



TECNOLOGIA DA INFORMAÇÃO NO CANTEIRO DE OBRAS/CALCULO DE CUSTO DE MATERIAIS

Caio Henrique Pinheiro Oliveira, Igor Pinto dos Santos

UNEB

INTRODUÇÃO

Tecnologia da informação na construção civil também se beneficia do uso da linguagem de programação Python. Python é amplamente utilizado para automatizar tarefas repetitivas, processar dados e criar ferramentas personalizadas para análise e modelagem. Sua flexibilidade e vasta biblioteca de recursos tornam Python uma escolha popular para desenvolver soluções de software que podem melhorar a eficiência no canteiro de obras. Além disso, a aplicação de aprendizado de máquina e análise de dados em Python permite prever problemas antes que ocorram, otimizando ainda mais os processos de construção. Assim, Python desempenha um papel importante na integração da tecnologia da informação na indústria da construção, contribuindo para a automação e aprimoramento contínuo do setor.

Objetivo

O objetivo principal deste trabalho é investigar e demonstrar como a programação em Python pode ser aplicada para atingir os seguintes objetivos na construção civil:

Automação de Tarefas Repetitivas: Utilizar scripts em Python para automatizar tarefas rotineiras, como o processamento de documentos, relatórios e medições. Isso economiza tempo e reduz erros humanos.

Gerenciamento de Projetos: Desenvolver aplicativos em Python para gerenciar projetos de construção, incluindo o acompanhamento do progresso, alocação de recursos e agendamento de tarefas.

Sustentabilidade e Eficiência Energética: Desenvolver soluções em Python para otimizar o consumo de energia e recursos naturais, tornando os edifícios mais sustentáveis.

Contudo, partindo desse ponto pode-se pensar no Python como uma ferramenta facilitadora desse processo, pois Python é uma linguagem de programação de nível alto, que usa instruções humanas e mais abstratas, incrementada através de um conjunto de instruções que permitem que uma máquina execute tarefas, como a calculadora de custo de materiais no canteiro de obras

OUTRAS FERRAMENTAS

Além do Python, várias ferramentas desempenharam um papel fundamental no desenvolvimento deste projeto:

Visual Studio Code (VSCode) Git e GitHub Trello Overleaf

Bibliotecas Python (TKINTER)

Conduzimos uma pesquisa ampla envolvendo análise de diversos tipos de fontes, desde artigos científicos até entrevistas com profissionais do setor de construção e tecnologia. Identificamos tendências e exemplos notáveis de integração da TI em canteiros de obras e discutimos os desafios e estratégias para maximizar os benefícios dessa evolução tecnológica na construção civil.

Proposta do programa

A proposta deste programa é fornecer aos profissionais da construção civil, engenheiros, arquitetos, empreiteiros e outros envolvidos na área uma ferramenta simples e eficaz para calcular o custo total de materiais de construção. A aplicação visa melhorar a eficiência, economizar tempo e reduzir erros comuns na estimativa de custos.

Script(interface gráfica do código anterior)

Nome do material:

Preço unitário:

Quantidade:

O custo total do material Cimento é de R\$ 8162.50


```
CODIGO ATUALIZADO.py > calcular_custo
1  # Importa as bibliotecas tkinter e ttk, que são usadas para criar a interface gráfica
2  import tkinter as tk
3  from tkinter import ttk
4
5  # Define a classe MaterialConstrucao
6  class MaterialConstrucao:
7      def __init__(self, nome, preco_unitario, quantidade):
8          # Inicializa um objeto MaterialConstrucao com nome, preço unitário e quantidade
9          self.nome = nome
10         self.preco_unitario = preco_unitario
11         self.quantidade = quantidade
12
13     def calcular_custo_total(self):
14         # Calcula o custo total multiplicando o preço unitário pela quantidade
15         return self.preco_unitario * self.quantidade
16
17     # Dicionário para armazenar informações sobre os materiais
18     materiais = {
19         "cimento": {"preco_unitario": 115.0},
20         "tijolos": {"preco_unitario": 2.5},
21         "madeira": {"preco_unitario": 18.5},
22         "telha ceramica": {"preco_unitario": 1.99},
23         "gesso": {"preco_unitario": 38.5}
24     }
```

```
22     'telha ceramica': {preco_unitario: 1.99},
23     "gesso": {"preco_unitario": 38.5}
24 }
25
26 historico_calculos = [] # Lista para armazenar o histórico de cálculos
27
28 # Função para calcular o custo
29 def calcular_custo():
30     nome_material = nome_entry.get() # Obtém o nome do material da entrada de texto
31     preco_unitario_str = preco_unitario_entry.get() # Obtém o preço unitário da entrada de texto
32     quantidade_str = quantidade_entry.get() # Obtém a quantidade da entrada de texto
33
34     if not nome_material or not preco_unitario_str or not quantidade_str:
35         # Verifica se algum dos campos está vazio e exibe uma mensagem de erro
36         resultado_label.config(
37             text="Por favor, preencha todos os campos.", foreground="red")
38         return
39
40     if nome_material not in materiais:
41         # Verifica se o material não está no dicionário de materiais e exibe uma mensagem de erro
42         resultado_label.config(text="Material desconhecido.", foreground="red")
43         return
44
45     try:
46         float(preco_unitario_str) # Tenta converter o preço unitário em um número de ponto flutuante
47     except :
48         # Se a conversão falhar, exibe uma mensagem de erro
49         resultado_label.config(
50             text="Preço unitário inválido. Insira um número válido.", foreground="red")
51         return
52
```

```
51     return
52
53     try:
54         quantidade = int(quantidade_str) # Tenta converter a quantidade em um número inteiro
55     except :
56         # Se a conversão falhar, exibe uma mensagem de erro
57         resultado_label.config(
58             text="Quantidade inválida. Insira um número inteiro válido.", foreground="red")
59     return
60
61     material = MaterialConstrucao(
62         nome_material, materiais[nome_material]["preco_unitario"], quantidade)
63     custo_total = material.calcular_custo_total() # Calcula o custo total do material
64
65     nome_material = nome_material.upper() # Converte o nome do material para maiúsculas
66     resultado_label.config(text=f"O custo total do material {nome_material} é de R$ {custo_total:.2f}", foregro
67
68     # Limpa as entradas de texto
69     nome_entry.delete(0, tk.END)
70     preco_unitario_entry.delete(0, tk.END)
71     quantidade_entry.delete(0, tk.END)
72
73     historico_calculos.append([
74         f"Material: {nome_material}, Custo: R$ {custo_total:.2f}"])
75
76     historico_text.config(state=tk.NORMAL)
77     historico_text.delete("1.0", tk.END) # Limpa o histórico anterior
78
79     for calculo in historico_calculos:
80         # Adiciona os cálculos anteriores ao histórico
81         historico_text.insert(tk.END, calculo + "\n")
82
```

```
80         # Adiciona os cálculos anteriores ao histórico
81         historico_text.insert(tk.END, calculo + "\n")
82
83     historico_text.config(state=tk.DISABLED)
84
85     # Configuração da interface gráfica
86     root = tk.Tk() # Cria a janela principal
87     root.title("Calculadora de Custo de Materiais de Construção") # Define o título da janela
88
89     frame1 = ttk.Frame(root) # Cria um quadro dentro da janela principal
90     frame1.pack(padx=10, pady=10) # Define o preenchimento do quadro
91
92     # Rótulos e entradas de texto para nome, preço unitário, quantidade e resultado
93     ttk.Label(frame1, text="Nome do material:").grid(row=0, column=0, sticky=tk.W)
94     ttk.Label(frame1, text="Preço unitário (R$):").grid(
95         row=1, column=0, sticky=tk.W)
96     ttk.Label(frame1, text="Quantidade:").grid(row=2, column=0, sticky=tk.W)
97     ttk.Label(frame1, text="Resultado:").grid(row=4, column=0, sticky=tk.W)
98
99     nome_entry = ttk.Entry(frame1) # Entrada de texto para o nome do material
100     preco_unitario_entry = ttk.Entry(frame1) # Entrada de texto para o preço unitário
101     quantidade_entry = ttk.Entry(frame1) # Entrada de texto para a quantidade
102
103     # Posições das entradas de texto
104     nome_entry.grid(row=0, column=1, sticky=tk.W)
105     preco_unitario_entry.grid(row=1, column=1, sticky=tk.W)
106     quantidade_entry.grid(row=2, column=1, sticky=tk.W)
107
108     # Botão para calcular o custo
109     calcular_button = ttk.Button(
110         frame1, text="Calcular Custo", command=calcular_custo)
111     calcular_button.grid(row=3, column=1)
```

```
111 calcular_button.grid(row=3, column=1)
112
113 resultado_label = ttk.Label(frame1, text="") # Rótulo para exibir o resultado
114 resultado_label.grid(row=4, column=1, sticky=tk.W) # Posição do rótulo de resultado
115
116 historico_label = ttk.Label(frame1, text="Histórico de Cálculos:") # Rótulo para o histórico
117 historico_label.grid(row=5, column=0, columnspan=2, sticky=tk.W) # Posição do rótulo de histórico
118
119 v historico_text = tk.Text(frame1, state=tk.DISABLED,
120 | | | | | | | | wrap=tk.WORD, height=5, width=40) # Área de texto para o histórico
121 historico_text.grid(row=6, column=0, columnspan=2) # Posição da área de texto
122
123 root.mainloop() # Inicia o loop principal da interface gráfica
```

