

Sistema de Gerenciamento de Doações e Montagem de Cestas Básicas

Sival Leão de Jesus

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

Abstract. *This report will present the procedures made for the creation of the software responsible for the "System for the Management of Donations and Assembly of Basic Baskets" at the request of the Santana Dispensary, where the Python programming language was used for the realization of the software. The software developed has as main function the storage of products received through donations and assembly of basic food baskets.*

Resumo. *Neste relatório será apresentado os procedimentos feitos para a criação do software responsável pelo "Sistema de Gerenciamento de Doações e Montagem de Cestas Básicas" a pedido do Dispensário Santana, onde foi utilizado a linguagem de programação Python para a realização do software. O software desenvolvido tem como principal função o armazenamento de produtos recebidos por meios de doações e montagem de cestas básicas.*

1. Introdução

A pandemia causada pelo coronavírus gerou um cenário de crise econômica no Brasil. Nessa perspectiva, com o isolamento social ficou muito mais difícil a busca por um emprego para se sustentar, como comprovado pelo estudo realizado pelo IBGE, o número de desempregados ultrapassou 15,2 milhões no primeiro trimestre de 2021 [Meneghetti, Luana 2021] .

Em virtude disso, muitos indivíduos ficaram com a sua renda comprometida. Nesse sentido, segundo o Datafolha 46% da população do Brasil teve sua renda familiar reduzida por conta da pandemia [Carrança, Thais 2020].

Com o objetivo de promover a segurança alimentar e nutricional, o Dispensário Santana, uma instituição feirense, tem como objetivo recolher donativos para distribuir a pessoas com baixa renda e idosos. No entanto, como a demanda por doações e distribuição de cestas básicas aumentou, o Dispensário Santana solicitou aos alunos de engenharia da computação da UEFS que desenvolvessem um software em linguagem python para ajudar nesse serviço.

O Dispensário Santana recebe doações de alimentos de pessoas físicas e jurídicas. Porém, devido a irregularidades de arrecadação e a alta procura de cestas por famílias necessitadas, o Dispensário encontrou várias dificuldades para organizar os itens recebidos e montar as cestas, comprometendo assim a eficiência desse serviço.

O programa "Sistema de Gerenciamento de Doações e Montagem de Cestas Básicas" desenvolvido tem como função armazenar os itens recebidos, para que nada seja perdido ou esquecido. Além disso, o próprio programa monta as cestas básicas, a fim de

que todos tenham as mesmas quantidades de alimentos. Dessa maneira, um número maior de pessoas são beneficiadas, oferecendo alimentação básica, além disso, com intuito de não haver desperdício, o sistema guarda os itens que restaram das montagens da cesta até que seja possível uma nova formação seja feita.

Diante disto, esse relatório tem como objetivo esclarecer e explicar as funcionalidades presentes no programa tais como são armazenadas e montadas as cestas básicas e o processo de desenvolvimento para a construção do software.

2. Metodologia

Durante as sessões tutoriais foi visto que as variáveis seria de extrema importância no código, como a maioria dos participantes havia participado da Semana de Integração de Engenharia de Computação (SIECOMP) ofertado pela UEFS nos dias de 21 a 25 de Fevereiro, já tínhamos ideias de como começar o nosso código e como fazer as funcionalidades básicas como entrada de dados e armazenamento, diante disto foi visto que o programa funcionaria a base de variáveis para armazenar os itens recebidos já que nos possibilita a manipulação de dados.

