



# Implementação de uma Lista de Tarefas com Prioridade em Python

Nesta apresentação, vamos explorar a implementação de uma lista de tarefas com prioridade utilizando Python, combinando as bibliotecas heapq e Tkinter para criar uma aplicação funcional e intuitiva.  
Por: Arthur Linzing e Lorenzo Canestraro.

# Objetivo da Implementação

## Objetivo

Criar uma aplicação Python que permite organizar tarefas com diferentes níveis de prioridade.

## Funcionalidades

A aplicação oferecerá a capacidade de adicionar, remover e concluir tarefas, além de organizar as tarefas por ordem de prioridade.



# Linguagem Utilizada

## Python

Python é a linguagem de programação escolhida para este projeto, devido à sua legibilidade, simplicidade e vasto ecossistema de bibliotecas.

## Bibliotecas

Utilizaremos heapq para gerenciar a fila de prioridade, Tkinter para a interface gráfica e ttk para widgets de estilo.

# Estrutura da Fila de Prioridade

1

## Descrição

Uma fila de prioridade é utilizada para armazenar as tarefas e garantir que as tarefas com maior prioridade sejam atendidas primeiro.

2

## Implementação

A biblioteca heapq do Python oferece uma implementação eficiente de uma fila de prioridade, permitindo ordenar as tarefas com base na sua prioridade.

3

## Exemplo de Código

A função heapq.heappush adiciona um novo elemento à fila de prioridade, mantendo a ordem de prioridade.



Made with Gamma

# Interface Gráfica com Tkinter



## Campos de Entrada

Campos de entrada para o usuário inserir a descrição da tarefa e a prioridade.



## Botões

Botões para adicionar e remover tarefas da lista.



## Lista de Tarefas

Uma lista exibindo as tarefas com seus níveis de prioridade, com checkboxes para marcar as tarefas como concluídas.

# Adicionando Tarefas

1

## Função

A função `adicionarTarefa` é responsável por inserir uma nova tarefa na fila de prioridade.

2

## Descrição

A função obtém a descrição da tarefa e a prioridade do usuário e utiliza `heapq.heappush` para adicionar a tarefa à fila de prioridade.

3

## Exemplo de Código

A função utiliza os campos de entrada do usuário para obter os dados da tarefa e da prioridade e chama a função `heapq.heappush()` para adicionar a tarefa à fila de prioridade.



# Removendo Tarefas

1

## Função

A função `removerTarefa` é responsável por remover a tarefa com a maior prioridade da fila de prioridade.

2

## Descrição

A função utiliza `heapq.heappop` para remover a tarefa com a maior prioridade da fila e exibe uma mensagem informativa ao usuário.

3

## Exemplo de Código

A função chama `heapq.heappop` para remover a tarefa de maior prioridade da fila de prioridade e utiliza uma caixa de diálogo para exibir uma mensagem de sucesso ao usuário.



# Concluindo Tarefas

Função

concluirTarefa

Descrição

Marca a tarefa como concluída e a remove da lista de tarefas.

Exemplo de Código

A função recebe o índice da tarefa na lista e utiliza a função `pop` para remover a tarefa da lista.



Made with Gamma



# Atualizando a Lista de Tarefas

1

## Função

A função `atualizarListaDeTarefas` é responsável por atualizar a interface gráfica para refletir as mudanças na lista de tarefas.

2

## Descrição

A função limpa a lista de tarefas existente na interface gráfica e a recria com base nas tarefas atuais na fila de prioridade.

3

## Exemplo de Código

A função percorre as tarefas na fila de prioridade e cria checkboxes para cada tarefa, atualizando a lista de tarefas na interface gráfica.



# Conclusão

1

## Resumo

Nesta apresentação, exploramos a implementação de uma lista de tarefas com prioridade utilizando Python, combinando as bibliotecas heapq e Tkinter para criar uma aplicação funcional e intuitiva.

2

## Benefícios

A aplicação oferece um método eficiente para gerenciar tarefas, organizar suas prioridades e manter um controle organizado das tarefas.

3

## Próximos Passos

Possíveis funcionalidades adicionais incluem a capacidade de salvar e carregar listas de tarefas de arquivos, além de opções de filtragem e pesquisa.