Universidade Estácio de Sá

Ciências da computação/ads

UNIDADE NOVA AMÉRICA

**TRABALHO DE Rad (Python)**

**EM Desenvolvimento Rápido de Aplicações em Python**

Rio de janeiro - RJ

Abril / 2024

202202713376 – Guilherme C Dias da Silva

202303405804 - Isaac Blanco Pacheco

202303589603 - Romário Henrique de Melo Silva

202303300638 - Iury Barbosa Veggi Rodrigues

202102214238 - Gustavo Freire Meyohas

**Trabalho de Rad (Python)**

**em Desenvolvimento Rápido de Aplicações em Python**

Trabalho de Rad (Python) apresentado a Universidade Estácio de Sá, como exigência para avaliação na disciplina Desenvolvimento Rápido de Aplicações em Python

Orientador:

Prof. Ronaldo Candido dos Santos

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc167624229)

[1.1 DESCRIÇÃO DO PROBLEMA 3](#_Toc167624230)

[1.2 OBJETIVOS 3](#_Toc167624231)

[2 DESENVOLVIMENTO 4](#_Toc167624232)

[2.1 RESULTADOS 4](#_Toc167624233)

[3 CONCLUSÃO 6](#_Toc167624234)

[3.1 Etapas do RAD 6](#_Toc167624235)

[REFERÊNCIAS 12](#_Toc167624236)

# INTRODUÇÃO

Uma loja, inicialmente pequena, dependia de anotações manuais ou lembretes para catalogar suas vendas e gerenciar o estoque. À medida que o negócio cresceu, a necessidade de uma abordagem mais prática e eficiente para registrar os ativos tornou-se evidente. Para resolver esse problema, foi desenvolvido um sistema que permite registrar todas as transações diárias com apenas alguns cliques, reduzindo a possibilidade de erros e aumentando a eficiência operacional da loja.

## DESCRIÇÃO DO PROBLEMA

O uso de anotações manuais ou a confiança na memória dos funcionários tornou-se inviável à medida que a loja expandiu suas operações, resultando em uma série de problemas, incluindo conflitos, ineficiência e erros. A dificuldade em manter registros precisos e consistentes levou a conflitos sobre as quantidades de estoque e transações, enquanto o tempo gasto em anotações manuais e correções de erros prejudicou a produtividade dos funcionários. Além disso, a suscetibilidade a erros humanos na contabilização de vendas e estoque comprometeu a precisão dos registros, gerando discrepâncias que afetaram tanto o atendimento ao cliente quanto a gestão de inventário.

## OBJETIVOS

O objetivo principal do sistema é fornecer uma ferramenta que permita aos atendentes da loja registrarem de forma precisa e eficiente todas as vendas e o controle de estoque. Especificamente, o sistema permitirá que os atendentes registrem rapidamente cada venda realizada, assegurando que todas as transações sejam documentadas de maneira precisa, além de proporcionar um meio para registrar a quantidade de itens em estoque, facilitando o monitoramento e a reposição dos produtos. Além disso, o sistema busca minimizar a probabilidade de erros humanos no registro de transações e no controle de estoque, aumentando a confiabilidade dos dados.

Para facilitar o uso, o sistema contará com uma interface intuitiva que reduzirá a carga de trabalho dos atendentes e melhorará a eficiência das operações diárias, com funções de pesquisa que permitirão a localização rápida de produtos e suas respectivas informações, otimizando o atendimento ao cliente. Com a automação e digitalização do processo de registro de vendas e controle de estoque, a loja será capaz de transformar a maneira como gerencia suas operações diárias, proporcionando um ambiente mais organizado, eficiente e menos propenso a erros.

