

Gerenciamento de Obras

Guilherme Lopes, Sara Stephanie
Salvador, BA, Abril de 2024

Universidade do Estado da Bahia (UNEB)



UNEB
UNIVERSIDADE DO
ESTADO DA BAHIA

Topico

① Conceitos

O que é Gerenciamento de Obras?
Qual a sua importância
Estratégia produtiva

② Proposta do Programa

③ Desenvolvimento

④ Referências

Conceitos: O que é Gerenciamento de Obras?

GERENCIAMENTO DE PROJETOS



Conceitos: Qual a sua importância

QUAIS OS COMPONENTES-CHAVE DE UM PLANO DE GERENCIAMENTO?



Figure: [4]

A importância do gerenciamento de obras pode ser definida por meio dos seus objetivos:

- Controle de estoque de materiais;
- Auxílio na fiel execução dos projetos;
- Gerenciamento de obras antecipa os problemas;
- Estratégia produtiva;
- Redução de custos.

Conceitos: Estratégia produtiva



Figure: [5]

Cada atividade da construção civil apresenta suas próprias especificações. Dessa forma, o gerenciamento contribui para tornar essas atividades mais efetivas, fazendo com que elas atendam a todos os requisitos necessários.

Topico

① Conceitos

② Proposta do Programa

Objetivo do Programa
Gerenciador de Tarefas

③ Desenvolvimento

④ Referências

Proposta do Programa: Objetivo do Programa

```
----- Gerenciador de Tarefas -----  
0 - Adicionar Tarefa  
1 - Remover Tarefa  
2 - Visualizar tarefas  
3 - Sair  
Digite o número do que deseja fazer: █
```

Este projeto teve como objetivo criar um programa em python que fosse capaz gerenciar tarefas básicas de uma obra da construção civil, buscando proporcionar uma maior organização nas tarefas que deverão ser executadas.

Proposta do Programa Objetivo do Programa

```
----- Visualizar Tarefas -----
```

```
1 - Nome Tarefa: Construir ponte
  Membros: Joao, Maria
  Equipamentos: Martelo
  Material: Madeira, Aco, Pregos
2 - Nome Tarefa: Contruir Parede
  Membros: Joao, Maria, Jose
  Equipamentos: Furadeira, Martelo
  Material: Areia, Cimento
(Pressione Enter para voltar ao menu)
```

```
----- Adicionar Tarefa -----
```

```
Digite o nome da tarefa: Construir Parede
Digite o nome do membro(Pressione Enter duas vezes quando terminar): Joao
Digite o nome do membro(Pressione Enter duas vezes quando terminar): Maria
Digite o nome do membro(Pressione Enter duas vezes quando terminar): Jose
Digite o nome do membro(Pressione Enter duas vezes quando terminar):
Digite o nome do equipamento(Pressione Enter duas vezes quando terminar): Furadeira
Digite o nome do equipamento(Pressione Enter duas vezes quando terminar): Martelo
Digite o nome do equipamento(Pressione Enter duas vezes quando terminar):
Digite o nome do material(Pressione Enter duas vezes quando terminar): Areia
Digite o nome do material(Pressione Enter duas vezes quando terminar): Cimento
Digite o nome do material(Pressione Enter duas vezes quando terminar):
Atualizando tabela...
Tarefa adicionada com sucesso!
(Pressione Enter para voltar ao menu)
```

```
----- Remover Tarefa -----
```

```
Indice: 0 Tarefa Construir parede
Indice: 1 Tarefa Construir ponte
Indice: 2 Tarefa Contruir Parede
Digite o indice da tarefa que sera removida(Deixe vazio para cancelar): 0
Atualizando tabela...
Removido com sucesso!
(Pressione Enter para voltar ao menu)
```

Proposta do Programa Gerenciador de Tarefas

Para o gerenciamento das tarefas foi criado a classe TaskManager. Ela é responsável por gerenciar a adição e remoção de tarefas e atualizar os dados em uma tabela excel.

