

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

ANA CAROLINA MORELLI CHAVES 23017617

CICERA EDUARDA DA COSTA 23016727

LUIZ GUSTAVO PINTO DA SILVA 23013028

**ALGORITMOS DE PROGRAMAÇÃO, PROJETOS E COMPUTAÇÃO - PRÁTICA:
PROJETO 01**

CAMPINAS

2023

SUMÁRIO

1. INTRODUÇÃO	3
2. OBJETIVOS	4
3. APRESENTAÇÃO DO PROJETO	5
4. ESCOPO E NÃO ESCOPO	5
4.1. ORGANIZAÇÃO DO PROJETO	6
5. DESENVOLVIMENTO DO PROJETO	7
6. TESTES	8
6.1. TESTES REALIZADOS SEM A INDUÇÃO DE ERROS	8
6.2. TESTES REALIZADOS SEM A INDUÇÃO DE ERROS BRUSCOS	8
6.3. TESTES REALIZADOS COM INDUÇÃO DE ERROS BRUSCOS	9
7. DESAFIOS	10
8. CONCLUSÃO	10
9. REFERÊNCIA BIBLIOGRÁFICA	12

1. INTRODUÇÃO

O mercado é um ambiente que movimenta um grande volume de dinheiro diariamente, e para que seja possível gerenciar essa movimentação financeira de forma eficiente e organizada, é importante contar com um sistema de caixa eficiente e confiável. A criação de um caixa de mercado em Python é um projeto que visa a automatização de um processo comum em estabelecimentos comerciais. Com o uso desta linguagem de programação, é possível desenvolver um software que facilite a gestão de vendas, gerenciamento de caixa e demais controles necessários no dia a dia .

Para o desenvolvimento deste projeto, foi necessário ter conhecimento básico em programação estruturada, sem a utilização de bibliotecas, conhecimento de git e GitHub. Além disso, é importante ter um entendimento dos processos de venda e funcionamento básico de um mercado.

Com a implementação deste software, espera-se aumentar a eficiência no gerenciamento de vendas, reduzindo erros e otimizando processos. Além disso, é possível a ampliação do programa para obter dados importantes sobre o comportamento do consumidor, como produtos mais vendidos e horários de maior movimento. Para o desenvolvimento deste projeto, pode-se utilizar metodologias ágeis de desenvolvimento de software, como Scrum ou Kanban. É importante também a utilização de ferramentas de versionamento de código, como Git.

Este projeto tem como objetivo apresentar a implementação de um sistema de caixa de mercado em Python. O sistema de caixa em questão terá como funcionalidades básicas a abertura através de senha, inserção de vendas, cálculo do valor total da venda, troco a ser devolvido, fechamento do caixa com retorno do número total de vendas, valor total de vendas e troco restante.

2. OBJETIVOS

Como objetivo geral tem-se gerar um sistema em Python capaz de verificar a veracidade da senha do caixa, permitir a inserção de valores dos itens comprados, calcular o valor a ser pago pelo cliente, realizar o cálculo para o troco e exibição de células que compõem o troco e então o fechamento do caixa.

Como objetivos específicos tem-se gerar um código organizado, funcional e documentado que possibilite sua ampliação e manutenção de maneira fácil e rápida que atenda as necessidades propostas inicialmente.

3. APRESENTAÇÃO DO PROJETO

O programa possui como finalidade automatizar o processo de venda de um caixa de supermercado, através da execução de funções cotidianas neste ambiente. Possibilitando assim, maior agilidade, confiabilidade e segurança no processo de vendas. Para realizar seu desenvolvimento, foi utilizada a linguagem de programação “Python”, o ambiente de desenvolvimento Visual Code, e GitHub para versionar e compartilhar o programa entre os membros do grupo. Com estas ferramentas, foi realizada a criação do software para caixa de supermercado, tendo como suas funcionalidades principais: A liberação do caixa, soma dos itens comprados, valor total, troco e fechamento.

