Universidade Estácio de Sá

curso Desenvolvimento Rápido de Aplicações em Python UNIDADE West shopping

**TRABALHO DE Gerenciador de tarefas**

**EM Desenvolvimento Rápido de Aplicações em Python**

Rio de Janeiro - RJ

abril / 2024

202302202829 – Gabriel Davi Tavares do Couto

**Trabalho de gerenciador de tarefas**

**em Desenvolvimento Rápido de Aplicações em Python**

Trabalho de gerenciador de tarefas 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](#_Toc168056933)

[1.1 DESCRIÇÃO DO PROBLEMA 3](#_Toc168056934)

[1.2 OBJETIVOS 3](#_Toc168056935)

[2 DESENVOLVIMENTO 4](#_Toc168056936)

[2.1 METODOLOGIA 4](#_Toc168056937)

[2.2 DISCUSSÕES 5](#_Toc168056938)

[2.3 GERENCIADOR DE TAREFAS 5](#_Toc168056939)

[2.4 BARRA DE TAREFAS 5](#_Toc168056940)

[2.5 BARRA DE PRIORIADE 6](#_Toc168056941)

[2.6 ADICIONAR E EXCLUIR 6](#_Toc168056942)

[3 CONCLUSÃO 8](#_Toc168056943)

[REFERÊNCIAS 10](#_Toc168056944)

# INTRODUÇÃO

Este projeto apresenta um gerenciador de tarefas desenvolvido para auxiliar na administração das atividades diárias em uma casa. A motivação para este tema surge da necessidade de simplificar o processo de organização das tarefas domésticas, permitindo uma melhor gestão do tempo e aumentando a produtividade dos usuários.

## DESCRIÇÃO DO PROBLEMA

Atualmente, muitas residências enfrentam desafios significativos relacionados à organização e eficiência no gerenciamento das atividades diárias. A falta de um sistema funcional e eficiente para organizar tarefas pode levar a problemas de organização, resultando em atrasos na realização de atividades, esquecimentos de compromissos importantes e um ambiente desorganizado que pode causar estresse e frustração aos moradores. Diante desse cenário, torna-se evidente a necessidade de uma solução prática e acessível que permita às famílias administrarem suas tarefas de forma mais eficiente, promovendo uma rotina doméstica mais organizada e tranquila.

## OBJETIVOS

O principal resultado almejado neste trabalho é a criação de um sistema em Python capaz de gerenciar as tarefas do usuário, desde as mais simples até as mais complexas. Através deste sistema, busca-se alcançar uma maior eficiência na administração do tempo, permitindo ao usuário otimizar suas atividades diárias, sejam elas de lazer ou domésticas. O sistema visa proporcionar uma experiência intuitiva e funcional, facilitando a organização e execução das tarefas, contribuindo assim para uma rotina mais produtiva e equilibrada.

# DESENVOLVIMENTO

O desenvolvimento do projeto de Gerenciador de Tarefas para desktop utilizando Python e SQLite é um esforço voltado para proporcionar uma ferramenta eficiente e fácil de usar para a administração das atividades diárias em um ambiente doméstico. A seguir, detalhamos as etapas principais do desenvolvimento, desde a criação dos componentes visuais até a configuração do banco de dados, incluindo a metodologia utilizada e as discussões pertinentes.

## 2.1 METODOLOGIA

A metodologia adotada para este projeto inclui a utilização da biblioteca Tkinter para a criação da interface gráfica, e do banco de dados SQLite para armazenamento das tarefas. A interface gráfica foi projetada para ser intuitiva e responsiva, permitindo aos usuários gerenciar suas tarefas de forma eficiente.

Criação da Interface Gráfica: A interface gráfica foi desenvolvida utilizando a biblioteca Tkinter, que é a biblioteca padrão para criação de interfaces gráficas em Python. A escolha por Tkinter deve-se à sua simplicidade e integração nativa com Python, facilitando o desenvolvimento e a manutenção do sistema.

