

GESTÃO DE RESÍDUOS NA CONSTRUÇÃO CIVIL

Prof Drº Robson Marinho da Silva
Salvador, BA, JUNHO de 2024

Universidade do Estado da Bahia (UNEB)



Agenda

- ① INTRODUÇÃO
CONCEITO
- ② MOTIVAÇÕES E JUSTIFICATIVA
- ③ PROGRAMA-
- ④ MATERIAL
- ⑤ INTERFACE

A gestão de resíduos na construção civil envolve práticas essenciais para manejar adequadamente os resíduos gerados durante a construção, reforma e demolição de edificações e infraestruturas. Esses resíduos, frequentemente compostos por concreto, madeira, metais, plásticos, gesso e outros materiais, precisam ser tratados de maneira a minimizar seu impacto ambiental.

- 1. Identificação e Classificação:** Reconhecimento e categorização dos resíduos gerados.
- 2. Minimização:** Redução da geração de resíduos através do uso eficiente de materiais e práticas sustentáveis.
- 3. Reutilização:** Aproveitamento de materiais reutilizáveis, como madeira e tijolos.
- 4. Reciclagem:** Envio de materiais recicláveis para empresas especializadas.
- 5. Disposição Final:** Descarte adequado em aterros sanitários ou locais licenciados.
- 6. Monitoramento e Controle:** Avaliação contínua e melhorias nas práticas de gestão.

Agenda

- ① INTRODUÇÃO
- ② MOTIVAÇÕES E JUSTIFICATIVA
- ③ PROGRAMA-
- ④ MATERIAL
- ⑤ INTERFACE

MOTIVAÇÕES E JUSTIFICATIVA

MOTIVAÇÕES

- Diminuir a quantidade de resíduos enviados a aterros sanitários, reduzindo a poluição do solo, ar e água.
- Diminuir os custos de disposição de resíduos e compra de novos materiais através da reutilização e reciclagem.
- Atender às exigências legais e regulatórias impostas por órgãos ambientais locais, estaduais e federais.
- Melhorar a eficiência do canteiro de obras, reduzindo desperdícios e aumentando a produtividade.
- Melhorar a eficiência do canteiro de obras, reduzindo desperdícios e aumentando a produtividade.

MOTIVAÇÕES E JUSTIFICATIVA

JUSTIFICATIVA

- A gestão adequada dos resíduos minimiza o impacto negativo no meio ambiente, preservando ecossistemas e biodiversidade.
- Transformar resíduos em recursos valiosos, gerando novas oportunidades de negócios.
- Minimizar os riscos à saúde associados ao manejo inadequado de resíduos perigosos.
- Alinhar-se às políticas e metas públicas de sustentabilidade e gestão de resíduos, contribuindo para o desenvolvimento sustentável das cidades.

Agenda

- ① INTRODUÇÃO
- ② MOTIVAÇÕES E JUSTIFICATIVA
- ③ PROGRAMA-
OBJETIVO
CÓDIGO
- ④ MATERIAL
- ⑤ INTERFACE

O objetivo do programa é fornecer uma ferramenta para a gestão eficiente de resíduos na construção civil. Através de uma interface gráfica construída com PySimpleGUI, o programa permite aos usuários cadastrar materiais, adicionar e verificar resíduos gerados, obter relatórios detalhados e receber recomendações sobre o destino apropriado dos resíduos. O programa também calcula a economia financeira decorrente do gerenciamento adequado dos resíduos. O programa visa promover a sustentabilidade e a eficiência na construção civil, facilitando a gestão de resíduos e promovendo práticas de reciclagem e reutilização. Ao calcular a economia financeira e fornecer recomendações sobre a destinação dos resíduos, a ferramenta incentiva a adoção de práticas mais sustentáveis e economicamente vantajosas.

```
1  # Import the necessary modules.
2  from gestao_residuos import GestaoResiduos
3  import interface
4  import PySimpleGUI as sg
5  from material import Material
6
7  def app():
8      window_start = interface.start_window()
9      window_cadastrar = None
10     window_verificar_residuo = None
11     gestao = GestaoResiduos()
12     material = Material()
13
14     while True:
15         window, event, values = sg.read_all_windows()
16
17         if window == window_start and (event == sg.WIN_CLOSED or event == "Sair"):
18             break
19
20         if window == window_start and event == "Cadastrar":
21             window_start.hide()
22             window_cadastrar = interface.cadastrar_window()
23
24         if window == window_cadastrar and (event == sg.WIN_CLOSED or event == "Voltar"):
25             window_start.un_hide()
26             window_cadastrar.close()
27             window_cadastrar = None
28
29
```

