



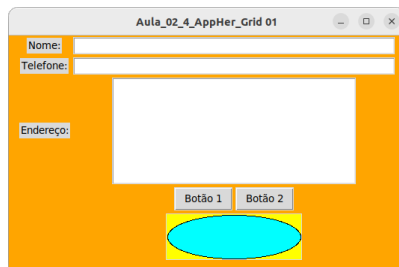
Iniciado em	terça-feira, 5 dez. 2023, 18:55
Estado	Finalizada
Concluída em	terça-feira, 5 dez. 2023, 20:10
Tempo empregado	1 hora 14 minutos
Notas	33,90/35,00
Avaliar	9,69 de um máximo de 10,00(96,86%)



Questão 1

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo, responda:**Janela.py**

```
01... from tkinter import *
02...
03... #####
04...
05... class Janela(Tk):
06...     def __init__(self, Str="Janela", px=0, py=0, dx=640, dy=480, cor="lighthgray"):
07...         super().__init__()
08...         super().title(Str)
09...         super().geometry("%dx%d+%d+%d" % (dx, dy, px, py))
10...         super().configure(bg=cor)
11...         self.inicialize()
12...
13...     def inicialize(self):
14...         Lb1=Label(self, text="Nome:")
15...         Lb2=Label(self, text="Telefone:")
16...         Lb3=Label(self, text="Endereço:")
17...
18...         Et1=Entry(self, width=52)
19...         Et2=Entry(self, width=52)
20...
21...         Txt1=Text(self, height=8, width=40)
22...
23...         Bt1=Button(self, text='Botão 1')
24...         Bt2=Button(self, text='Botão 2')
25...
26...         Cnv=Canvas(self, bd=0, bg="yellow", width=182, height=62)
27...         Cnv.create_oval(2, 2, 180, 60, fill = "cyan")
28...
29...         Lb1.grid(row=0, column=0, padx=2, pady=2)
30...         Lb2.grid(row=1, column=0, padx=2, pady=2)
31...         Lb3.grid(row=2, column=0, padx=2, pady=2)
32...         Et1.grid(row=0, column=1, columnspan=2, padx=2, pady=2)
33...         Et2.grid(row=1, column=1, columnspan=2, padx=2, pady=2)
34...         Txt1.grid(row=2, column=1, columnspan=2, padx=2, pady=2)
35...         Bt1.grid(row=3, column=1, sticky=E, padx=2, pady=2)
36...         Bt2.grid(row=3, column=2, sticky=W, padx=2, pady=2)
37...
38...         Cnv.grid(row=4, column=1, columnspan=2, padx=2, pady=2)
39...
40... #####
```

Aula_02_4_AppHer_Grid.py

```
41... from Janela import Janela
42...
43... #####
44...
45... Jan1 = Janela("Minha janela", 400, 200, 530, 310, "orange")
46... Jan1.mainloop()
47...
48... #####
```

Quais as alternativas corretas?

Escolha uma ou mais:

- ☐ a. Na linha 09 do programa há um `super(Janela, self).geometry(dx,`
- ☐ b. Na linha 10 do programa há um `super(Janela, self).configure(cor`
- ☒ c. Na linha 09 do programa há um `super(Janela, self).geometry("%c`
- ☐ d. Na linha 10 do programa há um `self.setConfiguracao(bg=cor)`
- ☒ e. Na linha 14 do programa o coma `"Nome:"`

Questão 2

Correto

Atingiu 1,00 de 1,00



Na questão anterior um comentário na linha 24 causa o seguinte efeito:

Escolha uma ou mais:

- ☐ a. O botão 2 fica com rótulo vazio
- ☐ b. Ocorre um erro de execução na linha 35
- ☒ c. Ocorre um erro de execução na linha 36 ✓
- ☐ d. O programa continua funcionando normalmente exceto pelo fato do botão 2 não aparecer na tela
- ☒ e. Ocorre o seguinte erro: `NameError: name 'Bt2' is not defined` ✓

Questão 3

Correto

Atingiu 1,00 de 1,00

Em relação ao código apresentado anteriormente, quais as alternativas estão corretas?

Escolha uma ou mais:

- ☐ a. Na linha 19 há um erro, o correto seria: `Et2=Textfield(width=52)`
- ☒ b. O comando da linha 19 cria uma caixa de digitação de Strings ✓
- ☐ c. Na linha 21 o correto seria: `Txt1=TextArea(altura=8, largura=40)`
- ☒ d. Na linha 21 não há necessidade de transformar o objeto **Txt1** em variável global, pois ele não será utilizado fora do contexto em que foi criado. ✓
- ☐ e. O comando da linha 23 cria um botão e atribui a ele um evento, por isso ele obrigatoriamente deve ser uma variável global.