Topico

- ① Conceitos
- ② Proposta do Programa
- ③ Desenvolvimento
Script em Python
- ④ Referências

Desenvolvimento: Script em Python

```
1 import pandas as pd # utilizado para gerenciar os dados e tabelas
2 from os import system # utilizado para limpar o terminal com os metodos system("
  cls") e system("clear")
3 import platform # utilizado para verificar o sistema operacional e deste modo
  utilizar o comando respectivo do sistema para limpar o terminal utilizando
  a funcao system do modulo os
4 import sys # Utilizado para fechar o programa
5
6 # O seguinte programa e um gerenciador de tarefas que pode adicionar, remover e
  visualizar as tarefas, e salva os dados em uma tabela excel
7
```

Desenvolvimento: Script em Python

```
1 class TaskManager:
2     # funcao construtora da classe que verifica se ja existe uma tabela com
3     # dados, se nao cria uma tabela para salvar os dados
4     def __init__(self):
5         try:
6             self.__dfTask = pd.read_excel("tasks_table.xlsx")
7             print("Lendo dados...")
8         except FileNotFoundError:
9             print("Criando tabela...")
10            self.__dfTask = pd.DataFrame({
11                "nomeTarefa": [],
12                "equipe": [],
13                "equipamentos": [],
14                "materiais": [],
15            })
16            self.updateTable()
```

Desenvolvimento: Script em Python

```
1 # Atualiza a tabela(excel) com os dados presentes na tabela __dfTask(  
2   DataFrame do pandas)  
3 def updateTable(self):  
4     self.__dfTask.to_excel("tasks_table.xlsx", index=False)  
5     print("Atualizando tabela...")  
6  
7 # Adiciona uma nova tarefa a tabela  
8 def addTask(self, nomeTarefa: str, membrosEquipe: list, equipamentos: list,  
9   materiais: list):  
10     newLine = {  
11         "nomeTarefa": nomeTarefa,  
12         "equipe": ', '.join(membrosEquipe),  
13         "equipamentos": ', '.join(equipamentos),  
14         "materiais": ', '.join(materiais)  
15     }  
16     self.__dfTask = pd.concat([self.__dfTask, pd.DataFrame([newLine])],  
17   ignore_index=True)  
18     self.updateTable()  
19  
20 # Remove uma tarefa da tabela  
21 def removeTask(self, index):  
22     try:  
23         self.__dfTask.drop(index, inplace=True)  
24         self.__dfTask.reset_index(drop=True, inplace=True)  
25         self.updateTable()  
26         print('Removido com sucesso!')  
27     except KeyError:  
28         print('Index nao encontrado!')
```

Desenvolvimento: Script em Python

```
1 # Retorna a tabela __dfTask
2 def getTableTasks(self):
3     return self.__dfTask
4
```

Desenvolvimento: Script em Python

```
1 # Instacia de um objeto TaskManager
2 taskManager = TaskManager()
3
4 # Metodo que imprime o menu principal da aplicacao e permite escolher outras
   acoes no sistema
5 def main():
6     cleanTerminal()
7
8     print('-'*10, 'Gerenciador de Tarefas', '-'*10)
9     print('0 - Adicionar Tarefa')
10    print("1 - Remover Tarefa")
11    print('2 - Visualizar tarefas')
12    print('3 - Sair')
13
14    numMenu = int(input('Digite o numero do que deseja fazer: '))
15
16    if numMenu == 0:
17        optionAdd()
18    elif numMenu == 1:
19        optionRemove()
20    elif numMenu == 2:
21        optionView()
22    elif numMenu == 3:
23        sys.exit()
24
```


Desenvolvimento: Script em Python

```
1 # Imprime no terminal a area de adicionar novas tarefas ao objeto taskManager
2 def optionAdd():
3     cleanTerminal()
4     print('-'*10, 'Adicionar Tarefa', '-'*10)
5
6     nomeTarefa = input('Digite o nome da tarefa: ')
7     membrosEquipe = []
8     equipamentos = []
9     materiais = []
10
11 while True:
12     nomeMembro = input("Digite o nome do membro(Pressione Enter duas vezes
13     quando terminar): ").strip()
14
15     if nomeMembro != "":
16         membrosEquipe.append(nomeMembro)
17     else:
18         break
19
20 while True:
21     equipamento = input("Digite o nome do equipamento(Pressione Enter duas
22     vezes quando terminar): ").strip()
23
24     if equipamento != "":
25         equipamentos.append(equipamento)
26     else:
27         break
```

