



# MANUAL DO USUÁRIO

LIK. EN

RODRIGO B. PAZOTE





FETS - FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA  
DEE - DEPARTAMENTO DE ENGENHARIA ELÉTRICA

# MANUAL DO USUÁRIO

LIK.EN

RODRIGO B. PAZOTE

# Sumário

1	Apresentação	1
2	Como usar o programa	2
3	Como funciona o programa	4
4	Possíveis desafios que você vai enfrentar	6
5	SAC	8

# 1 Apresentação

Olá,

Se você está lendo isso, provavelmente está tendo problemas com o programa que eu desenvolvi. Bom, se o problema for muito sério pode me mandar um e-mail: ou me contatar pelo whatsapp (Informações disponíveis no SAC). Se o problema não parecer tão sério, pode ser que você encontre ele aqui e consiga resolver por conta própria.

A versão do programa que você tem é bem simples, as linhas de código estão no meu perfil no github de forma pública para caso você queira alterar ou melhorar. Você pode baixar uma versão mais recente também se eu já desenvolvi.

A idéia deste programa é filtrar as tabelas de coletas de espécies de plantas para facilitar o trabalho manual de algumas pesquisas realizadas no departamento de biologia na FEIS UNESP. Se você está usando para outro propósito, provavelmente precisará alterar o código fonte pois este foi feito como um sapatinho de cristal para os problemas enfrentados pelos protetores de florestas da minha universidade.

Por problemas, como muito tempo livre, eu escalei este trabalho em um nível visivelmente desnecessário e, se você está lendo isso, já deve ter percebido que este próprio manual é completamente desnecessário mas eu quis fazer e agradeço a você por ler.

At.te

Rodrigo Barbosa Pazote

## 2 Como usar o programa

A este ponto, imagino que você tenha pelo menos executado o programa para conseguir abrir o manual de usuário, então vamos partir daqui.

Figura 1: Tela inicial do programa



Fonte: LIK.EN

Para usar a ferramenta de filtragem, primeiro você precisa apertar os botões escritos "procurar" e selecionar os dois arquivos \*em excel\* para que o programa leia e processe esses dados. Quando você selecionar um arquivo, o caminho deste arquivo deve aparecer na esquerda do botão, na região cinza. Se não funcionar, tente de novo, se mesmo assim não funcionar, baixe ele de novo pelo github.

Após selecionadas as duas tabelas, basta apertar o botão "Funcionar" na posição inferior do programa que a sua tabela resultante, filtrada e perfeita vai aparecer na mesma pasta que você está rodando o programa com o nome "TabelaFiltrada.xlsx".

As funções de escolher a pasta para onde vai enviar o arquivo ainda não está disponível na versão 2.1 do projeto.

\*ATENÇÃO: VOCÊ PRECISA ANTES DEIXAR A TABELA ALINHADA QUANTO AS NORMAS DE FUNCIONAMENTO DO PROGRAMA QUE SÃO: USAR OS NOMES CERTOS PARA CADA COLUNA DA TABELA E NA MESMA ORDEM. OU SEJA, UM PRÉ REQUISITO PARA USAR ESTE PROGRAMA É SABER COMO MOVER COLUNAS E MUDAR DADOS NO EXCEL.\* De qualquer forma, fiz um tutorial básico de como fazer isso no meu canal no YouTube:  
<https://www.youtube.com/channel/UCNNlwViSiuYIwLxquF4cKrg>

A seguir, tem o modelo de títulos que você deve seguir para que o programa funcione corretamente:

ESTADO  
HERBÁRIO DE ORIGEM  
N° DE CATÁLOGO  
FAMÍLIA;GÊNERO  
NOME CIENTÍFICO  
MODO DE VIDA  
AUTOR DO TÁXON  
DE:  
ANO DE COLETA  
MÊS DE COLETA  
DIA DE COLETA  
COLETOR  
N° DA COLETA  
ID. POR MUNICÍPIO  
MUNICÍPIO  
LONGITUDE  
LATITUDE  
DESCRIÇÃO DA PLANTA  
OBSERVAÇÕES

Basta copiar e colar eles na mesma ordem que eu mandei que vai dar tudo certo.

### 3 Como funciona o programa

Bom, a partir daqui eu explico fundamentos mais detalhados do sistema e preciso que você tenha alguns conhecimentos básicos em lógica de programação em Python, não é nada complicado mas se você não entender Python básico vc não vai entender o que eu mostrar aqui.

