

## Documentação de processos - Lista de Tarefas





Matheus Fernandes de Sousa

linkedin: <https://www.linkedin.com/in/56b49515b/>

link do projeto: [Portfólio Lista De Tarefas](#)

Este projeto foi desenvolvido com o objetivo de consolidar meus conhecimentos iniciais em **Python** e **lógica de programação**, como parte do meu processo de desenvolvimento na área de **Análise de Dados**. A aplicação criada é uma simples **lista de tarefas (to-do list)** em ambiente de terminal, que evolui progressivamente em complexidade e estrutura ao longo do código.

A proposta inicial é trabalhar com conceitos fundamentais como variáveis, entrada de dados via `input()`, estruturas de repetição com `while` e tomada de decisão com `if`, `elif` e `else`. Ao longo do desenvolvimento, funcionalidades importantes são implementadas, como:

-  **Adição e remoção de tarefas**
-  **Definição de prioridades** para cada tarefa, utilizando estruturas de **dicionário**
-  **Organização do código em funções**, promovendo reutilização e legibilidade
-  **Armazenamento persistente de dados** em um arquivo `.json`

A última etapa do projeto garante que os dados da lista sejam salvos de forma permanente, permitindo que as tarefas adicionadas pelo usuário sejam mantidas mesmo após o encerramento do programa. Isso é feito por meio da biblioteca `json`, que transforma a lista de tarefas em um formato legível e armazenável externamente.

Este projeto demonstra como pequenos conceitos podem ser integrados para criar uma aplicação funcional e reutilizável, servindo como um exercício prático e didático para quem está iniciando em programação com Python.

1. **iniciando o projeto, a ideia era criar uma variável(tarefa) para receber uma informação do usuário;**

```
1 # 1
2
3 tarefa = input('Adicione uma tarefa: ') #pede ao usuário para adicionar uma tarefa e armazena a resposta na variável tarefa
4
5 print('A tarefa foi adicionada!')
6 print('Tarefas: ',tarefa)
```

2. **Para melhorar o código, criei uma repetição(while) para que mais tarefas pudessem ser adicionadas. Além disso, criei uma condição (if) caso o usuário**

não tenha mais tarefas para adicionar, ativando a função break ao escrever (sair) para finalizar lista;

```
1 # 2
2
3 while True: #loop para adicionar tarefas
4     opcao = input("Digite uma tarefa (ou 'sair' para finalizar):")
5     if opcao.lower() == 'sair': #quando o usuário digitar 'sair', o looping se encerrará
6         break
7     else:
8         tarefa = opcao
9
10    print("Tarefa adicionada: ", tarefa)
11
12 print("Última tarefa adicionada: ", tarefa)
```

3. Na terceira etapa, adicionei um menu de opções com duas opções, sair ou adicionar tarefas. As tarefas se acumulam através de **append**. Caso o usuário escolha outra opção além das listadas, será informado que a opção é inválida;

```
1 # 3|
2
3 tarefas = []
4
5 while True:
6     print("*** Menu de Opções ***")
7     print("1 - Adicionar Tarefas")
8     print("2 - Sair")
9
10    opcao = input("Escolha uma opção: ")
11    if opcao == '1':
12        tarefa = input("Digite uma tarefa: ")
13        tarefas.append(tarefa) #utilizando append para adicionar e armazenar as tarefas na lista
14        print("Tarefa adicionada: ",tarefa)
15
16    elif opcao == '2': #utilizando elif para criar outra condição, a opção de sair
17        print("Saindo...")
18        break
19
20    else: #caso o usuário digite uma opção que não esteja listada, a função else informa que houve um erro
21        print("opção inválida!")
```

