

Projeto de Bloco: Fundamentos de Dados

[24E1_5]

Prof. Alcione Dolavale

Alberto Fontenelle Pluecker

03.03.2024



TP 1

Link github: https://github.com/albertofp/bloco5_infnet/blob/pb/pb/tp1/alberto_pluecker_tp1.py

Sumário

Este relatório descreve a implementação de um Sistema de Gestão de Tarefas em Python. O sistema permite adicionar, listar, marcar como concluída e remover tarefas, utilizando loops, manipulação de listas e funções para realizar essas operações. Também faz uso da library `datetime` para validação de datas e obtenção do dia atual ao fazer a criação de tarefas.

Funcionalidades

- Adicionar Tarefa: Permite ao usuário adicionar uma nova tarefa à lista de tarefas pendentes.
- Listar Tarefas: Mostra todas as tarefas pendentes na lista, enumerando-as.
- Marcar Tarefa como Concluída: Permite ao usuário marcar uma tarefa específica como concluída.
- Remover Tarefa: Permite ao usuário remover uma tarefa da lista.

Demonstrações

1. Estruturas de dados:

```
alberto_pluecker_tp1.py
class Urgencia(Enum):
    BAIXA = 1
    MEDIA = 2
    ALTA = 3

class Tarefa:
    def __init__(self, descricao: str, prazo_final: str, urgencia: Urgencia, data_criacao = date.today().strftime('%Y-%m-%d')):
        self.descricao = descricao
        self.data_criacao = data_criacao
        self.prazo_final = prazo_final
        self.urgencia = urgencia
        self.concluida = False
```

2. Adicionar: O usuário precisa inserir os dados necessários para instanciar uma nova Tarefa: descrição, prazo e urgência. Em seguida, é criada uma nova instância da classe Tarefa, que é adicionada à lista.

```
alberto_pluecker_tp1.py
def adicionar_tarefa(lista_tarefas: list[Tarefa], descricao: str, prazo_final: str, urgencia: Urgencia):
    """
    Adiciona uma nova tarefa à lista de tarefas.

    Args:
        lista_tarefas (list): Lista de tarefas onde a nova tarefa será adicionada.
        descricao (str): Descrição da tarefa.
        prazo_final (str, opcional): Prazo final da tarefa (formato: YYYY-MM-DD).
        urgencia (str, opcional): Nivel de urgência da tarefa.

    Returns:
        None
    """
    nova_tarefa = Tarefa(descricao, prazo_final, urgencia)
    lista_tarefas.append(nova_tarefa)
```

```

≡ Sistema de Gestão de Tarefas ≡
1. Adicionar Tarefa
2. Listar Tarefas
3. Marcar Tarefa como Concluída
4. Remover Tarefa
5. Sair
Escolha uma opção: 1
Digite a descrição da tarefa: Teste
Digite o prazo final da tarefa (formato: YYYY-MM-DD): 2024-10-10
Digite o nível de urgência da tarefa: (1 - Baixa, 2 - Media, 3 - Alta)2

```

3. Listar: Exibe todos os itens na lista de tarefas.

```

alberto_pluecker_tp1.py
def listar_tarefas(lista_tarefas: list[Tarefa]):
    """
    Lista todas as tarefas pendentes na lista.

    Args:
        lista_tarefas (list): Lista de tarefas a serem listadas.

    Returns:
        None
    """
    print("Lista de Tarefas Pendentes:")
    for idx, tarefa in enumerate(lista_tarefas, start=1):
        if not tarefa.concluida:
            print(f"{idx}. Descrição: {tarefa.descricao}")
            print(f"    Data de Criação: {tarefa.data_criacao}")
            print(f"    Prazo Final: {tarefa.prazo_final}")
            print(f"    Urgência: {tarefa.urgencia.name if tarefa.urgencia else 'Não especificada'}")
            print("")

```

```
> python alberto_pluecker_tp1.py
```

```
=== Sistema de Gestão de Tarefas ===
```

1. Adicionar Tarefa
2. Listar Tarefas
3. Marcar Tarefa como Concluída
4. Remover Tarefa
5. Sair

Escolha uma opção: 2

```
=== Tarefas Pendentes ===
```

Lista de Tarefas Pendentes:

1. Descrição: Reunião de equipe
Data de Criação: ('2024-03-03',)
Prazo Final: 2024-03-10
Urgência: MEDIA
2. Descrição: Preparar relatório mensal
Data de Criação: ('2024-03-03',)
Prazo Final: 2024-03-12
Urgência: ALTA
3. Descrição: Enviar convites para o evento
Data de Criação: ('2024-03-03',)
Prazo Final: 2024-03-08
Urgência: ALTA
4. Descrição: Revisar contrato com fornecedor
Data de Criação: ('2024-03-03',)
Prazo Final: 2024-03-15

4. Marcar como concluído: Edita o campo "concluída" em uma dada Tarefa, marcando como "concluída".

```

● ● ● alberto_pluecker_tp1.py
def marcar_tarefa_concluida(lista_tarefas: list[Tarefa], indice: int):
    """
    Marca uma tarefa específica como concluída.

    Args:
        lista_tarefas (list): Lista de tarefas onde a tarefa será marcada como concluída.
        indice (int): Índice da tarefa a ser marcada como concluída na lista.

    Returns:
        None
    """
    if 0 < indice <= len(lista_tarefas):
        lista_tarefas[indice-1].concluida = True
    else:
        print("Índice inválido!")

```

10. Descrição: Resolver problemas de TI
 Data de Criação: ('2024-03-03',)
 Prazo Final: 2024-03-11
 Urgência: ALTA

Digite o índice da tarefa a ser marcada como concluída: |

5. Remover: Remove a tarefa com ID selecionado da lista.

```

● ● ● alberto_pluecker_tp1.py
def remover_tarefa(lista_tarefas: list[Tarefa], indice: int):
    """
    Remove uma tarefa específica da lista.

    Args:
        lista_tarefas (list): Lista de tarefas onde a tarefa será removida.
        indice (int): Índice da tarefa a ser removida na lista.

    Returns:
        None
    """
    if 0 < indice <= len(lista_tarefas):
        del lista_tarefas[indice-1]
    else:
        print("Índice inválido!")

```

9. Descrição: Conferir faturas pendentes
 Data de Criação: ('2024-03-03',)
 Prazo Final: 2024-03-09
 Urgência: BAIXA

Digite o índice da tarefa a ser removida: 9