A essência do programa gira em torno deste primeiro projeto:

Figura 2: Back-end base do programa

```
#Importando tabelas

import pandas as pd
import io

tabela1 = pd.read_excel("RelatorioReflora.xlsx")
tabela2 = pd.read_excel("RelatorioSpecies.xlsx")
display(tabela2)
display(tabela1)

#Agrupando Dados
tabela_agrupada = pd.concat([tabela1, tabela2])
display(tabela_agrupada)

#Removendo duplicatas
df = tabela_agrupada.value_counts()
df = tabela_agrupada.drop_duplicates()
df = tabela_agrupada.drop_duplicates(subset = ['Nº DA COLETA', 'COLETOR'], keep='first')
df = tabela_agrupada.drop_duplicates(subset = ['Nº DA COLETA', 'DE: '], keep='first')

display(df)

duplicadas = tabela_agrupada

#Exportando arquivo
tabela_agrupada.to_excel("planilhaagrupada.xlsx", index=False, encoding='latin1')
df.to_excel("Planilhaamoana.xlsx", index=False, encoding='latin1')
```

Fonte: LIK.EN

Este programa que você está entrando em contato agora é uma recriação mais caprichada deste primeiro protótipo em blocos. Como pode perceber, está tudo indicado e separado, basicamente importamos as bibliotecas pandas e io que servem especificamente para tratar com tabelas e caminhos de arquivos, importamos ambas as tabelas, concatenamos (uma palavra bonita para agrupar, isso leva em conta os nomes presentes nas colunas, por isso o módulo anterior é muito importante). Feito a junção das duas tabelas, temos um DataFrame resultante que só precisa ser tratado.

Para tratar o dataframe, primeiro tiramos as duplicatas completamente idênticas, depois usamos 3 parâmetros: se o numero da coleta e o coletor for o mesmo em mais de uma linha, a segunda é retirada mantendo apenas uma e o mesmo ocorre se o numero da coleta e a data for a mesma. Obviamente não será feito o mesmo para caso a data e o coletor forem os mesmos pois uma pessoa pode executar mais de uma coleta por dia o que pode causar perda de dados.

O programa que você está tendo contato tem uma base um pouco mais refinada, a base ainda é a mesma mas feita por funções que são usadas por botões no programa:

Figura 3: Back-end base do programa atual

```
1 import pandas as pd
2 from tkinter import *
3 import tkinter as tk
4 from tkinter.filedialog import askopenfilename
5 import os
6
7 def selecionar_arquivo():
8     path = askopenfilename(title="Selecione um arquivo em Excel para abrir")
9     return path
10
11 def ler_tabela(caminho):
12     path = caminho
13     tabela = pd.read_excel(path)
14     return tabela
15
16 def concatenar_tabelas(tabela_1, tabela_2):
17     tabela_agrupada = pd.concat([tabela_1, tabela_2])
18     return tabela_agrupada
19
20 def remover_duplicatas(tabela_agrupada):
21     df = tabela_agrupada.value_counts()
22     df = tabela_agrupada.drop_duplicates()
23     df = tabela_agrupada.drop_duplicates(subset = ['Nº DA COLETA', 'COLETOR'], keep='first')
24     df = tabela_agrupada.drop_duplicates(subset = ['Nº DA COLETA', 'DE:'], keep='first')
25     return df
26
27 def exportar_tabela(tabela_exportar):
28     tabela_exportar.to_excel("TabelaFiltrada.xlsx", index=False, encoding='latin1')
29
30 def mostrar_tabela(tabela):
31     print(tabela)
32
33
34 def rodando(path_1, path_2):
35     df_1 = pd.read_excel(path_1.get())
36     df_2 = pd.read_excel(path_2.get())
37     tabela_duplicada = concatenar_tabelas(df_1, df_2)
38     tabela_filtrada = remover_duplicatas(tabela_duplicada)
39     exportar_tabela(tabela_filtrada)
40
41 # ... (rest of the code) ...
```

Fonte: LIK.EN

Como pode perceber, o projeto linear de antes agora é um conjunto de funções que são chamados por outras funções pelos botões do front-end:

Figura 4: Back-end dos botões

```
43 #Botão Procurar primeira tabela.
44 def btn_clicked_2():
45     arquivo_1 = selecionar_arquivo()
46     path_1.set(arquivo_1)
47     if path_1:
48         entry0['text'] = f"Arquivo selecionado: {arquivo_1}"
49
50 #Botão procurar segunda tabela.
51 def btn_clicked_3():
52     arquivo_2 = selecionar_arquivo()
53     path_2.set(arquivo_2)
54     if path_2:
55         entry2['text'] = f"Arquivo selecionado: {arquivo_2}"
56
57 #Botão procurar pasta de salvamento.
58 def btn_clicked_4():
59     entry1['text'] = f"Indisponível"
60
61 #Botão Manual de instruções.
62 def btn_clicked_5():
63     os.startfile("manual.txt")
64
65 #Botão funcionar.
66 def btn_clicked():
67     rodando(path_1, path_2)
```

Fonte: LIK.EN

O restante do projeto é firulinha gráfica como aquele patinho do canto inferior esquerdo fazendo uma piada incrível que eu sei que você adora < 3.