Ao iniciar o processo de desenvolvimento do código, de princípio foram adicionadas as variáveis necessárias para chegar no objetivo, como armazenar os itens recebidos, onde essa variável tinha uma função de acréscimo e de decremento para as montagens da cesta básica. Também foram atribuídas mais 3 variáveis totalizando 4 para cada item que compõem a cesta básica, além das variáveis responsáveis por mostrar o total ao fim do programa ou quando solicitado, como visto na figura 1. Porém viu-se a necessidade de adicionar outras para que o código funcione em um loop onde foi discutido nas sessões tutoriais o uso de menu e submenu utilizado o comando “while” da linguagem de programação Python que significa “enquanto”, cada while recebia uma variável própria para não interferir nas demais, onde esta variável muda dependendo da situação.

```

19 # 20 # 21 # 22 #
20 acucar = 0 40 macarrao = 0 60 feijao = 0 80 #
21 acucar_j = 0 41 macarrao_j = 0 61 feijao_j = 0 70 total_extra = 0
22 acucar_f = 0 42 macarrao_f = 0 62 feijao_f = 0 71 extra = 0
23 acucar_t = 0 43 macarrao_t = 0 63 feijao_t = 0 72 #
24 # 44 # 64 # 73 cesta_basica = 0
25 arroz = 0 45 pct_bolacha = 0 65 sal = 0 74 cesta_basica_t = 0
26 arroz_j = 0 46 pct_bolacha_j = 0 66 sal_j = 0 75 cesta_basica_ext = 0
27 arroz_f = 0 47 pct_bolacha_f = 0 67 sal_f = 0 76 #
28 arroz_t = 0 48 pct_bolacha_t = 0 68 sal_t = 0 77 p_fisica_kg = 0
29 # 49 # 69 # 78 p_fisica_l = 0
30 cafe = 0 50 oleo = 0 70 # 79 p_fisica_pct = 0
31 cafe_j = 0 51 oleo_j = 0 71 # 80 p_fisica_un = 0
32 cafe_f = 0 52 oleo_f = 0 72 # 81 p_fisica_ext = 0
33 cafe_t = 0 53 oleo_t = 0 73 # 82 #
34 # 54 # 74 # 83 p_juridica_kg = 0
35 extrato_de_tomate = 0 55 farinha_de_trigo = 0 75 # 84 p_juridica_l = 0
36 extrato_de_tomate_j = 0 56 farinha_de_trigo_j = 0 76 # 85 p_juridica_pct = 0
37 extrato_de_tomate_f = 0 57 farinha_de_trigo_f = 0 77 # 86 p_juridica_un = 0
38 extrato_de_tomate_t = 0 58 farinha_de_trigo_t = 0 78 # 87 p_juridica_ext = 0

```

Figura 1. Variáveis responsáveis pelo funcionamento do código

Nas montagens das cestas básicas em uma primeira tentativa foi se utilizando o comando "if" que significa "se" ou seja se determinada condição for verdadeira executa tal comando, porém esse comando só executava uma única vez e o objetivo não era esse e sim repetir em um loop até que a condição seja falsa, para solucionar esse problema foi utilizado novamente o comando while junto com as variáveis de acréscimo e decremento, citados acima, para que assim processe corretamente e não ficassem infinitas às montagens das cestas.

Os submenus foi utilizado para procedimento simples com a separação de pessoas físicas e jurídicas, além de questionar o usuário sobre doação de itens não mencionados, no caso um item extra poderia ser aceito pelo programa, que era um requisito do sistema. Apesar de ser um submenu, o conceito é o mesmo utilizado no principal.

3.Resultados e discussões

Ao decorrer das sessões e à medida que nós elaboramos os códigos, foram surgindo vários questionamentos durante as sessões. Uma das principais dúvidas gerada no grupo era o que condizia ao total de itens doados por pessoas físicas e jurídicas a ser exibido no final do software, já que as unidades de medida eram diferentes como quilograma (Kg), litros, unidades e pacotes, desse modo não poderia ser somado esses valores, sendo assim, foi decidido por mostra o total separado por unidade de medida dos itens, para que respeite as unidades de medida.

Outra questão é como deveria ser tratado os itens extras em relação às unidades de medida, já que o software não é capaz de distinguir por conta própria a unidade de medida do produto em se, ele apenas recebe e processa, então para que não ficasse errados as medidas foram consideradas que para produtos com unidades de medida de quilograma e litros as unidades seriam as representantes. Como por exemplo, se 1 produto de quilograma ou litro fosse doado, o software imediatamente considera como 1 quilograma ou um litro. E vale ressaltar que só seria aceito em valores inteiros para que assim ficasse condizente se por acaso receber uma unidade ou pacote.

Um ponto bem importante é como seria aceito os valores de doação dos alimentos inteiros ou reais ou de acordo com o que fornecido no mercado, mas discutidos entre os membros e com orientação do tutor foi visto que produtos que são medidos por quilogramas e litros deveria ser aceito em números do conjunto reais, ou seja valores que possuem casa decimais, contudo os itens que tinha por unidade de medida pacotes e unidades deveria ser em valores inteiros, vale ressaltar que o código é desenvolvido na linguagem Python, linguagem essa que tem como fundação a língua inglesa, ou seja, as operações matemáticas e entrada de dados vai respeitar as normas da língua inglesas onde os números decimais utilizam um ponto ".", e não uma vírgula ",", para separar a parte inteira da parte decimal.