# DESENVOLVIMENTO

Primeiro foram escritos códigos com base naqueles que foram fornecidos pelo professor em sala de aula, para que pudéssemos estudar e ter uma ideia de como faríamos nosso projeto. Depois de testes e pesquisas, começamos a pensar sobre o tema, que foi decidido após uma breve discussão onde as opiniões do grupo foram unânimes. Com o tema definido, começamos a pensar nos requisitos do cliente. Trouxemos problemas, possíveis ferramentas (opcionais e fundamentais), ideias de design e algumas possíveis soluções. Após essa preparação o código começou a ser escrito, indo de um rascunho que gerava um programa com poucas funções, até aplicação de interface gráfica e banco de dados.

## RESULTADOS

O sistema, embora em desenvolvimento, cumpre com as expectativas do cliente, deixando o processo de vender mais rápido, além de um programa com interface simples e intuitiva, o que facilita que mais vendedores da loja usem, se necessário. Alguns pontos ainda não foram implementados, como lucro e número total de vendas. Há conversas onde isso procura ser resolvido, além de reflexões sobre o que falta no programa.

2.2 DISCUSSÕES

Houve discussões em relação à interface gráfica. No começo, foi cogitado o framework Flask - pois o grupo já conhecia a linguagem HTML. Porém, levando em consideração onde e para que seria usado o programa, decidimos utilizar outro (Tkinter).

2.3 RECURSOS CITADOS

**Flask**: Framework utilizado no desenvolvimento de aplicativos WEB;

**Tkinter:** é uma biblioteca padrão da linguagem de programação Python, focado na criação de aplicações desktop (dentro da máquina do usuário);

**framework:** é uma ferramenta usada pelo desenvolvedor na criação de aplicações. É utilizado também pois já conta com predefinições de coisas como janelas, visores (como da calculadora), dentre outros recursos que podem trazer velocidade de conclusão do código

# CONCLUSÃO

Um projeto de (RAD) serve para essas situações onde um problema pode ser resolvido de forma rápida, com feedback constante de quem usará a aplicação e que seja apto a funcionar em qualquer máquina, além de ter fácil interação com outros sistemas. Observando o resultado do projeto, percebe-se que os objetivos foram cumpridos. Porém ainda há muito a evoluir. Seja um pequeno detalhe ou um erro na aplicação, o programa exige constante tratamento, evolução e contato direto com o cliente.

# ETAPAS DO RAD

**Modelagem de Negócio:**

O primeiro passo para o desenvolvimento rápido (RAD) deste sistema foi a modelagem do negócio, que envolveu a compreensão detalhada das operações da loja e das necessidades dos usuários. A loja, inicialmente pequena, dependia de anotações manuais para catalogar vendas e gerenciar o estoque. À medida que a loja cresceu, essas práticas se tornaram ineficazes, resultando em erros e ineficiências. Portanto, o objetivo principal do sistema é registrar transações diárias e gerenciar o estoque de maneira mais prática e eficiente, reduzindo a probabilidade de erros humanos.

Para alcançar isso, identificamos os principais processos de negócio, que incluem o registro de vendas, o gerenciamento do estoque e a autenticação de usuários. Os atores principais são os atendentes, que registrarão vendas e atualizarão o estoque, e os gerentes, que monitorarão as operações e gerarão relatórios.

**Modelagem de dados:**

A modelagem de dados foi a etapa seguinte, na qual estruturamos as informações essenciais para o funcionamento do sistema. Criamos duas tabelas principais: usuários e produtos. A tabela usuário armazena as credenciais de login (id, username, password), enquanto a tabela produtos guarda informações sobre os produtos disponíveis na loja (id, nome, categoria, preço).