Widgets Utilizados: Foram utilizados diversos widgets, como Label, Entry, Combobox e Button, organizados dentro de um Frame centralizado para proporcionar uma experiência de usuário intuitiva.

Configuração Responsiva: A janela principal foi configurada para ser responsiva, ajustando-se automaticamente ao redimensionamento da janela do usuário.

Configuração do Banco de Dados: Para o armazenamento das tarefas, foi utilizado o SQLite, um banco de dados relacional leve e de fácil integração com Python.

Criação da Tabela: A tabela tarefas foi criada para armazenar as tarefas com os campos id, tarefa e prioridade. O campo id é a chave primária que permite identificar de forma única cada tarefa.

Operações CRUD: Foram implementadas operações de criação, leitura, atualização e exclusão (CRUD) para gerenciar as tarefas no banco de dados.

Funcionalidades Principais: O sistema inclui funcionalidades essenciais para o gerenciamento das tarefas:

Adicionar Tarefas: Permite ao usuário adicionar novas tarefas, especificando a descrição da tarefa e sua prioridade (Baixa, Média, Alta). Após a adição, a lista de tarefas é atualizada automaticamente.

Listar Tarefas: Exibe todas as tarefas armazenadas no banco de dados em um Treeview, permitindo ao usuário visualizar rapidamente suas tarefas e suas prioridades.

Excluir Tarefas: Permite ao usuário excluir tarefas selecionadas na lista, garantindo a confirmação da exclusão para evitar remoções acidentais.  
  
etapas do RAD:Identificação da necessidade de um gerenciador de tarefas eficiente para otimizar o tempo dos usuários.

Definição das funcionalidades essenciais: adicionar, listar e excluir tarefas.

Design da Interface:Criação de uma interface gráfica intuitiva utilizando Tkinter.

Definição dos widgets necessários (labels, entradas de texto, combobox, botões e treeview).

Construção Rápida:Implementação inicial da interface e funcionalidade básica de gerenciamento de tarefas.

Integração do banco de dados SQLite para armazenamento persistente.

Feedback do Usuário:Testes iniciais com usuários para coletar feedback sobre a interface e funcionalidade.

Ajustes baseados no feedback para melhorar a usabilidade e eficiência.

Finalização: Refatoração do código para maior clareza e manutenção.

Testes finais para garantir que todas as funcionalidades funcionem conforme o esperado.

2.2 DISCUSSÕES

Durante o desenvolvimento, algumas decisões importantes foram tomadas para garantir a funcionalidade e a usabilidade do sistema:

Escolha das Tecnologias: A escolha por Tkinter e SQLite foi baseada na simplicidade e eficiência dessas ferramentas para o desenvolvimento rápido e fácil de um aplicativo de desktop.

Interface Intuitiva: O design da interface focou na simplicidade e na facilidade de uso, permitindo que qualquer usuário, mesmo com pouca experiência em tecnologia, possa utilizar o sistema sem dificuldades.

Gestão Eficiente do Tempo: O sistema foi desenvolvido com o objetivo de otimizar o tempo dos usuários, proporcionando uma ferramenta prática para a organização de suas atividades diárias, tanto domésticas quanto de lazer.

Resultados

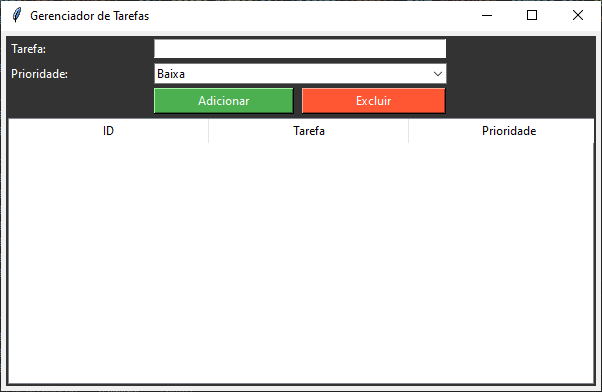
O projeto resultou em um sistema funcional de gerenciamento de tarefas que atende aos objetivos propostos:

Eficiência no Gerenciamento de Tarefas: O sistema permite a organização eficiente das tarefas diárias, ajudando os usuários a priorizar suas atividades e melhorar a produtividade.