Devido às exigências fornecidas em discussão viu-se a necessidade de validar as entradas do usuário tanto para que o processamento ocorra perfeitamente quanto para que o programa não feche caso insira um valor não permitido.

4. Validações

Com o intuito de fazer o programa funcionar em um loop ou até o usuário decidir encerrar foi necessário fazer validações nas entradas do usuário para que só aceite o que foi solicitado, impedindo assim o fechamento do programa e a perda de dados cadastrados anteriormente garantido assim também o funcionamento corretamente.

Foram feitas validações em várias partes do código como menus, cadastro de pessoas, entrada de itens e entrada de item extra, onde se inserido um valor não desejado

o programa retornaria a pergunta. Ao longo da produção do código algumas validações precisaram ser mudadas.

4.1. Menu

Nas entradas de menu o sistema foi feito no comando while onde depende de uma condição para funcionar foi se utilizado juntamente o comando if pois se o usuário digitar um valor diferente o while repete o menu como mostra a figura 2. Esse mesmo menu é responsável por cadastrar mais de uma pessoa e garantir que o software não pare a não ser que seja desejado por um usuário. As entradas corretas para esta seção são, a tecla “1” onde o usuário será encaminhado a parte de cadastro e doações, a tecla “2” onde poderá ver o relatório parcial e a tecla “3” que fechará o software exibindo o relatório final.

```
119 on = 0
120 #loop para o código sempre roda
121 #menu
122 while on == 0:
123     print("*100")
124     print()
125     print("-"*5,"MENU","-"*5 )
126     menu = input("\nDigite 1 para doações \nDigite 2 para ver o relatório \nDigite 3 para encerrar \n>>> ")
127     #verificacao se a entrada inserida é correta, caso contrario repete a pergunta
128     if menu != "1" and menu != "2" and menu != "3":
129         on = 0
```

Figura 2. Comando responsável por fazer o loop

A variável “on” mostrada na figura acima na linha 119 é responsável pelo funcionamento do while e enquanto ela for 0 (zero) o comando vai repetir. Na linha 128 o if garante que se for inserido um valor diferente do desejado o comando vai repetir mantendo o software ativo.

4.2. Nome do doador

Para evitar que o usuário inserisse por acidente valores numérico no cadastro de nome foi se utilizando a função “isalpha” onde esse comando só aceita caracteres alfabéticos porém ele não aceita o caractere de espaço, onde se é utilizado para separar o sobrenome, necessitando de outro comando para operar juntamente com ele, onde foi utilizado a função “replace”, que significa substituir, onde o caractere de espaço foi substituído por nada como se fosse apenas um único nome. Vale ressaltar que esse comando foi utilizado no while apenas para validar, só será executado se não possuir apenas caracteres alfabéticos, como qualquer valor numérico, como visto na figura 3, se o usuário digitar corretamente fica salvo como foi digitado, ocorrendo a separação de nome e sobrenome normalmente.

```
134 nome = input("\nNome do doador: ").strip()
135 while not nome.replace(" ", "").isalpha():
136     nome = input("Nome do doador (A nome deve conter apenas letras): ")
```

Figura 3. Comando responsável pela validação de nome

4.3. Tipo de pessoa

Para separar pessoa física e jurídica inicialmente foi feito por meio de uma condição dentro do comando while, onde se o usuário não digitar “F” ou “J”, letras definidas para pessoa física e jurídica respectivamente, o programa repetia a pergunta como mostra a figura 4.

```
105
106     #tipo de pessoa
107     tipo_pessoa = input("\nDigite [ F ] para pessoa Física \nDigite [ J ] para pessoa Jurídica \n>>> ").strip().lower()
108     while tipo_pessoa not in "f" and tipo_pessoa not in "j":
109         tipo_pessoa = input("Digite apenas [ F ] ou [ J ]: ").strip().lower()
```