A liberação do caixa conta com uma senha, que permite o acesso ao sistema, contribuindo para a segurança. A entrada de dados é realizada através da inserção das compras feitas por cada cliente e ao final calculando o valor total que o cliente deverá pagar. Nesta etapa ocorrerá também a inserção do valor pago pelo cliente e o cálculo para a devolução do troco que o caixa deverá retornar ao cliente. Deixando explícito a quantidade de cédulas que o caixa deve dar ao cliente; e ao final, o fechamento do caixa, contabilizando o total de itens vendidos pelo mercado no dia, valor total de vendas daquele dia e o valor total de troco existente no caixa.

4. ESCOPO E NÃO ESCOPO

Neste projeto será realizado uma aplicação feita em Python, que irá fazer a liberação do caixa, soma dos itens comprados, valor total, troco e seu fechamento. Este programa facilitará ao usuário o entendimento do procedimento de venda e cobrança do estabelecimento, de maneira simples e direta. O projeto apresentará ainda seu desenvolvimento pautado na proposta estabelecida pelo cliente, em que utilizará somente ferramentas de ATRIBUIÇÃO, SELEÇÃO, REPETIÇÃO (WHILE e/ou FOR) e STRING, INT e FLOAT. Contará ainda com uma senha pré determinada de 4 dígitos e valor máximo para troco de 300 reais.

Neste projeto não serão contemplados, recuperação de senha, inserção de dados do cliente, cálculo de impostos, armazenamento de dados e interface.

4.1. ORGANIZAÇÃO DO PROJETO

Como citado no item **3. APRESENTAÇÃO DO PROJETO**, o projeto foi organizado no GitHub, uma ferramenta que possibilita o backup e o versionamento do projeto, ou seja, realiza o gerenciamento das versões do código. Tendo em vista o desenvolvimento gradual das etapas, cada integrante realizou o compartilhamento da etapa desenvolvida com os demais integrantes para aprimoramentos, correções de bugs e melhorias em questão de lógica. Abaixo segue o endereço do repositório:

<https://github.com/Luyzao/PUC-23.git>

senha projeto: 1234

A organização do projeto foi subdesenvolvido nas etapas descritas abaixo:

- **PRIMEIRA ETAPA - RECEBIMENTO E LEITURA DO DOCUMENTO:** Após a disponibilização do documento inicial de proposta de projeto, contendo as informações da atividade, foi realizado a leitura do mesmo para entendimento das funções requeridas pelo cliente.
- **SEGUNDA ETAPA - TESTE DE MESA:** Após entendimento sobre as diretrizes do projeto, foi realizado um esquema contendo a lógica inicial do programa. Nesta etapa o projeto foi desen
- **TERCEIRA ETAPA - CRIAÇÃO DO REPOSITÓRIO:** Nesta etapa foi criado o repositório no GitHub.
- **QUARTA ETAPA - PROGRAMAÇÃO:** Início das atividades de programação para o desenvolvimento das partes, como validação da senha do caixa, leitura das vendas, cálculo do valor total, o valor do troco, e finalização do caixa. Durante esta etapa ocorreu a estruturação da documentação do projeto;
- **QUINTA ETAPA - TESTE DO PROGRAMA:** Após a finalização da programação, testes foram realizados para a verificação da funcionalidade básica do programa, erros encontrados e testes realizados estão descritos no item **6.TESTES**;
- **SEXTA ETAPA - CORREÇÃO DE BUGS:** Nesta etapa arrumamos os erros encontrados no programa, realizou-se as mudanças necessárias

para a melhoria do código. Durante o desenvolvimento desta etapa, ocorreu o alinhamento com o cliente sobre demais dúvidas.

- **SÉTIMA ETAPA - FINALIZAÇÃO:** Finalização da parte estrutural e lógica de programação, formatação e checagem de variáveis e comentários, possibilitando futuras manutenções e implementações de forma simples. Na documentação, relatou-se o desenvolvimento do projeto e o detalhamento do programa em si.