4. Nesta etapa, adicionei a opção de listar tarefas usando condicionais como if, elif e else;

```
1 # 4|
2
3 tarefas = []
4
5 while True:
6     print("*** Menu de Opções ***")
7     print("1 - Adicionar Tarefas")
8     print("2 - Listar Tarefas")
9     print("3 - Sair")
10
11    opcao = input("Escolha uma opção: ")
12    if opcao == '1':
13        tarefa = input("Digite uma tarefa: ")
14        tarefas.append(tarefa)
15        print("Tarefa adicionada: ",tarefa)
16
17    elif opcao == '2':
18        print("Lista de tarefas:")
19        for tarefa in tarefas: #a cada tarefa encontrada na lista tarefas, ele escreverá a informação para o usuário
20            print("- ", tarefa)
21
22    elif opcao == '3':
23        print("Saindo...")
24        break
25
26    else:
27        print("opção inválida!")
```

5. Seguindo com o desenvolvimento da lista de tarefas, adicionei uma nova opção para que o usuário remova as tarefas já finalizadas ou que não são mais úteis. Seguindo o mesmo padrão de condições, adicionando outro elif e utilizando remove para retirar essas tarefas;

```

1 # 5
2
3 tarefas = []
4
5 while True:
6     print("*** Menu de Opções ***")
7     print("1 - Adicionar Tarefas")
8     print("2 - Listar Tarefas")
9     print("3 - Remover Tarefas")
10    print("4 - Sair")
11
12    opcao = input("Escolha uma opção: ")
13    if opcao == '1':
14        tarefa = input("Digite uma tarefa: ")
15        tarefas.append(tarefa)
16        print("Tarefa adicionada: ",tarefa)
17
18    elif opcao == '2':
19        print("Lista de tarefas:")
20        for tarefa in tarefas:
21            print("- ", tarefa)
22

```

```

21        print("- ", tarefa)
22
23    elif opcao == '3':
24        tarefa = input("Digite a tarefa a ser removida: ")
25        if tarefa in tarefas:
26            tarefas.remove(tarefa) #utilizando o remove para o usuário retirar informações que não são mais necessárias
27            print("Tarefa removida:", tarefa)
28        else:
29            print("Tarefa não encontrada!")
30
31    elif opcao == '4':
32        print("Saindo...")
33        break
34
35    else:
36        print("opção inválida!")

```

6. Nesta etapa adicionei mais possibilidades, permitindo que o usuário coloque o nível de prioridade para cada tarefa. Através da criação de dicionários, é possível incluir todas as tarefas em um único grupo, auxiliando na organização do código;

```

1 # 6|
2
3 tarefas = []
4
5 while True:
6     print("*** Menu de Opções ***")
7     print("1 - Adicionar Tarefas")
8     print("2 - Listar Tarefas")
9     print("3 - Remover Tarefas")
10    print("4 - Sair")
11
12    opcao = input("Escolha uma opção: ")
13    if opcao == '1':
14        nome = input("Digite o nome da tarefa: ") #input 1
15        prioridade = input("Digite a prioridade da tarefa: ") #input 2
16        # criando dicionário para guardar informações de nome e prioridade
17        tarefa = {
18            'nome': nome, #guarda a informação do input 1
19            'prioridade': prioridade #guarda a informação do input 2
20        }
21
22        tarefas.append(tarefa) #adiciona cada informação encontrada no dicionário tarefa, na lista denominada tarefas
23        print("Tarefa adicionada: ",tarefa)
24

```

```

23     print("Tarefa adicionada: ", tarefa)
24
25 elif opcao == '2':
26     print("Lista de tarefas:")
27     for tarefa in tarefas:
28         print(f"- {tarefa['nome']} (Prioridade {tarefa['prioridade']})") #lista a tarefa e sua prioridade
29
30 elif opcao == '3':
31     nome = input("Digite o nome da tarefa a ser removida: ")
32     for tarefa in tarefas:
33         if tarefa['nome'] == nome:
34             tarefas.remove(tarefa)
35             print("Tarefa removida:", nome)
36             break
37     else:
38         print("Tarefa não encontrada!")
39
40 elif opcao == '4':
41     print("Saindo...")
42     break
43
44 else:
45     print("opção inválida!")