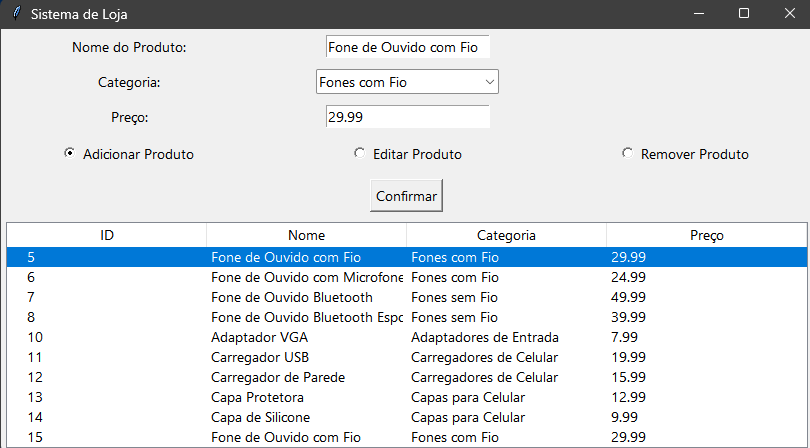
Essas tabelas foram definidas para suportar as funcionalidades de autenticação e gerenciamento de produtos. A integridade e a segurança dos dados são garantidas através da utilização de hashing (SHA-256) para as senhas dos usuários. Além disso, implementamos mecanismos para verificar e inserir dados fictícios, garantindo que haja informações iniciais suficientes para testar o sistema.



**Design da Interface:**

O design da interface foi uma parte crucial do desenvolvimento, garantindo que o sistema fosse intuitivo e fácil de usar. Criamos uma interface gráfica utilizando o Tkinter, que inclui uma tela de login e uma tela principal para o gerenciamento de produtos.

A tela de login permite que os usuários insiram seu nome de usuário e senha. Se as credenciais forem válidas, o usuário é redirecionado para a tela principal. A tela principal contém formulários para adicionar, editar e remover produtos, bem como uma tabela que exibe todos os produtos no estoque. A interface também inclui funcionalidades de pesquisa para facilitar a localização de produtos específicos.



**Implementação e Testes:**

A implementação começou com a codificação das funcionalidades principais, conforme descrito no código fornecido. Implementamos a autenticação de usuários, onde as senhas são armazenadas de forma segura utilizando hashing. A tela principal do sistema permite realizar operações de CRUD (criar, ler, atualizar, deletar) para os produtos, e a funcionalidade de pesquisa foi adicionada para melhorar a eficiência na localização de produtos.

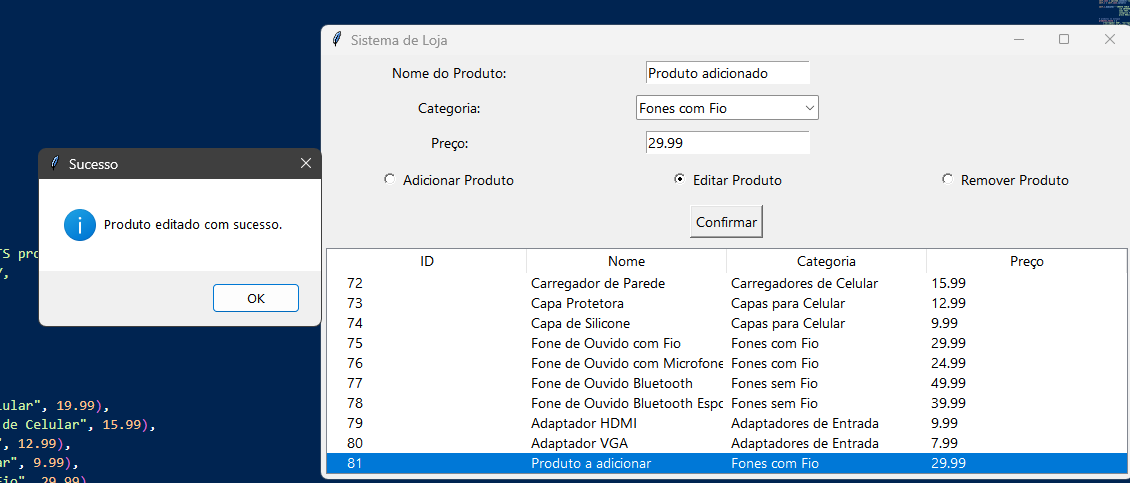
Após a implementação, passamos para os testes. Primeiramente, escrevemos testes unitários para cada função e método do sistema, utilizando a biblioteca unittest em Python. Esses testes garantiram que cada componente individual do sistema funcionasse conforme o esperado. Por exemplo, testamos o processo de login com credenciais válidas e inválidas para verificar a robustez da autenticação.