Questão 4

Correto

Atingiu 1,00 de 1,00

No código apresentado anteriormente pode-se afirmar o seguinte a respeito do componente *Canvas* :

Escolha uma ou mais:

- ☒ a. É um componente gráfico que pode ser utilizado em Janelas na biblioteca TKinter do Python ✓
- ☐ b. O comando da linha 27 cria um círculo com raio de 180 pixels no Canvas.
- ☒ c. O comando da linha 26 determina que o fundo do Canvas seja amarelo ✓
- ☐ d. O comando da linha 27 cria uma oval com borda de cor **Cyan** e preenchimento verde
- ☒ e. O comando da linha 26 determina que a largura do Canvas seja 182 pixels e sua altura 62 pixels. ✓



Questão 5

Correto

Atingiu 1,00 de 1,00

Sobre o GridLayout, apresentado no código anterior, é verdadeiro afirmar que:

Escolha uma ou mais:

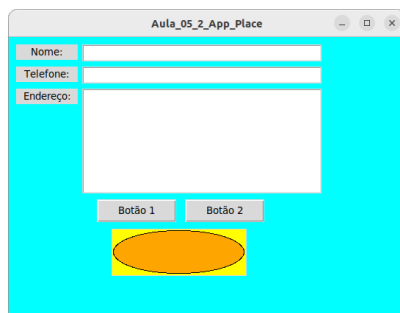
- ☐ a. Os parâmetros apresentados no comando da linha 29 são:
Linha, Coluna, posição x, posição y
- ☒ b. Os parâmetros apresentados no comando da linha 29 são:
Linha, Coluna, separação entre componentes no eixo x, separação entre componentes no eixo y ✓
- ☐ c. Na linha 32 o parâmetro `columnspan=2` significa que duas colunas não serão ocupadas por nenhum componente
- ☒ d. Na linha 32 o parâmetro `columnspan=2` significa que o componente ocupará duas colunas. ✓
- ☒ e. Na linha 35 o parâmetro `sticky=E` significa que o botão ocupará o lado leste da segunda coluna. ✓



Questão 6

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo:

Janela.py	
01...	from tkinter import *
02...	
03...	#####
04...	
05...	class Janela(Tk):
06...	def __init__(self, Str="Janela", px=0, py=0, dx=640, dy=480, cor="ligthgray"):
07...	super(Janela, self).__init__()
08...	super(Janela, self).title(Str)
09...	super(Janela, self).geometry("%dx%d+%d+%d" % (dx, dy, px, py))
10...	super(Janela, self).configure(bg=cor)
11...	self.inicialize()
12...	
13...	def Posiciona_Componentes(self):
14...	self.Lb1.place(x=10, y=10)
15...	self.Lb2.place(x=10, y=40)
16...	self.Lb3.place(x=10, y=70)
17...	self.Et1.place(x=100, y=10)
18...	self.Et2.place(x=100, y=40)
19...	self.Txt1.place(x=100, y=70)
20...	self.Bt1.place(x=120, y=220)
21...	self.Bt2.place(x=240, y=220)
22...	self.Cnv.place(x=140, y=260)
23...	
24...	def inicialize(self):
25...	self.Lb1 = Label(self, text="Nome:", height=1, width=10)
26...	self.Lb2 = Label(self, text="Telefone:", height=1, width=10)
27...	self.Lb3 = Label(self, text="Endereço:", height=1, width=10)
28...	
29...	self.Et1 = Entry(self, width=40)
30...	self.Et2 = Entry(self, width=40)
31...	
32...	self.Txt1 = Text(self, height=8, width=40)
33...	
34...	self.Bt1 = Button(self, text='Botão 1', height=1, width=10)
35...	self.Bt2 = Button(self, text='Botão 2', height=1, width=10)
36...	
37...	self.Cnv = Canvas(self, bd=0, bg="yellow", width=182, height=62)
38...	self.Cnv.create_oval(2, 2, 180, 60, fill="orange")
39...	
40...	self.Posiciona_Componentes()
41...	
42...	#####

Aula_05_2_App_Place.py

```
43... from Janela import Janela
44...
45... #####
46...
47... Jan1=Janela("Minha janela", 400, 200, 540, 380, "cyan")
48... Jan1.mainloop()
49...
50... #####
```

A omissão da linha 40 através de um comentário causa o seguinte efeito:

Escolha uma ou mais:

- ☐ a. Ocorre um erro de execução na linha 47, NullPointerException, pois não houve posicionamento dos componentes na tela
- ☐ b. Quando omitimos o Layout de apresentação da tela, o Python assume o Gridlayout como Layout padrão
- ☒ c. O Python roda o programa e omite a apresentação dos componentes ✓
- ☐ d. Ocorre um erro de execução na linha 48, pois o sistema tentará posicionar um componente na tela
- ☒ e. O Python roda o programa e apresenta uma tela vazia ✓



Questão 7

Correto

Atingiu 2,00 de 2,00

Considerando o código apresentado anteriormente, responda qual o significado dos comandos:

01	<code>__init__</code>	Construtor da classe
02	<code>place</code>	Comando para posicionar componentes
03	<code>super(Janela, self).__init__()</code>	Chama construtor da classe base
04	<code>geometry</code>	Comando para posicionar a janela na tela e atribuir uma largura e altura
05	<code>self</code>	Faz referência a variáveis globais ou métodos da própria classe
06	<code>Entry</code>	Componente visual para digitação de texto de uma única linha
07	<code>Text</code>	Componente visual para digitação de texto de várias linhas
08	<code>Button</code>	Componente visual que representa um botão na interface
09	<code>Canvas</code>	componente visual para desenhar figuras
10	<code>mainloop</code>	Cria um loop infinito para que a janela fique na tela até que o usuário feche-a



Questão 8

Correto

Atingiu 1,00 de 1,00

Considerando o código apresentado anteriormente, responda quais as alternativas corretas:

Escolha uma ou mais:

- ☐ a. O comando **super(Janela, self).__init__()** pode ser substituído por **super(Tk).__init__()**
- ☒ b. O comando **super(Janela, self).__init__()** pode ser substituído por **super().__init__()** ✓
- ☒ c. O comando **super(Janela, self).__init__()** refere-se ao Python 2 ✓
- ☒ d. O comando **super().__init__()** refere-se ao Python 3
- ☒ e. O comando **super(Janela, self).title(Str)** atribui um título à janela ✓



Questão 9

Correto

Atingiu 1,00 de 1,00

Analise o código abaixo:

Aula_06_2a_App_1Cam_Aluno

Nome=	João
Sexo=	Masc.
Nota1=	7.5
Nota2=	8.6
Nota3=	9.4
Média=	8.50
Calcular	

Janela.py



```
01... from tkinter import *
02...
03... #####
04...
05... class Janela(Tk):
06...     __Et_Nota1=None
07...     __Et_Nota2=None
08...     __Et_Nota3=None
09...     __Et_Media=None
10...
11...     #####
12...     ##                                ##
13...     ##          Complete o código          ##
14...     ##                                ##
15...     ##                                ##
16...     #####
17...
18...     def action_Bt_Calc(self):
19...         #####
20...         ##                                ##
21...         ##    Complete o código do evento    ##
22...         ##                                ##
23...         ##                                ##
24...         #####
25...
26...     def inicialize(self):
27...         #####
28...         ##                                ##
29...         ##    Complete o código de alocação    ##
30...         ##                                ##
31...         ##                                ##
32...         #####
33...
34...         Lb_Nome.configure(bg='yellow')
35...         Lb_Sexo.configure(bg='yellow')
36...         Lb_Nota1.configure(bg='yellow')
37...         Lb_Nota2.configure(bg='yellow')
38...         Lb_Nota3.configure(bg='yellow')
39...         Lb_Media.configure(bg='yellow')
40...
41...         Et_Nome=Entry(self, width=52)
42...         self.__Et_Nota1=Entry(self, width=52)
43...         self.__Et_Nota2=Entry(self, width=52)
44...         self.__Et_Nota3=Entry(self, width=52)
45...         self.__Et_Media=Entry(self, width=52)
46...
47...         var=StringVar(self)
48...         var.set('Escolha a opção')
49...         OM_Sexo=OptionMenu(self, var, 'Escolha a opção', 'Masc.', 'Fem.')
50...         OM_Sexo.config(width=32)
51...
52...         Bt_Calc=Button(self, text='Calcular', command=self.action_Bt_Calc)
53...
54...         #####
55...         ##                                ##
56...         ##    Complete o código para acrescentar ##
57...         ## os componentes faltantes na tela    ##
58...         ##                                ##
59...         #####
60...         Bt_Calc.grid(row=6, column=0, sticky=NW, padx=4, pady=4)
61...
62...         Et_Nome.grid(row=0, column=1, sticky=NW, padx=4, pady=4)
63...         OM_Sexo.grid(row=1, column=1, sticky=NW, padx=4, pady=4)
64...         self.__Et_Nota1.grid(row=2, column=1, sticky=NW, padx=4, pady=4)
```



65...	self.__Et_Nota2.grid(row=3, column=1, sticky=NW, padx=4, pady=4)
66...	self.__Et_Nota3.grid(row=4, column=1, sticky=NW, padx=4, pady=4)
67...	self.__Et_Media.grid(row=5, column=1, sticky=NW, padx=4, pady=4)
68...	
69...	#####

Aula_06_2a_App_1Cam_Aluno.py	
70...	from Janela import Janela
71...	
72...	#####
73...	
74...	Jan1=Janela("Minha janela", 400, 200, 540, 380, "orange")
75...	Jan1.mainloop()
76...	
77...	#####

As linhas de código suprimidas nas
linhas 11 à 16 devem conter o
construtor da classe Janela.