Desenvolvimento: Script em Python

```
1 while True:
2     material = input("Digite o nome do material(Pressione Enter duas
3     vezes quando terminar): ").strip()
4
5     if material != "":
6         materiais.append(material)
7     else:
8         break
9
10 taskManager.addTask(nomeTarefa, membrosEquipe, equipamentos, materiais)
11
12 print('Tarefa adicionada com sucesso!')
13 input('(Pressione Enter para voltar ao menu)')
```

Desenvolvimento: Script em Python

```
1 # Imprime a area para remover tarefas cadastradas no sistema
2 def optionRemove():
3     cleanTerminal()
4     print('-'*10, 'Remover Tarefa', '-'*10)
5     if len(taskManager.getTableTasks()) != 0:
6         i = 0
7         while i < len(taskManager.getTableTasks()):
8             print('Indice:', i, ' ', 'Tarefa', taskManager.getTableTasks().loc[i,
9                 'nomeTarefa'])
10            i += 1
11            indexLine = input("Digite o indice da tarefa que sera removida(Deixe
12                vazio para cancelar): ").strip()
13
14            if indexLine != "":
15                taskManager.removeTask(int(indexLine))
16            else:
17                print('Remocao cancelada!')
18
19 else:
20     print("Sem tarefas cadastradas!")
21     input('(Pressione Enter para voltar ao menu)')
```

Desenvolvimento: Script em Python

```
1 #Imprime as tarefas cadastradas no sistema
2 def optionView():
3     cleanTerminal()
4
5     print('-'*10, 'Visualizar Tarefas', '-'*10)
6
7     tb = taskManager.getTableTasks()
8     i = 0
9
10    if len(tb) > 0:
11        while i < len(tb):
12            print((i + 1), '- Nome Tarefa:', tb.loc[i, "nomeTarefa"])
13            print('    Membros:', tb.loc[i, 'equipe'])
14            print('    Equipamentos:', tb.loc[i, 'equipamentos'])
15            print('    Material:', tb.loc[i, 'materiais'])
16            i += 1
17    else:
18        print("Nenhuma tarefa cadastrada!")
19    input('(Pressione Enter para voltar ao menu)')
```

Desenvolvimento: Script em Python

```
1 # Limpa o terminal para manter as informacoes organizadas
2 def cleanTerminal():
3     operatingSystem = platform.system()
4     if operatingSystem == 'Windows':
5         system('cls')
6     else:
7         system('clear')
8
9 # Loop que repetir de forma infinita chamando o metodo main(). Este loop so
10 # acabara se o metodo sys.exit() for chamado dentro da logica da funcao main
11 # ()
12 while True:
13     main()
```

Topico

- 1 Conceitos
- 2 Proposta do Programa
- 3 Desenvolvimento
- 4 Referências

Referências I

- [1] Gerenciamento de obras: o que é e por que é tão importante - blog da orcafascio. <https://www.orcafascio.com/papodeengenheiro/gerenciamento-de-obras>, 2024.
Accessed: 2024-04-14.
- [2] Resultado da pesquisa de imagens do google.
<https://images.app.goo.gl/qRfPTRec7aSJYvuv8>, 2024.
Accessed: 2024-04-14.
- [3] Resultado da pesquisa de imagens do google.
<https://images.app.goo.gl/YQ3MLbiVTDPVM4VGA>, 2024.
Accessed: 2024-04-14.
- [4] Resultado da pesquisa de imagens do google.
<https://images.app.goo.gl/WbeFAPwh94jRRZ5P9>, 2024.
Accessed: 2024-04-14.

Referências II

- [5] Resultado da pesquisa de imagens do google.
<https://images.app.goo.gl/7WPavKYywBLB3n8e9>, 2024.
Accessed: 2024-04-14.