## 4 Possíveis desafios que você vai enfrentar

Espero que você não precise lêr esta seção, mas como você está lendo vou supor que "deu ruim", bom, vou apresentar alguns dos problemas que você pode ter enfrentado:

Figura 5: Colunas em branco

Arquivos

Página inicial

Inserir

Layout da Página

Formulas

Dados

Revisão

Exibir

Ajuda

Digite-me o que você deseja fazer

Calcular

11

A

A

Quadrar Texto Automaticamente

Geral

%

60

00

Formatar como Tabela

Normal

Bom

Neutro

Formatação de Células

Estilos

Inserir

Excluir

Formatar

Autocorr

Limpar

Classificar

Localizar e Selecionar

Cor

Recortar

Copiar

Princípio de Formatação

Área de Transfêrência

Fonte

Alinhamento

Numeração

Estado

A1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

AA

AB

AC

AD

AE

AF

AG

AH

AI

AJ

AK

AL

AM

AN

AO

AP

AQ

AR

AS

AT

AU

AV

AW

AX

AY

AZ

BA

BB

BC

BD

BE

BF

BG

BH

BI

BJ

BK

BL

BM

BN

BO

BP

BQ

BR

BS

BT

BU

BV

BW

BX

BY

BZ

CA

CB

CC

CD

CE

CF

CG

CH

CI

CJ

CK

CL

CM

CN

CO

CP

CQ

CR

CS

CT

CU

CV

CW

CX

CY

CZ

DA

DB

DC

DD

DE

DF

DG

DH

DI

DJ

DK

DL

DM

DN

DO

DP

DQ

DR

DS

DT

DU

DV

DW

DX

DY

DZ

EA

EB

EC

ED

EE

EF

EG

EH

EI

EJ

EK

EL

EM

EN

EO

EP

EQ

ER

ES

ET

EU

EV

EW

EX

EY

EZ

FA

FB

FC

FD

FE

FF

FG

FH

FI

FJ

FK

FL

FM

FN

FO

FP

FQ

FR

FS

FT

FU

FV

FW

FX

FY

FZ

GA

GB

GC

GD

GE

GF

GG

GH

GI

GJ

GK

GL

GM

GN

GO

GP

GQ

GR

GS

GT

GU

GV

GW

GX

GY

GZ

HA

HB

HC

HD

HE

HF

HG

HH

HI

HJ

HK

HL

HM

HN

HO

HP

HQ

HR

HS

HT

HU

HV

HW

HX

HY

HZ

IA

IB

IC

ID

IE

IF

IG

IH

II

IJ

IK

IL

IM

IN

IO

IP

IQ

IR

IS

IT

IU

IV

IW

IX

IY

IZ

JA

JB

JC

JD

JE

JF

JG

JH

JI

IJ

JK

KL

LM

LN

LO

LP

LQ

LR

LS

LT

LU

LV

LW

LX

LY

LZ

MA

MB

MC

MD

ME

MF

MG

MH

MI

MJ

MK

ML

MM

MN

MO

MP

MQ

MR

MS

MT

MU

MV

MW

MX

MY

MZ

NA

NB

NC

ND

NE

NF

NG

NH

NI

NJ

NK

NL

NM

NN

NO

NP

NQ

NR

NS

NT

NU

NV

NW

NX

NY

NZ

OA

OB

OC

OD

OE

OF

OG

OH

OI

OJ

OK

OL

OM

ON

OO

OP

OQ

OR

OS

OT

OU

OV

OW

OX

OY

OZ

PA

PB

PC

PD

PE

PF

PG

PH

PI

PJ

PK

PL

PM

PN

PO

PP

PQ

PR

PS

PT

PU

PV

PW

PX

PY

PZ

QA

QB

QC

QD

QE

QF

QG

QH

QI

QJ

QK

QL

QM

QN

QO

QP

QQ

QR

QS

QT

QU

Fonte: Próprio autor

Isso aconteceu porque você não colocou as colunas direito igual eu mandei, dá uma lida na primeira seção que vc resolve isso rapidinho e é só rodar de novo o programa, n precisa nem apagar a tabela resultante do erro.

Figura 6: Erro 2

The image shows a screenshot of a Microsoft Excel spreadsheet. The title bar indicates the file is named 'Tabela filtrada - Excel'. The ribbon at the top includes tabs for 'Arquivo', 'Página Inicial', 'Inserir', 'Layout da Página', 'Formulas', 'Dados', 'Referências', 'Exibir', and 'Ajuda'. The 'Página Inicial' tab is active, showing options for font, paragraph, and styles. The spreadsheet itself contains a large table with columns labeled A through AF. The first few columns (A-D) contain text, likely species names and locations. Columns E through AF contain numerical data. The table is organized into rows, with the first row (row 1) containing headers. The data appears to be a list of species observations with associated measurements and locations. The spreadsheet is displayed on a grid, and the status bar at the bottom shows 'Pronto' and 'Acessibilidade: tudo certo'.

Fonte: Próprio autor

Cara, o programa é muito simples, não consegui achar outra forma de você cometer um erro. Qualquer outro problema, pode ir direto para o SAC

## 5 SAC

Bem vindo ao sistema de atendimento ao cliente, bom, se você leu até aqui é porque a coisa tá feia mesmo e precisa de mim. tenho atendimento 24h no email e no Whatsapp abaixo, só chamar que eu respondo assim que ver:

(18)997308738  
rodrigopazote@hotmail.com

Reclamações:

farias.seabra@unesp.br