

# Introdução

Este sistema foi desenvolvido utilizando a biblioteca Tkinter para criação de interfaces gráficas em Python. O objetivo do programa é permitir a adição, remoção, listagem e salvamento de voluntários em um banco de dados SQLite. A interface foi projetada para ser simples e de fácil utilização, com componentes centralizados e uma aparência organizada.

## Funcionalidades

O sistema implementa as seguintes funcionalidades:

1. **Adicionar Voluntário:** Insere o nome e o e-mail de um voluntário no banco de dados.
2. **Remover Voluntário:** Remove o voluntário com base no ID fornecido.
3. **Listar Voluntários:** Exibe todos os voluntários cadastrados.
4. **Salvar Lista de Voluntários:** Salva a lista de voluntários em um arquivo de texto no computador.

## Explicação do Código

### 1. Importações e Configuração do Banco de Dados

python

Copiar código

```
import tkinter as tk
from tkinter import messagebox, filedialog
import sqlite3
```

- **tkinter:** Usado para criar a interface gráfica.
- **sqlite3:** Usado para gerenciar o banco de dados local onde os voluntários são armazenados.

python

Copiar código

```
conn = sqlite3.connect('esslog_voluntarios.db')
c = conn.cursor()
```

- Conexão e cursor do banco de dados SQLite são criados.
- O arquivo `esslog_voluntarios.db` é onde os dados de voluntários serão armazenados.

python

Copiar código

```
c.execute('''
    CREATE TABLE IF NOT EXISTS voluntarios (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        nome TEXT NOT NULL,
        email TEXT NOT NULL
    )
''')
conn.commit()
```

- Aqui, criamos a tabela `voluntarios`, se ela ainda não existir. Ela possui três campos: `id`, `nome`, e `email`.

### 2. Funções Principais

#### a. Adicionar Voluntário

python

Copiar código

```
def adicionar_voluntario():
    nome = entry_nome.get()
    email = entry_email.get()
    if nome and email:
        c.execute('INSERT INTO voluntarios (nome, email) VALUES (?, ?)', (nome, email))
        conn.commit()
        messagebox.showinfo("Sucesso", "Voluntário adicionado!")
        listar_voluntarios()
        entry_nome.delete(0, tk.END)
        entry_email.delete(0, tk.END)
    else:
        messagebox.showwarning("Erro", "Por favor, preencha todos os campos.")
```

- Esta função insere um voluntário no banco de dados se os campos de nome e e-mail forem preenchidos corretamente.
- Uma mensagem de sucesso é exibida ao concluir o cadastro.

## b. Remover Voluntário

python

Copiar código

```
def remover_voluntario():
    id_voluntario = entry_id.get()
    if id_voluntario:
        c.execute('DELETE FROM voluntarios WHERE id = ?', (id_voluntario,))
        conn.commit()
        messagebox.showinfo("Sucesso", "Voluntário removido!")
        listar_voluntarios()
        entry_id.delete(0, tk.END)
    else:
        messagebox.showwarning("Erro", "Por favor, insira o ID do voluntário.")
```

- Remove o voluntário cujo ID for fornecido. Exibe uma mensagem de sucesso ou erro, dependendo do input.

## c. Listar Voluntários

python

Copiar código

```
def listar_voluntarios():
    lista.delete(0, tk.END)
    c.execute('SELECT * FROM voluntarios')
    for voluntario in c.fetchall():
        lista.insert(tk.END, f"ID: {voluntario[0]}, Nome: {voluntario[1]}, Email: {voluntario[2]}")
```

- Esta função exibe os voluntários cadastrados no banco de dados em um ListBox, formatando as informações de forma organizada.

## d. Salvar Lista de Voluntários

python

Copiar código

```
def salvar_lista():
    voluntarios = c.execute('SELECT * FROM voluntarios').fetchall()
    if voluntarios:
        filepath = filedialog.asksaveasfilename(defaultextension=".txt",
                                                filetypes=[("Text files", "*.txt"),
("All files", "*.*")])
        if filepath:
            with open(filepath, 'w') as file:
                for voluntario in voluntarios:
                    file.write(f"ID: {voluntario[0]}, Nome: {voluntario[1]}, Email:
{voluntario[2]}\n")
            messagebox.showinfo("Sucesso", "Lista de voluntários salva!")
        else:
            messagebox.showwarning("Aviso", "Nenhum voluntário para salvar!")
```

- Esta função permite salvar a lista de voluntários cadastrados em um arquivo de texto, utilizando a janela de diálogo de salvamento (filedialog).

### 3. Interface Gráfica (GUI)

#### a. Janela Principal

python

Copiar código

```
janela = tk.Tk()
janela.title("Gerenciamento de Voluntários ESSLOG")
janela.state('zoomed')
```

- A janela principal é criada e maximizada com o método `state('zoomed')` para preencher toda a tela.

#### b. Frame e Organização dos Componentes

python

Copiar código

```
frame_principal = tk.Frame(janela, padx=20, pady=20)
frame_principal.pack(expand=True)
```

- O frame principal (frame\_principal) é centralizado e contém todos os widgets (rótulos, campos de entrada, botões, etc.).

#### c. Componentes de Entrada e Botões

python

Copiar código

```
tk.Label(frame_principal, text="Nome", font=('Arial', 12)).grid(row=0, column=0,
sticky=tk.W, pady=10)
entry_nome = tk.Entry(frame_principal, width=40, font=('Arial', 12))
entry_nome.grid(row=0, column=1, pady=10)
```

- Rótulos (`Label`), campos de entrada (`Entry`), e botões (`Button`) são organizados em um grid, e todos têm uma formatação consistente, com espaçamento (`padx`, `pady`) para melhorar o layout.

#### d. Lista de Voluntários (`ListBox`)

python

Copiar código

```
lista = tk.Listbox(frame_principal, width=80, height=10, font=('Arial', 12))
lista.grid(row=5, column=0, columnspan=2, pady=10)
```

- A `ListBox` exibe os voluntários cadastrados de forma legível.

### 4. Fechamento da Aplicação

python

Copiar código

```
janela.mainloop()
conn.close()
```

- A função `mainloop()` mantém a janela aberta até que o usuário a feche.
- Após o fechamento da janela, a conexão com o banco de dados é encerrada corretamente com `conn.close()`.

## *Conclusão*

Este sistema simples permite que os usuários gerenciem os voluntários de maneira eficiente, com operações de CRUD (Create, Read, Update, Delete) básicas. A interface gráfica é amigável e centraliza os elementos para uma boa experiência de usuário.