

# Resumo da Aula: Projeto de Cadastro Automático de Alunos

Nesta aula, iniciamos o desenvolvimento de uma aplicação gráfica com Tkinter para cadastrar alunos, com campos para nome e e-mail, e uma tabela para exibir os dados. O objetivo é criar uma interface que será automatizada com PyAutoGUI em uma aula futura, permitindo o preenchimento automático dos campos.

## Criação da Aplicação com Tkinter

A aplicação é construída usando o módulo `tkinter`, nativo do Python. Criamos uma janela principal com título "Cadastro de Alunos":

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
root.title("Cadastro de Alunos")
```

## Tabela TreeView

Uma tabela (`ttk.Treeview`) exibe os dados cadastrados, com colunas para "name" e "email":

```
tree = ttk.Treeview(root, columns=('name', 'email'))
tree.heading('name', text='Name')
tree.heading('email', text='Email')
tree.pack()
```

A tabela é configurada para exibir os nomes e e-mails dos alunos cadastrados.

## Campos de Entrada (Nome e E-mail)

Dois campos de entrada (`tk.Label` e `tk.Entry`) são criados para nome e e-mail:

```
label_name = tk.Label(root, text='Name')
label_name.pack()
entry_name = tk.Entry(root)
entry_name.pack()

label_email = tk.Label(root, text='Email')
label_email.pack()
entry_email = tk.Entry(root)
entry_email.pack()
```

Esses campos permitem ao usuário inserir os dados manualmente.

## Botão de Adicionar

Um botão "Add" é adicionado para inserir os dados na tabela:

```
def add_student():
    name = entry_name.get()
    email = entry_email.get()
    tree.insert('', 'end', values=(name, email))
    entry_name.delete(0, tk.END)
    entry_email.delete(0, tk.END)

button_add = tk.Button(root, text='Add', command=add_student)
button_add.pack()
```

A função `add_student` captura os valores dos campos, insere-os na tabela e limpa os campos após a inserção.

## Executando a Aplicação

O loop principal do Tkinter mantém a janela ativa:

```
root.mainloop()
```

Ao executar, a aplicação exibe uma janela com campos para nome e e-mail, um botão "Add" e uma tabela que lista os alunos cadastrados.

## Gerando o Executável

Para criar um executável, usamos o `PyInstaller`:

```
pip install pyinstaller
pyinstaller --onefile 23_gui.py
```

Isso gera um arquivo executável na pasta `dist`, permitindo rodar a aplicação sem o Python instalado.

## Pontos-Chave

- **Interface Gráfica:** Tkinter é usado para criar uma janela com campos de entrada, um botão e uma tabela.
- **Tabela:** `ttk.Treeview` exibe os dados em colunas (nome e e-mail).
- **Funcionalidade:** O botão "Add" insere os dados na tabela e limpa os campos.
- **Executável:** PyInstaller converte o script em um arquivo executável.
- **Próximos Passos:** A aplicação será automatizada com PyAutoGUI para preenchimento automático dos campos.

## Dicas para Alunos

- Teste a aplicação incrementalmente, verificando cada componente (tabela, campos, botão).
- Use nomes consistentes para variáveis (ex.: `entry_name`, `entry_email`).

- Certifique-se de ter o `PyInstaller` instalado antes de gerar o executável.
- Prepare uma lista de nomes e e-mails para testar a automação com `PyAutoGUI` na próxima aula.