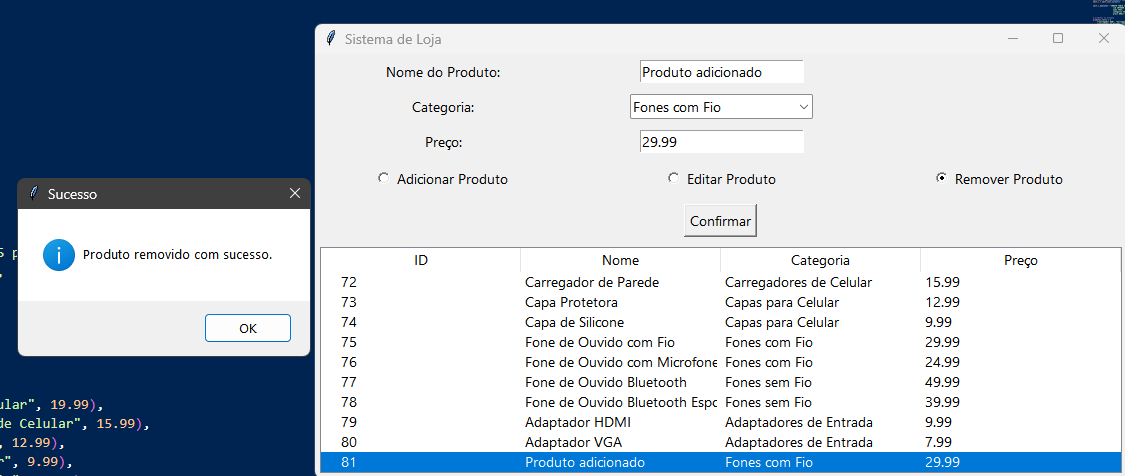
Em seguida, realizamos testes de integração para assegurar que os diferentes módulos do sistema funcionassem bem juntos. Testamos fluxos completos, como o login de um usuário, a adição de um produto, a pesquisa de produtos e a edição de informações de produtos. Esses testes foram cruciais para identificar e corrigir quaisquer problemas de interação entre os componentes do sistema.

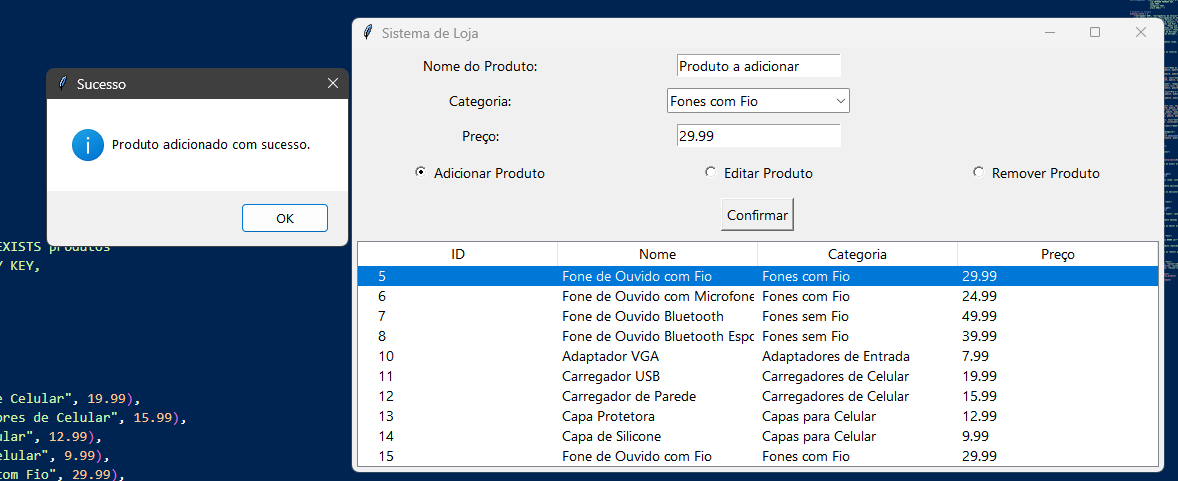
Não realizamos testes de aceitação com usuários reais nesta fase inicial. No entanto, colocamo-nos à disposição para coletar feedbacks dos atendentes e gerentes após o sistema entrar em operação. Esse feedback será essencial para realizar ajustes e melhorias no código, garantindo que o sistema atenda plenamente às necessidades dos usuários finais e opere de maneira eficiente no ambiente da loja.