PROGRAMA- CÓDIGO

```
30 if window == window_start and event == "Verificar Residuo":
31     window_start.hide()
32     window_verificar_residuo = interface.verificar_residuo_window()
33
34 if window == window_verificar_residuo and (event == sg.WIN_CLOSED or event == "Voltar"):
35     window_start.un_hide()
36     window_verificar_residuo.close()
37     window_verificar_residuo = None
38
39 if window == window_verificar_residuo and event == "Verificar Residuo":
40     tipo_residuo = window["tipo_residuo"].get()
41     quantidade_residuo = float(window["quantidade_residuo"].get())
42     gestao.adicionar_residuo(tipo_residuo, quantidade_residuo)
43     sg.popup(f'{gestao.relatorio_residuos()}\n{gestao.recomendacoes()}\n{gestao.calcular_economia()}')
44
45 if window == window_start and event == "Listar":
46     sg.popup(material.listaTabelaMaterial())
47
```

```
7  def app():
47
48      if window == window_cadastrar and event == "SALVAR":
49          nome_material = window["material"].get()
50          quantidade_material = window["quantidade"].get()
51          organica = window["is_organica"].get()
52
53          material.adicionarMaterial(nome_material, quantidade_material, organica)
54          sg.popup("Material Adicionado!")
55
56          window["material"].update("")
57          window["quantidade"].update("")
58          window["is_organica"].update(False)
59
60
61      window_start.close()
62
63  # Run the main function.
64  if __name__ == "__main__":
65      app()
66
```

```
1 # Define the Gesta Resíduos class.
2 class GestaoResiduos:
3     def __init__(self):
4         self.residuos = {}
5         self.economia = 0
6
7     def adicionar_residuo(self, tipo, quantidade):
8         if tipo in self.residuos:
9             self.residuos[tipo] += quantidade
10        else:
11            self.residuos[tipo] = quantidade
12
13    def relatorio_residuos(self):
14        texto = ""
15        for tipo, quantidade in self.residuos.items():
16            texto = texto + (f"{tipo}: {quantidade}kg\n")
17
18        return texto
19
```

```
19
20 def recomendacoes(self):
21     texto = ""
22     for tipo in self.residuos:
23         if tipo == "concreto":
24             texto = texto + ("- Concreto: Pode ser triturado e reutilizado como base para pavimentação.\n")
25         elif tipo == "madeira":
26             texto = texto + ("- Madeira: Pode ser reciclada ou utilizada para geração de energia.\n")
27         elif tipo == "metal":
28             texto = texto + ("- Metal: Deve ser enviado para reciclagem.\n")
29         elif tipo == "vidro":
30             texto = texto + ("- Vidro: Pode ser reciclado e reutilizado em produtos de vidro.\n")
31         elif tipo == "plastico":
32             texto = texto + ("- Plástico: Deve ser reciclado e reutilizado em produtos de plástico.\n")
33         elif tipo == "papel":
34             texto = texto + ("- Papel: Pode ser reciclado e reutilizado em produtos de papel.\n")
35         elif tipo == "organico":
36             texto = texto + ("- Orgânico: Pode ser compostado e utilizado como adubo.\n")
37         else:
38             texto = texto + ("#- {tipo}: Verificar regulamentações locais para destinação adequada.\n")
39     return texto
```

```
40 def calcular_economia(self):
41     self.economia = 0
42     for tipo, quantidade in self.residuos.items():
43         if tipo == "concreto":
44             self.economia += quantidade * 0.5 # economia de 50% ao reutilizar concreto
45         elif tipo == "madeira":
46             self.economia += quantidade * 0.3 # economia de 30% ao reciclar madeira
47         elif tipo == "metal":
48             self.economia += quantidade * 0.7 # economia de 70% ao reciclar metal
49         elif tipo == "vidro":
50             self.economia += quantidade * 0.4 # economia de 40% ao reciclar vidro
51         elif tipo == "plastico":
52             self.economia += quantidade * 0.2 # economia de 20% ao reciclar plástico
53         elif tipo == "papel":
54             self.economia += quantidade * 0.1 # economia de 10% ao reciclar papel
55         elif tipo == "orgânico":
56             self.economia += quantidade * 0.6 # economia de 60% ao compostar orgânico
57     return (f"\nEconomia total: R$ {self.economia:.2f}")
58
```