Figura 4. Primeira tentativa de validar tipo pessoa

Porém, foi encontrado uma brecha neste trecho do código onde se o botão “Enter” fosse apertado sem nenhuma entrada do usuário o software continuava normalmente sem fazer o cadastro do tipo de pessoa e como isso não podia acontecer essa parte do código foi alterada. Como tinha o mesmo princípio de menu foi adaptado o código da figura 2 para que assim não aceitasse simplesmente o botão de confirmar, “Enter”, como visto na figura 5. Para que o software possa aceitar tanto letra maiúscula quando minúscula, para facilitar e agilizar o cadastro foi se utilizado a função “lower” onde transforma a letra inserida pelo usuário em minúscula independente de como ele digitou e foi usado a função “strip” que remove os espaços do início e fim da digitação garantido uma maior agilidade no momento do cadastro. O mesmo código foi usado para perguntar se o usuário gostaria de adicionar um item extra, foi mudado apenas as letras para “S” para sim e “N” para não, como pode se observar na figura 6.

```
140     #tipo de pessoa
141     off = 0
142     while off == 0:
143         tipo_pessoa = input("\nDigite [ F ] para pessoa Física \nDigite [ J ] para pessoa Jurídica: ").strip().lower()
144         if tipo_pessoa != "f" and tipo_pessoa != "j":
145             print("Digite apenas [ F ] ou [ J ]")
146             off = 0
```

Figura 5. Comando responsável pela validação de tipo pessoa

```
333     off2 = 0
334     while off2 == 0:
335         p_extra = input("\nGostaria de doar algo que não foi mencionado acima \nDigite [ S ] para sim e [ N ] para não: ").strip().lower()
336         if p_extra != "s" and p_extra != "n":
337             print("Digite apenas [ S ] ou [ N ]")
338             off2 = 0
339
```

Figura 6. Comando responsável pela validação da pergunta em relação ao item extra

4.4. Valores reais e inteiros

Como as operações ocorridas no código são de extrema importância não podiam haver erros nestas operações e já que as unidades de medidas são diferentes, não podiam ser somadas ou subtraídas, ou seja não pode ser aceito valores reais em unidade por

exemplo, devido este fato, foi feito duas validações para inserir a quantidade de alimento doados onde os produtos que são de quilogramas e litros recebem valores reais e os de pacote e unidade recebem valores inteiros. Apesar de serem duas validações, os códigos são semelhantes mudando apenas a palavra "float" que são os valores reais e a palavra "int" que são os valores inteiros, podemos comparar nas figuras 7 e 8.

```
161 doacao = ""
162 while type(doacao) != int:
163     try:
164         doacao = int(input(f"Quantos {i} você deseja doar: "))
165         while doacao < 0:
166             doacao = int(input(f"Quantos {i} você deseja doar( apenas numero maior que zero): "))
167     except ValueError:
168         print("ERRO!!! Por Favor Digite Apenas Numeros Validos")
169         doacao = ""
```

Figura 7. Comando responsável pela validação de valores inteiros

```
238 doacao = ""
239 while type(doacao) != float:
240     try:
241         doacao = float(input(f"Quantos kg de {i} você deseja doar: "))
242         while doacao < 0:
243             doacao = float(input(f"Quantos kg {i} você deseja doar( apenas numero maior que zero): "))
244     except ValueError:
245         print("ERRO!!! Por Favor Digite Apenas Numeros Validos")
246         doacao = ""
```

Figura 8. Comando responsável pela validação de valores reais

Para essas validações foi usado a instrução "try" e "except" que tem como função tratar erros, ou seja o programa já espera pelo o erro para que assim possa executar outro comando definido pelo programador, com base nisso foi feito um código onde já considerava como uma informação falsa e um while para repetir até que seja verdadeira, explicando mais detalhadamente a variável responsável por receber esse valor foi definida como "string" que reconhece qualquer entrada como texto e não valores numéricos em seguida no while foi definido que se a variável que foi definida como string for diferente de int ou float iria executar o loop garantido que se o usuário erase a pergunta seria retomada, agora entrando na instrução try foi solicitado a pergunta esperando a resposta verdadeira, em outra ocasião se a resposta fosse falsa o software automaticamente fecharia alegando um erro de valor neste caso "ValueError" mas colocando este erro dentro do except ele trata o erro impedido que o software feche e executa outra sequência de código responsável por retornar o início do loop.