5. DESENVOLVIMENTO DO PROJETO

Como citado anteriormente, o desenvolvimento foi realizado na linguagem 'Python' – 3.10.10, contando com o IDE (ambiente de desenvolvimento integrado) Visual Code - 17.5. Vale destacar, a utilização apenas de comandos básicos (for, range, if, elif, else, while, break) sem a utilização de classes, funções, listas, dicionários, etc. Como relatado no item anterior **4. ORGANIZAÇÃO DO PROJETO**, o mesmo foi dividido em etapas, abaixo segue o detalhamento do desenvolvimento da etapa da quarta, referente a programação:

PRIMEIRO PASSO - ABRIR O CAIXA: Neste passo, tivemos que criar uma senha e pedir ao usuário para inseri-la, tendo direito a três tentativas caso o usuário erre. Aqui também já informamos que o caixa quando aberto recebeu R\$ 1280,00 para o troco ser distribuído entre cédulas indicadas no pdf;

SEGUNDO PASSO - LEITURA DAS VENDAS: Aqui, o programa começa já a contar os itens e realizar a conta total da compra, ou seja, ele pede que o caixa insira todos os valores, de cada item comprado. Para fechar a compra, basta digitar '0' ou para corrigir um valor, basta digitar '-1';

TERCEIRO PASSO - FINALIZAÇÃO DA VENDA E CÁLCULO DO TROCO: Aqui as informações colocadas no segundo passo vão ser processadas, e a conta do total da compra será feita e o sistema vai pedir para inserir o valor pago ao caixa, irá processar e fazer o cálculo do troco e assim informará ao caixa o tanto de troco que deve ser dado, especificando as cédulas que devem ser dadas no troco;

QUARTO PASSO - FECHAMENTO DO CAIXA: Aqui nós realizamos o fechamento do caixa, ou seja, após cada compra o caixa pergunta ao usuário se ele quer fechar, se a resposta for "não", o programa continua, mas se a resposta for

“sim”, o programa processa tudo, e fecha, informando a quantidade de clientes atendidos, o valor total das vendas, o valor existente no caixa e o tanto de cédulas que há no caixa, e assim o caixa fecha;

6. TESTES

Durante a execução deste projeto foram realizados testes básicos para garantir uma entrega com menor incidência de erros. Melhorando assim a experiência do usuário e performance do programa. Abaixo segue a lista que foi executada, respectivamente sua resposta e resolução.

6.1. TESTES REALIZADOS SEM A INDUÇÃO DE ERROS

Os testes abaixo ocorreram com a inserção de todas as variáveis de maneira adequada, sem a inserção de letras em campos numéricos, por exemplo.

- 1- Digitar a senha correta - Programa segue o fluxo esperado
- 2 - Valor inserido do item 1 até item 13 - Programa segue o fluxo esperado
- 3 - Valor total da compra 1470 reais - Programa realiza a soma corretamente
- 4 - Valor digitado para troco 1670 - Programa realiza a captura corretamente
- 5 - Valor do troco 200 reais - Programa calcula corretamente
- 6 - Notas a serem devolvidas 1 de 200 reais - Programa informa nota corretamente
- 7- Deseja fechar o caixa? s - Programa segue o fluxo esperado
- 8 - Fechamento realizado - Programa segue o fluxo esperado
- 9- Informações - Programa retorna número de cliente correto, quantidade de notas correta (todas as notas aparecem)
- 10- Número de clientes correto - Fluxo correto
- 11- Valor total das vendas correto - Fluxo correto

6.2. TESTES REALIZADOS SEM A INDUÇÃO DE ERROS BRUSCOS

Os testes abaixo ocorreram com a inserção de todas as variáveis de maneira tentar induzir erros, sem a inserção de letras em campos numéricos, por exemplo. Mas solicitando alteração de pequenos erros realizados pelo usuário.

- 1 - Digitar senha incorreta 1x e 2x - Programa informa número de tentativas

- 2 - Ao digitar a senha correta em qualquer uma das tentativas programa prossegue
- 3 - Solicitar valor do 1 item - Programa segue fluxo esperado
- 4- Valor negativo qualquer inserido - Retorna valor invalido - Programa segue fluxo esperado
- 5 - Item = -1. Solicita valor corrigido do item anterior - Programa segue fluxo esperado
- 6 - Continua com numeração de item correta após correção
- 7 - Quando inserido letra programa quebra - Corrigido com implementação de condição.
- 8 - Programa realiza a soma de todos os clientes e valores - Programa segue o fluxo esperado

6.3. TESTES REALIZADOS COM INDUÇÃO DE ERROS BRUSCOS

Os testes abaixo ocorreram com a inserção de todas as variáveis de maneira inadequada e não compatível com os campos destinados, com a inserção de letras em campos numéricos, por exemplo.

