entradas que podem ocorrer no sistema.

4. Referências

Dispensário Santana (2017) “O Dispensário”,
[Dispensário Santana \(dispensariosantana.com.br\)](http://dispensariosantana.com.br), Fevereiro.

Python #05 - Condicionais, operadores relacionais e lógicos, João Paulo Just Peixoto, Professor Just, publicado em 04 de maio de 2020 na plataforma Youtube.

Disponível em: [\(4227\) Python #05 - Condicionais, operadores relacionais e lógicos - YouTube](#)

Python #07 - Laços de repetição WHILE, João Paulo Just Peixoto, Professor Just, publicado em 08 de maio de 2020 na plataforma Youtube. Disponível em: [\(4227\)](#)

[Python #07 - Laços de repetição WHILE - YouTube](#)