Curso de Tecnologia em Sistemas Para Internet.
Disciplina: Programação Desktop
Prof.: Abraão Gualberto Nazário

Código de entrega trabalho: Integração com a API “viacep”

Código Para Simples Verificação:

```
import tkinter as tk
from tkinter import messagebox
import requests

def get_address():
    cep = entry_cep.get()
    if len(cep) != 8:
        messagebox.showerror("Erro", "CEP deve ter 8 dígitos.")
        return

    url = f"https://viacep.com.br/ws/{cep}/json/"
    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()
        if "erro" in data:
            messagebox.showerror("Erro", "CEP não encontrado.")
        else:
            address = f"{data['logradouro']}, {data['bairro']}, {data['localidade']} - {data['uf']}"
            label_address.config(text=address)
    else:
        messagebox.showerror("Erro", "Falha ao obter o endereço.")

# Cria a janela principal
root = tk.Tk()
root.title("Consulta de CEP")
root.configure(bg="green")
root.geometry("600x300")
root.resizable(True, True)
```

```
    # Cria os widgets
    label_cep = tk.Label(root, text="Digite o CEP:", bg="black", fg="yellow")
    entry_cep = tk.Entry(root, bg="black", fg="yellow", insertbackground="yellow")
    button_consultar = tk.Button(root, text="Consultar", command=get_address, bg="black", fg="yellow")
    label_localizacao = tk.Label(root, text="Localização:", bg="black", fg="yellow")
    label_address = tk.Label(root, text="", wraplength=300, justify="left", bg="black", fg="yellow")

    # Posiciona os widgets na janela
    label_cep.pack(pady=5)
    entry_cep.pack(pady=5)
    button_consultar.pack(pady=5)
    label_localizacao.pack(pady=5)
    label_address.pack(pady=5)

    # Inicia o loop principal da aplicação
    root.mainloop()
```

Explicação o código da integração com API "ViaCEP"

1 - Importações de Bibliotecas:

```
import tkinter as tk
from tkinter import messagebox
import requests
```

tkinter as tk: Importa a biblioteca Tkinter para criar a interface gráfica.

messagebox: Importa a caixa de mensagem do Tkinter para exibir mensagens de erro.

requests: Importa a biblioteca Requests para fazer requisições HTTP.

2 - Definição da Função get_address:

```
def get_address():
    cep = entry_cep.get()
    if len(cep) != 8:
        messagebox.showerror("Erro", "CEP deve ter 8 dígitos.")
        return
```

3 - Define a função get_address que será chamada quando o botão "Consultar" for clicado.

Obtém o valor do campo de entrada entry_cep.

Verifica se o CEP tem 8 dígitos. Se não tiver, exibe uma mensagem de erro e retorna.

4 - Requisição à API do ViaCEP:

```
url = f"https://viacep.com.br/ws/{cep}/json/"
    response = requests.get(url)
```

5 - Constrói a URL para a requisição à API do ViaCEP.

Faz a requisição GET à API.

Tratamento da Resposta da API:

6 - if response.status_code == 200:

```
data = response.json()
if "erro" in data:
   messagebox.showerror("Erro", "CEP não encontrado.")
else:
   address = f"{data['logradouro']}, {data['bairro']}, {data['localidade']} - {data['uf']}"
   label_address.config(text=address)
else:
   messagebox.showerror("Erro", "Falha ao obter o endereço.")
```

7 -Verifica se a resposta da API tem status 200 (OK).

Converte a resposta para JSON.

Verifica se a resposta contém um erro. Se sim, exibe uma mensagem de erro.

Se não houver erro, constrói a string do endereço e atualiza o texto do rótulo label_address.

Se a resposta não for 200, exibe uma mensagem de erro.

8 - Criação da Janela Principal:

```
root = tk.Tk()
root.title("Consulta de CEP")
root.configure(bg="black")
root.geometry("600x300")
root.resizable(True, True)
```

Cria a janela principal.

Define o título da janela como "Consulta de CEP".

Configura o fundo da janela como preto.

Define as dimensões da janela como 600x300 pixels.

Permite que a janela seja redimensionável.

9 - Criação dos Widgets:

label_cep = tk.Label(root, text="Digite o CEP:", bg="black", fg="yellow")
entry_cep = tk.Entry(root, bg="black", fg="yellow", insertbackground="yellow")
button_consultar = tk.Button(root, text="Consultar", command=get_address, bg="black", fg="yellow")
label_localizacao = tk.Label(root, text="Localização:", bg="black", fg="yellow")
label_address = tk.Label(root, text="", wraplength=300, justify="left", bg="black", fg="yellow")
Cria um rótulo label_cep com o texto "Digite o CEP:".
Cria um campo de entrada entry_cep para o usuário digitar o CEP.
Cria um botão button_consultar com o texto "Consultar" que chama a função get_address quando clicado.
Cria um rótulo label_localizacao com o texto "Localização:".
Cria um rótulo label_address para exibir o endereço obtido da API.

10 - Posicionamento dos Widgets na Janela:
label_cep.pack(pady=5)
entry_cep.pack(pady=5)
button_consultar.pack(pady=5)
label_localizacao.pack(pady=5)
label_address.pack(pady=5)
Posiciona os widgets na janela usando o método pack com um espaçamento vertical (pady) de 5 pixels entre eles.
Início do Loop Principal da Aplicação:

11 - root.mainloop()
Inicia o loop principal da aplicação Tkinter, que mantém a janela aberta e responde a eventos do usuário.