4.4.1 Valores inteiros para itens extra

Apesar de ser possível a utilização do comando visto acima, na figura 7, as validações de números inteiros Inicialmente foi desenvolvido com base em outro conceito semelhante ao visto em validação de nome do doador, onde neste foi usado a função "isdigit" onde esse comando só aceita valores numéricos inteiros. Foi optado por conservar esse código como visto na figura 9.

```

355     quant_outros = input(f"Quantos {outros} deseja doar?: ").strip()
356     while not quant_outros.isdigit():
357         quant_outros = input(f"Quantos {outros} deseja doar? (apenas números): ").strip()

```

Figura 9. Comando responsável pela validação de valores inteiros em itens extra

5. Manual de uso

Esse software foi desenvolvido na linguagem de programação Python utilizando o sistema operacional Windows da Microsoft criado em um ambiente de desenvolvimento integrado (IDE) onde se foi utilizado Visual Studio Code (VS Code) que é um editor de códigos desenvolvido pela própria Microsoft, apesar de ser devolvido no sistema operacional Windows o software é capaz de ser executado no sistema operacional Linux.

Ao iniciar o software aparecer aviso com a seguinte mensagem “Esse código é desenvolvido na linguagem Python, linguagem essa que tem como fundação a língua inglesa, ou seja, as operações matemáticas e entrada de dados vai respeitar as normas da língua inglesas onde os números decimais utilizam um ponto ".", e não uma vírgula ",", para separar a parte inteira da parte decimal.” acompanhado da mensagem dizendo “Press Enter para continuar” onde o usuário deverá apertar a tecla “Enter” do computador para prosseguir, essa mensagem aparecerá em várias áreas do software com a mesma função, logo em seguida a mensagem de boas vindas aparece e o mesmo procedimento deve ser feito. veja na figura 10.

```

=====
                        ATENÇÃO!

    Esse código é desenvolvido na linguagem Python, linguagem essa que tem como fundação
    a língua inglesa, ou seja, as operações matemáticas e entrada de dados vai respeitar
    as normas da língua inglesas onde os números decimais utilizam um ponto ".", e não uma vírgula ",",
    para separar a parte inteira da parte decimal.

    Press Enter para continuar
=====

                        Seja Bem-vindo
                        Ao
    Sistema de Gerenciamento de Doações e Montagem de Cestas Básicas.
                        Dispensário Santana

    Press Enter para continuar █

```

Figura 10. Mensagem de aviso e de boas vindas

5.1. Menu

Após as mensagens exibidas, o software mostra o menu principal como mostra a figura 11.

```
=====
--===== MENU =====

Digite 1 para doações
Digite 2 para ver o relatorio
Digite 3 para encerrar
>>> |
```

Figura 11. Menu principal

O usuário deve digitar os valores correspondentes ao que deseja fazer, onde se digitado “1” será encaminhado à seção de cadastro de doações, “2” será encaminhado para o relatório parcial, “3” para um submenu de confirmação para fechar o programa como mostra a figura 12, onde se o usuário digitar “S” será exibido o relatório final e logo em seguida o software será fechada, caso contrária, se “N” for digitado o menu principal será mostrado novamente.

```
=====
                ATENÇÃO
    Todos os dados serão perdidos

Deseja continuar [S/N]: |
```

Figura 12. Submenu de confirmação

5.2. Cadastro doações

Se o usuário digitar “1” no menu principal ele será encaminhado ao cadastro de doação, onde deve se inserir o nome do doador e o tipo de pessoa, física ou jurídica, como é mostrado na figura 13.

```
=====
Nome do doador: Anônimo

=====

Digite [ F ] para pessoa Física
Digite [ J ] para pessoa Jurídica: |
```

Figura 13. Cadastro de nome e tipo de pessoa

Em seguida deverão ser inseridos os valores correspondentes a quantidade de item doado na sua respectiva área como mostrado na figura 14.


```
===== Anônimo =====  
  
Quantos kg de açúcar você deseja doar: 1  
Quantos kg de arroz você deseja doar: 4  
Quantos kg de café você deseja doar: 2  
Quantos extrato de tomate você deseja doar: 2  
Quantos macarrão você deseja doar: 3  
Quantos pct de bolacha você deseja doar: 1  
Quantos litros de óleo você deseja doar: 1  
Quantos kg de farinha de trigo você deseja doar: 1  
Quantos kg de feijão você deseja doar: 4  
Quantos kg de sal você deseja doar: 1
```