```

## 7. A seguir, otimizei o código para que fique mais "limpo" e funcional utilizando funções

```

1 # 7 otimizando código com funções
2
3 tarefas = []
4
5
6 def adicionar_tarefa(nome, prioridade): #função para otimizar processos, ao invés de escrever o código por inteiro sempre que for necessário.
7     tarefa = { #A função otimiza o texto, bastando escrever apenas a função para que todo código seja lido
8         'nome': nome,
9         'prioridade': prioridade
10    }
11
12    tarefas.append(tarefa)
13    print("Tarefa adicionada: ", tarefa)
14
15 def listar_tarefas():
16     print("Lista de tarefas:")
17     for tarefa in tarefas:
18         print(f"- {tarefa['nome']} (Prioridade {tarefa['prioridade']})")
19
20 def remover_tarefas(nome):
21     for tarefa in tarefas:
22         if tarefa['nome'] == nome:
23             tarefas.remove(tarefa)
24             print("Tarefa removida:", nome)
25             break
26     else:
27         print("Tarefa não encontrada!")
28
29
30 while True:

```

```

29
30 while True:
31     print("*** Menu de Opções **")
32     print("1 - Adicionar Tarefas")
33     print("2 - Listar Tarefas")
34     print("3 - Remover Tarefas")
35     print("4 - Sair")
36
37     opcao = input("Escolha uma opção: ")
38     if opcao == '1':
39         nome = input("Digite o nome da tarefa: ")
40         prioridade = input("Digite a prioridade da tarefa: ")
41
42         adicionar_tarefa(nome, prioridade) #substituindo o código inteiro pela função, reduzindo a quantidade de informações.
43
44     elif opcao == '2':
45         listar_tarefas()
46
47     elif opcao == '3':
48         nome = input("Digite o nome da tarefa a ser removida: ")
49         remover_tarefas(nome)
50
51     elif opcao == '4':
52         print("Saindo...")
53         break
54
55     else:
56         print("opção inválida!")

```

8. Para finalizar todo o código, exparto a lista para o tipo json. Esta etapa é fundamental para garantir a permanência da aplicação. Ao salvar a lista de tarefas em um arquivo .json, o programa permite que as informações inseridas pelo usuário sejam mantidas mesmo após o encerramento do código.

```
1 # 8 salvando tarefas
2
3 import json #importando a biblioteca json
4
5 tarefas = []
6
7 def salvar_tarefas():
8     with open("tarefas.json", "w") as arquivo: #salvando a lista de tarefas no arquivo json
9         json.dump(tarefas, arquivo, ensure_ascii=False) #json.dump para salvar o arquivo json
10    print("Tarefas salvas com sucesso!") #ensure_ascii=False para permitir que os caracteres especiais sejam salvos corretamente
11
12 def carregar_tarefas(): #função para abrir um arquivo de tarefas json
13    try: #caso não haja nenhum arquivo para carregar, a função cai no except e começa uma lista nova.
14        with open("tarefas.json", "r") as arquivo:
15            tarefas = json.load(arquivo)
16            print("Tarefas carregadas com sucesso!")
17            return tarefas
18    except:
19        print("Nenhum arquivo encontrado! Começando nova lista.")
20    return []
21
```

```
21
22 def adicionar_tarefa(nome, prioridade):
23     tarefa = {
24         'nome': nome,
25         'prioridade': prioridade
26     }
27
28     tarefas.append(tarefa)
29     print("Tarefa adicionada: ", tarefa)
30
31 def listar_tarefas():
32     print("Lista de tarefas:")
33     for tarefa in tarefas:
34         print(f"- {tarefa['nome']} (Prioridade {tarefa['prioridade']})")
35
36 def remover_tarefas(nome):
37     for tarefa in tarefas:
38         if tarefa['nome'] == nome:
39             tarefas.remove(tarefa)
40             print("Tarefa removida:", nome)
41             break
42     else:
43         print("Tarefa não encontrada!")
44
```

```
45 #começo do código, definindo tarefas com a função de carregar tarefas
46 tarefas = carregar_tarefas()
47
48 while True:
49     print("*** Menu de Opções ***")
50     print("1 - Adicionar Tarefas")
51     print("2 - Listar Tarefas")
52     print("3 - Remover Tarefas")
53     print("4 - Salvar Tarefas")
54     print("5 - Sair")
55
56     opcao = input("Escolha uma opção: ")
57     if opcao == '1':
58         nome = input("Digite o nome da tarefa: ")
59         prioridade = input("Digite a prioridade da tarefa: ")
60
61         adicionar_tarefa(nome, prioridade)
62
63     elif opcao == '2':
64         listar_tarefas()
65
66     elif opcao == '3':
67         nome = input("Digite o nome da tarefa a ser removida: ")
68         remover_tarefas(nome)
69
70     elif opcao == '4':
71         salvar_tarefas()
72
73     elif opcao == '5':
74         print("Saindo...")
75         break
76
77     else:
78         print("opção inválida!")
```