- 1- Ao digitar uma letra no campo de senha - O sistema quebra - Aviso de tipo de dados
- 2 - Ao digitar uma letra no campo de item - O sistema quebra - Mensagem de erro
- 3 - Ao digitar uma letra no campo de senha - O sistema quebra - Mensagem de erro e solicitação de dados corretos.
- 4 - Ao digitar um número na opção "Deseja Finalizar" - O sistema segue como programado
- 5 - Ao digitar uma letra no campo pagamento - O sistema quebra
- 6 - Ao digitar uma letra no campo para fazer o pagamento - O sistema quebra
- 7 - Ao digitar um número no campo fechar o caixa - O sistema segue como programado

Uma alternativa para a solução da mensagem de erro e quebra do código seria o código abaixo, figura01 através da utilização de comandos simples para validação. Entretanto, estes tipos de comandos, embora façam parte do leque de possibilidades oferecidos pelos loops While, não fazem parte do escopo estabelecido pelo projeto.

```
nota = -1
while nota < 0 or nota > 10:
    try:
        nota = int(input("Informe a nota entre 0 e 10: "))
        print("A nota foi:", nota)
        break
    except:
        print("Valor inválido")
```

figura01: exemplo de validação de dados de maneira básica

7. DESAFIOS

Durante o desenvolvimento do projeto, desafios foram surgindo e superados da melhor maneira técnica conhecida pelos integrantes, respeitando as condições de execução do projeto. O ponto principal foram dúvidas pontuais em relação ao atendimento dos requisitos propostos pelo cliente. Estas foram esclarecidas e alinhadas pelo mesmo, possibilitando o avanço.

O desenvolvimento da etapa de troco do caixa proporcionou o segundo grande desafio, a arquitetura inicial proposta não atendia o objetivo de especificação das células de troco a serem devolvidas para o cliente. Esta parte foi separada em outra *branche* para melhor entendimento e desenvolvimento até a obtenção do atendimento do requisito. A estruturação das sintaxes e indentações, devido ao número de condições aninhadas, causou problemas em diversos momentos. Um exemplo que pode ser citado, ocorreu durante o fechamento do caixa, ao fechar o caixa o mesmo retornava para a etapa de venda. Outro desafio na fase final deu-se por conta dos valores negativos adquiridos pela quantidade de células existentes. Variáveis de contagem de células disponíveis, que deveriam retornar valores como zeros e inteiros positivos, retornavam “-4”. Vale ressaltar, que todos os erros encontrados durante o período de teste (item **6.Testes**) foram corrigidos. O maior desafio foi a restrição de bibliotecas e ferramentas oferecidas pela linguagem.

8. CONCLUSÃO

Os objetivos principais foram cumpridos de acordo com as diretrizes estabelecidas pelo cliente para obtenção do produto final. O código permite ampliação e manutenção de maneira rápida e fácil. O projeto contribuiu para o aprendizado concreto e sólido de condições aninhadas com destaque para while e if's. Vale destacar, que a restrições de possibilidade de ferramentas proporcionou o desenvolvimento de raciocínio lógico bem como melhor aproveitamento das ferramentas disponíveis.

9. REFERÊNCIA BIBLIOGRÁFICA

VAN ROSSUM, Guido. The Python Language Reference. Python Software Foundation, 2019.

PRESSMAN, Roger S. Engenharia de Software: Uma abordagem profissional. McGraw-Hill, 2016.

Downey, A. B. (2012). Pense em Python: Pense como um cientista da computação. Novatec Editora.

Lutz, M. (2013). Aprendendo Python: Programação de computadores em Python Novatec Editora.

Van Rossum, G., & Drake, F. L. (2003). Python 2.2 documentation. Python Software Foundation.