Experiência de Usuário Positiva: A interface gráfica desenvolvida é intuitiva e de fácil navegação, proporcionando uma experiência agradável ao usuário.

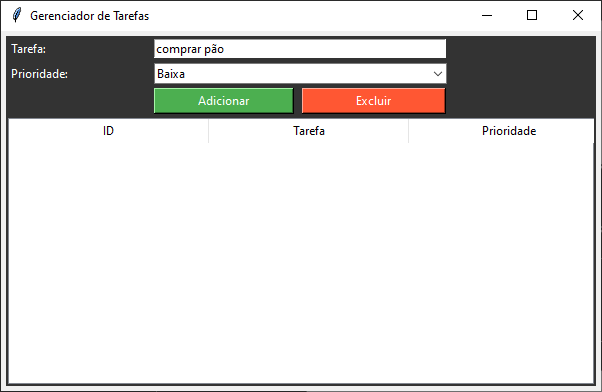
Flexibilidade e Extensibilidade: O uso de Tkinter e SQLite garante que o sistema possa ser facilmente mantido e expandido no futuro, adicionando novas funcionalidades conforme necessário.

## GERENCIADOR DE TAREFAS



## BARRA DE TAREFAS

Nessa parte você deverá escrever sua tarefa que deseja armazenar podendo ela ser tanto tarefa de trabalho quanto uma tarefa domestica como no exemplo abaixo que a tarefa foi definida com “comprar pão”



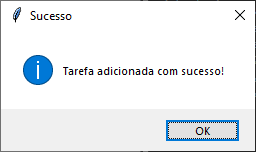
## BARRA DE PRIORIADE

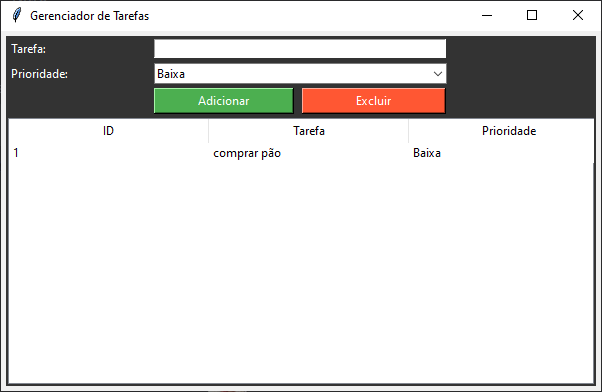
Nessa parte você deve definir a prioridade da tarefa sendo ela baixa, média ou alta

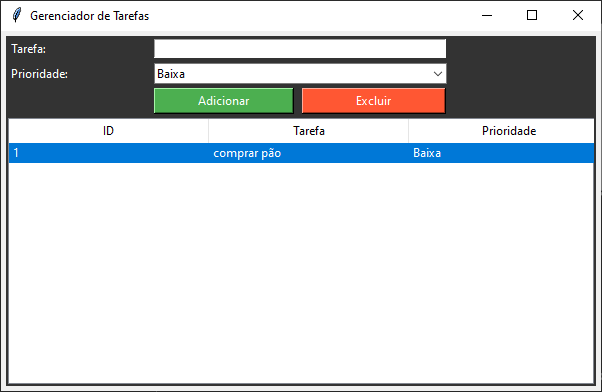


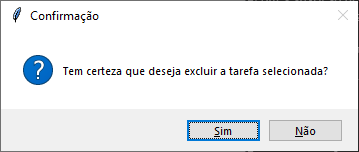
## ADICIONAR E EXCLUIR

Esse são os botões usados para que quando você tiver definido a tarefa você clique no botão adicionar e a tarefa seja adicionada com a aparição de uma caixa de mensagem confirmando que foi adicionada e quando a tarefa tiver sido concluída a tarefa possa ser apagada clicando sobre a tarefa e pressionando excluir e uma tela perguntando se deseja realmente excluir ira aparecer