### Execução do código:

Este é um demonstrativo de como o código funciona ao ser executado:

- 1 - Adicionarei 4 tarefas
- 2 - Removerei 1 tarefa
- 3 - Salvarei o projeto
- 4 - Mostrarei o projeto aberto no arquivo json

```
*** Nenhum arquivo encontrado! Começando nova lista.
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: 1
Digite o nome da tarefa: Criar projeto de python
Digite a prioridade da tarefa: Alta
Tarefa adicionada: {'nome': 'Criar projeto de python', 'prioridade': 'Alta'}
*** Menu de Opções ***
```

```
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: 1
Digite o nome da tarefa: Fazer prova da faculdade
Digite a prioridade da tarefa: Alta
Tarefa adicionada: {'nome': 'Fazer prova da faculdade', 'prioridade': 'Alta'}
```

Nesta execução abaixo, o código informou que houve uma opção errada por ter digitado o nome da tarefa antes de ter escolhido uma ação

```
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: Fazer atividade física
opção inválida!
```

Listando Tarefas:

```
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: 2
Lista de tarefas:
- Criar projeto de python (Prioridade Alta)
- Fazer prova da faculdade (Prioridade Alta)
- Fazer atividade física (Prioridade Média)
- Lavar tênis (Prioridade Baixa)
** Menu de Opções **
```

## Removendo Tarefas:

```
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: 3
Digite o nome da tarefa a ser removida: Criar projeto de python
Tarefa removida: Criar projeto de python
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
```

Ao digitar uma tarefa que não existe, o usuário é informado que a tarefa não foi encontrada:

```
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: 3
Digite o nome da tarefa a ser removida: a
Tarefa não encontrada!
```

Como a tarefa “criar projeto de python” foi removida, ela não aparece na nova listagem

```
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: 2
Lista de tarefas:
- Fazer prova da faculdade (Prioridade Alta)
- Fazer atividade física (Prioridade Média)
- Lavar tênis (Prioridade Baixa)
```

## Projeto salvo e encerrado

```
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: 4
Tarefas salvas com sucesso!
```

```
** Menu de Opções **
1 - Adicionar Tarefas
2 - Listar Tarefas
3 - Remover Tarefas
4 - Salvar Tarefas
5 - Sair
Escolha uma opção: 5
Saindo...
```

Arquivo json com as tarefas salvas

```
tarefas.json X
1 [{"nome": "Fazer prova da faculdade", "prioridade": "Alta"},
2 {"nome": "Fazer atividade física", "prioridade": "Média"},
3 {"nome": "Lavar tênis", "prioridade": "Baixa"}]
```

Ao desenvolver este projeto, aprendi a estruturar o código de forma lógica e progressiva, pensando em cada etapa com clareza e intenção. Pratiquei o uso de **estruturas de repetição e condicionais** para criar alternativas e fluxos de decisão que tornam a aplicação interativa e funcional. Também compreendi a importância de organizar o código em **funções**, o que trouxe mais **limpeza, reutilização e legibilidade** ao projeto.

Mais do que aplicar conceitos técnicos, este projeto foi essencial para **reforçar minha base em lógica de programação**, algo fundamental para resolver problemas e construir soluções na área de **Análise de Dados**, onde clareza, estrutura e organização do raciocínio são indispensáveis.