Script (interface gráfica do código atualizado)

Nome do material: tijolos

Preço unitário (R\$): 2.5

Quantidade: 22

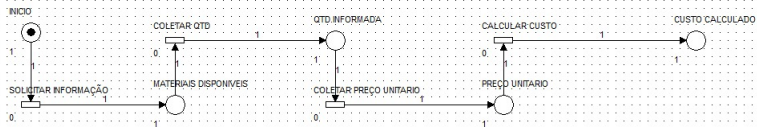
Calcular Custo

Resultado: O custo total do material TIJOLOS é de R\$ 55.00

Histórico de Cálculos:

Material: TIJOLOS, Custo: R\$ 62.50

Material: TIJOLOS, Custo: R\$ 55.00



Este projeto exemplifica a aplicação prática da tecnologia da informação na construção civil por meio de uma "Calculadora de Custo de Materiais de Construção" em Python. Destaca a importância da pesquisa para identificar tendências e desafios nessa integração da TI na indústria. A tecnologia demonstrada simplifica o cálculo de custos, economizando tempo e reduzindo erros. Em resumo, ressalta a relevância da TI na modernização e melhoria da eficiência na construção civil.

FEREGUETTI, Larissa. O que é a language Pmython e como ela está presente na engenharia?. [S. l.], 12 maio 2019. Disponível em: <https://engenharia360.com/linguagem-python-na-engenharia/>.

Acesso em: 23 mar. 2023.

COUTINHO, Thiago. Como o Python pode ajudar um engenheiro e quais as suas aplicações na engenharia?. [S. l.], 18 mar. 2021.

Disponível em:

<https://www.voitto.com.br/blog/artigo/python-para-engenheiros>.

Acesso em: 23 mar. 2023

RESUMO GERAL:

A objetivo deste programa e fornecer aos profissionais da construção civil, engenheiros, arquitetos, empreiteiros e outros envolvidos na area uma ferramenta simples e eficaz para calcular o custo total de materiais de construção. A aplicação visa melhorar a eficiência, economizar tempo e reduzir erros comuns na estimativa de custos.

- FUNCIONALIDADES PRINCIPAIS:
- Interface Gráfica Intuitiva
- Cálculo dinâmico
- Validações Integradas
- Histórico de cálculos