Figura 14. Área de doação

Depois questiona se o usuário gostaria de doar um item extra, que não foi mencionado acima, como visto na figura 15.

```
-----  
Gostaria de doar algo que não foi mencionado acima  
Digite [ S ] para sim e [ N ] para não: █
```

Figura 15. Questionamento sobre o item extra

Se o usuário digitar “S” será direcionado a seção de doação de extra, onde deve inserir a quantidade de novos produtos em seguida o nome do produto e logo depois a quantidade do produto, como mostra a figura 16.

```
-----  
Quantos produtos novos gostaria de adicionar: 1  
1ª O que deseja doar?: Milho de pipoca  
Quantos Milho de pipoca deseja doar?: 1
```

Figura 16. Doação de item extra

Após finalizar o questionamento da figura 15 aparecer novamente. Se inserir “N” uma mensagem de agradecimento aparecerá como mostra a figura 17. Depois é retomado o menu principal.

```
-----  
  
"Aquele pessoa que ajuda os outros simplesmente porque deveria ou precisa ser feito,  
e porque é a coisa certa a fazer, é sem dúvida, um super-herói de verdade."  
  
-Stan Lee  
  
Agradecemos sua doação Anônimo  
Press Enter para continuar █
```

Figura 17. Mensagem de agradecimento

5.3. Relatórios

Para ver o relatório parcial basta digitar “2” no menu principal e o relatório será exibido a seguir e para ver o relatório final é necessário digitar “3” no menu principal e confirmar digitando “S” no submenu de confirmação visto na figura 12 logo em seguida o relatório final é mostrado e o código é encerrado. Imagem do relatório a seguir na figura 18, observação tanto o relatório final como o parcial tem a mesma estrutura e são exibidos um topico abaixo do outro e não ao lado como na figura 18.

TOTAL		TOTAL POR PESSOA		ITENS QUE SOBRARAM		CESTAS BÁSICAS	
Itens	Arrecardado	Física		Itens		Cestas	
Acúcar	0 kg	Quilos	0	Acúcar	0 kg	C/extra	0
Arroz	0 kg	Litros	0	Arroz	0 kg	S/extra	0
Café	0 kg	Pacotes	0	Café	0 kg	Total	0
Extrato de tomate	0 un	Unidades	0	Extrato de tomate	0 un		
Macarrão	0 un	Extras	0	Macarrão	0 un		
Bolacha	0 pct	Jurídica		Bolacha	0 pct		
Óleo	0 L	Quilos	0	Óleo	0 L		
Farinha de trigo	0 kg	Litros	0	Farinha de trigo	0 kg		
Feijão	0 kg	Pacotes	0	Feijão	0 kg		
Sal	0 kg	Unidades	0	Sal	0 kg		
Outros	0	Extras	0	Outros	0		

Figura 18. Relatório

5.4. Observações

Nas seções de relatório pode ocorrer um erro na formatação da tabela, se um determinado número de casas decimais for excedido, nos tópicos de itens arrecadados e que sobraram os itens açúcar, arroz, café, bolacha, óleo, feijão, sal e itens extras o limite é de 33 casas decimais já nos itens extrato de tomate e farinha de trigo limite é de 17 casas decimais e macarrão 25 casas decimais em pessoas físicas e jurídicas os quilos, litros, pacotes e itens extras são 36 casas decimais e unidades 28 casas decimais no tópico de cestas básicas, com extra, sem extra e total todos com 36 casas decimais. Como visto na figura 19.

TOTAL	
Itens	Arrecardado
=====	
Açúcar	1.2345678901234568e+35 kg
Arroz	1.2345678901234568e+33 kg
Café	1.2345678901234569e+32 kg
Extrato de tomate	1234567890123456789012345678901 un
Macarrão	123456789012345678901234567890123456 un
Bolacha	123456789012345678901234567890 pct
Óleo	1.2345678901234568e+39 L
Farinha de trigo	1.2345678901234568e+37 kg
Feijão	1234567890.0 kg
Sal	12345.0 kg
Outros	0
=====	

Figura 19. Relatório com erro de formatação

5.5. Citação no código

Como forma de agradecer a doação recebida, o software mostra na tela uma frase do escritor Stan Lee, que diz "Aquelela pessoa que ajuda os outros simplesmente porque deveria ou precisa ser feito, e porque é a coisa certa a fazer, é sem dúvida, um super-herói de verdade.".