Agenda

- ① INTRODUÇÃO
- ② MOTIVAÇÕES E JUSTIFICATIVA
- ③ PROGRAMA-
- ④ MATERIAL**
- ⑤ INTERFACE


```
1 import pandas as pd
2
3 class Material:
4
5     def __init__(self):
6         try:
7             self.__tabelaMaterial = pd.read_excel("tabela_material.xlsx")
8         except FileNotFoundError:
9             self.__tabelaMaterial = pd.DataFrame({
10                 "material": ["concreto", "vidro", "plastico", "papel", "organico", "metal", "madeira"],
11                 "quantidade": ["10", "10", "10", "10", "10", "10", "10"],
12                 "organica": [False, False, False, False, True, False, True]
13             })
14             self.__atualizarTabela()
15
16     def __atualizarTabela(self):
17         self.__tabelaMaterial.to_excel("tabela_material.xlsx", index=False)
18
```

```
18
19 def adicionarMaterial(self, material, quantidade, organica):
20     novoMaterial = {
21         "material": material,
22         "quantidade": quantidade,
23         "organica": organica
24     }
25     self.__tabelaMaterial = pd.concat([self.__tabelaMaterial, pd.DataFrame([novoMaterial])], ignore_index=True)
26     self.__atualizarTabela()
27
28 def listaTabelaMaterial(self):
29     texto = ""
30     i = 0
31
32     if len(self.__tabelaMaterial) > 0:
33         while i < len(self.__tabelaMaterial):
34             texto += "-----\n"
35             texto += ("Material: {}".format(self.__tabelaMaterial.loc[i, "material"]))
36             texto += (f"Quantidade: {self.__tabelaMaterial.loc[i, 'quantidade']}\n")
37             texto += (f"Organica: {self.__tabelaMaterial.loc[i, 'organica']}\n")
38             texto += "-----\n"
39             i += 1
40     return texto
41
```

Agenda

- ① INTRODUÇÃO
- ② MOTIVAÇÕES E JUSTIFICATIVA
- ③ PROGRAMA-
- ④ MATERIAL
- ⑤ INTERFACE

Facilitar a Gestão de Resíduos: A interface visa tornar o processo de gerenciamento de resíduos simples e eficiente, permitindo que os usuários verifiquem, listem e cadastrem materiais rapidamente.

Intuitiva e Acessível: Usando PySimpleGUI, a interface é desenhada para ser intuitiva, com uma aparência limpa e botões claramente rotulados para realizar as ações principais.

Automatização e Persistência de Dados: Ao integrar com uma classe que gerencia a tabela de materiais, a interface não só permite a interação com os dados, mas também assegura que as alterações sejam persistentes através do armazenamento em arquivos Excel.

```
1  import PySimpleGUI as sg
2
3  def start_window():
4      sg.theme("Reddit")
5
6      layout = [
7          [sg.Button("Verificar Residuo")],
8          [sg.Button("Listar")],
9          [sg.Button("Cadastrar")],
10         [sg.Button("Sair")]
11     ]
12
13     return sg.Window("Gestão de Resíduos", layout=layout, finalize=True)
```

```
14
15 def verificar_residuo_window():
16     sg.theme("Reddit")
17
18     layout = [
19         [sg.Text("Tipo de residuo:"), sg.Input(key="tipo_residuo")],
20         [sg.Text("Quantidade:"), sg.Input(key="quantidade_residuo")],
21         [sg.Button("Verificar Residuo")],
22         [sg.Button("Voltar")]
23     ]
24
25     return sg.Window("Verifar Residuo", layout=layout, finalize=True)
26
27 def cadastrar_window():
28     sg.theme("Reddit")
29
30     layout = [
31         [sg.Text("Material"), sg.Input(key="material")],
32         [sg.Text("Quantidade"), sg.Input(key="quantidade")],
33         [sg.Text("Organica"), sg.Check("", key="is_organica")],
34         [sg.Button("SALVAR")],
35         [sg.Button("Voltar")]
36     ]
37
38     return sg.Window("Cadastrar Material", layout=layout, finalize=True)
39
```

Guilherme Lopes Uneb

Cadastrar Material

Material

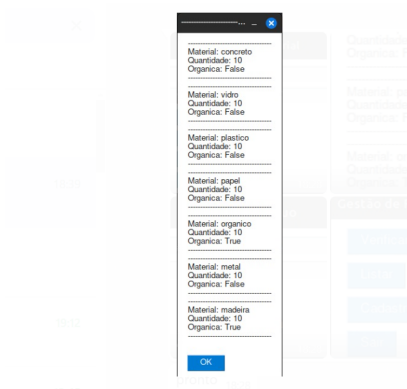
Quantidade

Organica ☐

SALVAR

Voltar

INTRODUÇÃO MOTIVAÇÕES E JUSTIFICATIVA PROGRAMA- MATERIAL INTERFACE



Verificar Residuo [Close]

Tipo de residuo:

Quantidade:

Verificar Residuo

Voltar

