

## Projeto: Criando uma Lista de Tarefas

**Objetivo:** Desenvolver uma lista de tarefas em Python para praticar conceitos de programação como classes, métodos, listas e interfaces gráficas com Tkinter. Este guia passo a passo ajudará a criar o projeto de forma clara e detalhada.

### Introdução

Neste projeto, você irá criar uma lista de tarefas que permite adicionar e remover tarefas. A lista de tarefas será estruturada usando a linguagem de programação Python e a biblioteca Tkinter para a interface gráfica.



### Passo 1: Configurando o Ambiente

#### Preparar o Ambiente de Desenvolvimento:

- Certifique-se de ter o Python instalado no seu computador.
- Instale a biblioteca Tkinter, se ainda não estiver instalada.
- Abra o VS Code ou outro editor de sua preferência para começar a codificar.

#### Explicação:

- O Python é necessário para escrever e executar o código da lista de tarefas.
- Tkinter é uma biblioteca padrão do Python usada para criar interfaces gráficas de usuário (GUIs).
- O VS Code é um editor de código que facilitará a escrita e teste do seu programa.

### Passo 2: Criando a Classe da Lista de Tarefas

#### Definir a Classe:

- Comece definindo uma classe chamada `ToDoListApp` para representar a lista de tarefas.
- Dentro da classe, crie um método `__init__` que inicializa a interface gráfica e define os elementos da interface.

#### Explicação:

- A classe `ToDoListApp` conterá métodos para configurar a interface e gerenciar as tarefas.

- O método `__init__` é um construtor que inicializa os elementos gráficos como rótulos, entradas, botões e a lista de tarefas.

### Passo 3: Configurando a Interface Gráfica

#### Criar a Janela Principal:

- Dentro da classe `ToDoListApp`, inicialize a janela principal usando `Tkinter`.
- Defina o título da janela como "Lista de Tarefas".

#### Explicação:

- A janela principal será a interface da lista de tarefas onde os elementos gráficos serão exibidos.
  - Definir um título para a janela ajuda a identificar o programa.



#### Adicionar os Elementos da Interface:

- Adicione um rótulo (Label) para solicitar a tarefa do usuário.
- Adicione uma entrada (Entry) para o usuário digitar a tarefa.
- Adicione um botão (Button) para adicionar a tarefa à lista.
- Adicione uma lista (Listbox) para exibir as tarefas.
- Adicione um botão (Button) para remover a tarefa selecionada.

#### Explicação:

- O rótulo ajuda a identificar o que o usuário deve digitar na entrada.

- A entrada permite que o usuário digite a tarefa.
- O botão "Adicionar Tarefa" aciona a função de adicionar a tarefa à lista.
- A lista exibe as tarefas adicionadas.
- O botão "Remover Tarefa" aciona a função de remover a tarefa selecionada.

### Passo 4: Criando as Funções da Lista de Tarefas

#### Função para Adicionar Tarefa:

- Crie uma função chamada `add_task` que será responsável por obter a tarefa da entrada, adicionar a tarefa à lista de tarefas e atualizar a lista exibida.

Explicação:

- Esta função obtém a tarefa digitada pelo usuário, adiciona a tarefa à lista e atualiza a exibição da lista.
- A função também limpa a entrada após adicionar a tarefa.

Função para Remover Tarefa:

- Crie uma função chamada `remove_task` que será responsável por remover a tarefa selecionada da lista de tarefas e atualizar a lista exibida.

Explicação:

- Esta função remove a tarefa selecionada pelo usuário e atualiza a exibição da lista.
- A função também trata erros caso o usuário não selecione uma tarefa, exibindo uma mensagem de aviso.

Função para Atualizar a Lista de Tarefas:

- Crie uma função chamada `update_tasks_listbox` que será responsável por atualizar a exibição da lista de tarefas na interface gráfica.

Explicação:

- Esta função atualiza a lista de tarefas exibida, garantindo que a interface gráfica reflita a lista de tarefas atual.

Passo 5: Executando o Programa Principal

Configurar o Loop Principal:

- Fora da classe `ToDoListApp`, crie uma instância da classe e inicie o loop principal da interface Tkinter para exibir a janela da lista de tarefas.

Explicação:

- O loop principal do Tkinter é necessário para manter a interface gráfica em execução e responder às ações do usuário, como pressionar botões.

Prof. Thier Alarcon