5.6. Easter eggs

Easter eggs, também conhecido como ovos de páscoa, são funções escondidos dos usuários onde por meio de uma palavra chave ou se uma sequência exatas de caracteres for digitada uma parte secreto do código será exibido e se no Submenu de confirmação, visto na figura 12, for digitado a palavra “criador” será exibido na tela o nome do criador do código, o tutor responsável e os colegas de sessões.

6. Teste

Para testar o programa foram feitos dois cadastros, um para cada tipo de pessoa sendo uma pessoa física e uma pessoa jurídica, onde se foram inseridos valores imaginários apenas para teste e verificação do funcionamento do código, como mostra a tabela 1 .

Tabela 1. Entrada de testes

TIPO DE PESSOA	AÇÚCAR	ARROZ	CAFÉ	EXTRATO DE TOMATE	MACARRÃO	BOLACHA	ÓLEO	FARRINHA DE TRIGO	FEIJÃO	SAL	EXTRAS
FÍSICA	5.500	3	0	6	2	0	2	1	1.500	2	1
JURÍDICA	2.500	7	5	4	7	13	2	6	9.500	5	-

Como os requisitos para as montagens das cestas são 1 kg de açúcar, 4 kg de arroz, 2 kg de café, 2 unidades de extrato de tomate, 3 unidades de macarrão, 1 pacote de bolacha, 1 L de óleo, 1 kg de farinha de trigo, 4 kg de feijão, 1 kg de sal e 1 produto extra porém o item extra não é obrigatório ou seja só será adicionado a cesta se tiver , com essas entradas é esperado que o programa exiba 2 cestas no total onde uma possui item extra e a outra não. Podemos ver o resultado do código exibido na figura 20.

TOTAL		TOTAL POR PESSOA		ITENS QUE SOBRARAM		CESTAS BÁSICAS	
Itens	Arrecadado			Itens		Cestas	
=====	=====	===== Física =====		=====		=====	
Açúcar	8.0 kg	Quilos	13.0	Açúcar	6.0 kg	C/extra	1
Arroz	10.0 kg	Litros	2.0	Arroz	2.0 kg	S/extra	1
Café	5.0 kg	Pacotes	0	Café	1.0 kg	Total	2
Extrato de tomate	10 un	Unidades	8	Extrato de tomate	6 un		
Macarrão	9 un	Extras	1	Macarrão	3 un		
Bolacha	13 pct	===== Jurídica =====		Bolacha	11 pct		
Óleo	4.0 L	Quilos	35.0	Óleo	2.0 L		
Farinha de trigo	7.0 kg	Litros	2.0	Farinha de trigo	5.0 kg		
Feijão	11.0 kg	Pacotes	13	Feijão	3.0 kg		
Sal	7.0 kg	Unidades	11	Sal	5.0 kg		
Outros	1	Extras	0	Outros	0		
=====	=====			=====			

Figura 20. Resultado obtido das entradas testes

7. Conclusão

Todos os objetivos propostos foram cumpridos com eficácia, buscando uma maior acessibilidade para o usuário, com textos simples e intuitivos, para garantir que o programa não feche acidentalmente por um erro de digitação foram feitas validações nas entradas para que assim não se perca os dados de doações e ninguém seja prejudicado.

Vale ressaltar sobre os limites de casas decimais que podem desconfigurar a formatação dos relatórios, e o valores incorretos dos produtos que aceitam valores reais. Já que os produtos que aceitam inteiro, ocorre apenas um deslocamento da tabela. Porém, esse limite pode ser aumentado se houver uma necessidade. Para isso basta aumentar as linhas da tabela onde estão em 45 casas decimais e em seguida ajustar os limites dos produtos expostos no relatório.

8. Referências

<https://www.google.com/amp/s/www1.folha.uol.com.br/amp/mercado/2020/08/quase-a-metade-dos-brasileiros-viu-renda-familiar-diminuir-na-pandemia-diz-datafolha.shtml>, Acesso em: 29 mar. de 2022

<https://www.google.com/amp/s/veja.abril.com.br/economia/ibge-desemprego-durante-a-pandemia-foi-maior-que-o-estimado/amp/>, Acesso em: 01 abr. de 2022

<https://www.pensador.com/frase/MjQ0MzY3Mg/>, Acesso em: 03 abr. de 2022