**Demonstrativo da Aplicação:**

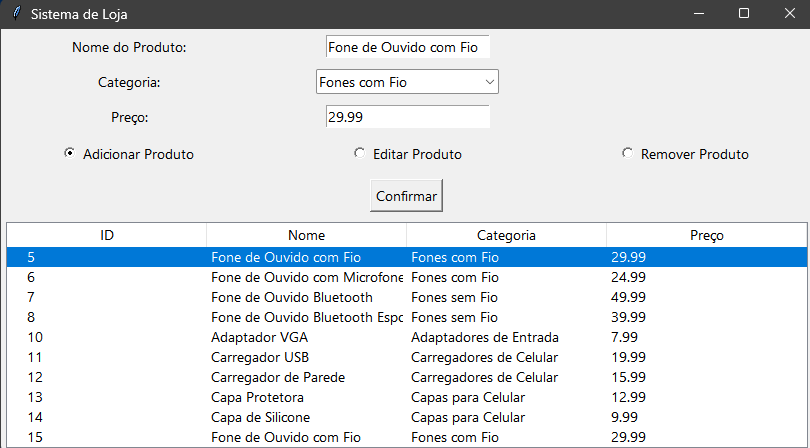
Resumo: Um gerenciador de produtos, onde podemos adicionar, editar e excluir produtos do nosso estoque, assim fazendo um quantitativo diário dos produtos dentro da lojá.







Os produtos criados ficam dentro de sua própria categoria e com o seu preço a amostra para ser consultado de maneira rápida, também foi adicionada uma ferramenta de pesquisa, para que o usuário consiga mais rapidamente localizar o produto dentro do seu estoque.



Foi elaborado uma tela de login de usuário, por questão de segurança e controle, onde o login e a senha estão criptografadas dentro do banco de dados.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Dentro do código a tabela é criado caso não exista, dando nome as suas tabelas e adicionando a qual será a chave primeira desse amontoado de dados que iram entrar na tabela.

Texto

Descrição gerada automaticamente

# REFERÊNCIAS

Como Criar uma Tela em Python Para Seus Códigos - [Interface Gráfica Intuitiva com Tkinter]. Disponível em: https://www.youtube.com/watch?v=AiBC01p58oI. Acesso em: 14 maio. 2024.

KOFFMAN, E. B.; WOLFGANG, P. A. T. Abstração, Estruturas de Dados e Projeto Usando C++. Rio de Janeiro: LTC, 2008. Capítulo 10: Ordenação.

Rapid Application Development. Disponível em: https://blog.cronapp.io/como-aplicar-o-rad-no-desenvolvimento-de-softwares. Acesso em: 14 maio. 2024.

SQL Tutorial. Disponível em: https://www.w3schools.com/sql/default.asp

. Acesso em: 14 maio. 2024.

Python GUI Programming With Tkinter. Disponível em: https://realpython.com/python-gui-tkinter. Acesso em: 14 maio. 2024.