# CONCLUSÃO

Na parte final deste trabalho, apresentamos as considerações finais e uma recapitulação sintética dos dados obtidos ao longo do desenvolvimento do projeto. O sistema de Gerenciador de Tarefas desenvolvido em Python, utilizando Tkinter para a interface gráfica e SQLite para o armazenamento dos dados, demonstrou ser uma solução eficaz e acessível para a organização de tarefas diárias.

Considerações Finais

Eficiência e Usabilidade: O sistema mostrou-se eficiente na gestão das tarefas, permitindo ao usuário adicionar, visualizar e excluir tarefas com facilidade. A interface gráfica intuitiva e amigável contribuiu para uma experiência de uso satisfatória, facilitando a interação do usuário com o sistema.

Organização e Produtividade: A possibilidade de definir prioridades para as tarefas ajudou na organização e no planejamento das atividades, promovendo uma melhor gestão do tempo e aumentando a produtividade dos usuários.

Tecnologias Utilizadas: A escolha de Python, Tkinter e SQLite revelou-se acertada, proporcionando um desenvolvimento rápido e uma manutenção simples do sistema. Essas tecnologias se mostraram adequadas para o escopo do projeto, garantindo uma boa performance e estabilidade.

Futuras Implementações

Embora o sistema atual atenda aos objetivos propostos, há sempre espaço para melhorias e novas funcionalidades. Futuramente, uma função de edição de tarefas será adicionada. Essa funcionalidade permitirá que os usuários modifiquem tarefas existentes, ajustando descrições e prioridades conforme necessário. Esta melhoria irá:

Aumentar a Flexibilidade: Proporcionando aos usuários a capacidade de atualizar informações sem precisar excluir e recriar tarefas.

Melhorar a Experiência do Usuário: Facilitando a gestão de tarefas em situações onde as prioridades ou detalhes mudem frequentemente.

Enriquecer as Funcionalidades: Tornando o sistema ainda mais robusto e completo, atendendo melhor às necessidades dos usuários.

Com essas considerações, conclui-se que o projeto atingiu seus objetivos iniciais, proporcionando uma ferramenta útil e eficiente para o gerenciamento de tarefas diárias. A implementação futura da função de edição de tarefas promete agregar ainda mais valor ao sistema, aprimorando sua usabilidade e funcionalidade.

# REFERÊNCIAS

Curso Python Tkinter - Entry Tkinter, com exemplo prático - João Futi Muanda. Disponível em: <https://youtu.be/FdTYu6TP3hs?si=CciVL\_DnMNWrseaW>. Acesso em: 10 mai. 2024.

Python + SQLite3: Desenvolva um app de gerenciamento de tarefas passo a passo! – Programador Aventureiro. Disponível em: <https://youtu.be/dpcj3HEKnbg?si=EyKTsraXSs9xdUuN>. Acesso em: 24 abril. 2024.

Aplicativos Python - Gerenciador de Atividades/tarefas - Thales Veloso. Disponível em: < https://youtu.be/BaPeP3v8OLU?si=lF7lvq3AxBiEUlv4>. Acesso em: 19 abril. 2024.

Visual Studio Code: Download do software. Disponível em: <https://code.visualstudio.com/>. Acesso em: 01 mai. 2024.

NORMAS ABNT. Normas para elaboração de trabalhos acadêmicos. Disponível em: <https://www.normasabnt.org/>. Acesso em: 28 mai. 2021.