Qual a alternativa correta para a sua
implementação?

Escolha uma ou mais:

☐ a.

11...	def Construtor
12...	super().__init__
13...	super().__init__
14...	super(Janela)
15...	super(Janela)
16...	self.inicializar

☐ b.

11...	definir init
12...	super().__init__
13...	super().__init__
14...	super(Janela)
15...	super(Janela)
16...	self.inicializar

☐ c.

11...	def __init__(self):
12...	super().__init__
13...	super().__init__
14...	super(Janela)
15...	super(Janela)
16...	self.inicializar

☒ d.

11...	def __init__(self):
12...	super().__init__
13...	super().__init__
14...	super(Janela)
15...	super(Janela)
16...	self.inicializar



<input type="checkbox"/> e.	11...	def __Janela_
	12...	super(thi
	13...	super(thi
	14...	super(Jar
	15...	super(Jar
	16...	self.inic

Questão 10

Correto

Atingiu 1,00 de 1,00

As linhas de código de 27 à 32
necessárias para alocar todos labels
amarelos são:

Escolha uma ou mais:

<input type="checkbox"/> a.	27...	Lb_Nome=Lak
	28...	Lb_Sexo=Set
	29...	Lb_Nota1=Lε
	30...	Lb_Nota2=Lε
	31...	Lb_Nota3=Lε
	32...	Lb_Media=Lε

<input checked="" type="checkbox"/> b.	27...	Lb_Nome=Lak
	28...	Lb_Sexo=Lak
	29...	Lb_Nota1=Lε
	30...	Lb_Nota2=Lε
	31...	Lb_Nota3=Lε
	32...	Lb_Media=Lε

<input type="checkbox"/> c.	27...	Lb_Nome=Lak
	28...	Lb_Sexo=Lak
	29...	Lb_Nota1=Lε
	30...	Lb_Nota1=Lε
	31...	Lb_Nota1=Lε
	32...	Lb_Media=Lε

<input type="checkbox"/> d.	27...	Lb_Nome=Lak
	28...	Lb_Nome=Lak
	29...	Lb_Nota2=Lε
	30...	Lb_Nota2=Lε
	31...	Lb_Nota2=Lε
	32...	Lb_Media=Lε

<input type="checkbox"/> e.	27...	Lb_Nome=Ent
	28...	Lb_Sexo=Ent
	29...	Lb_Nota1=Er
	30...	Lb_Nota2=Lε
	31...	Lb_Nota3=Lε
	32...	Lb_Media=Lε



Questão 11

Correto

Atingiu 1,00 de 1,00

Nas linhas 54 à 59 faltam os comandos necessários para acrescentar alguns componentes na tela.

A alternativa correta é:

Escolha uma opção:

☐ a.

54...	Lb_Nome.gr
55...	Lb_Sexo.gr
56...	Lb_Nota1.ç
57...	Lb_Nota2.ç
58...	Lb_Nota3.ç
59...	Lb_Media.ç

☐ b.

54...	grid(Lb_Nc
55...	grid(Lb_Se
56...	grid(Lb_Nc
57...	grid(Lb_Nc
58...	grid(Lb_Nc
59...	grid(Lb_Me

☐ c.

54...	Et_Nome.gr
55...	Et_Sexo.gr
56...	Et_Nota1.ç
57...	Et_Nota2.ç
58...	Et_Nota3.ç
59...	Et_Media.ç

☐ d.

54...	Lb_Nome.gr
55...	Lb_Sexo.gr
56...	Lb_Nota1.ç
57...	Lb_Nota2.ç
58...	Lb_Nota3.ç
59...	Lb_Media.ç

☒ e.

54...	Lb_Nome.gr
55...	Lb_Sexo.gr
56...	Lb_Nota1.ç
57...	Lb_Nota2.ç
58...	Lb_Nota3.ç
59...	Lb_Media.ç



Questão 12

Correto

Atingiu 1,00 de 1,00

No programa apresentado anteriormente, complete o código que processa o evento associado ao botão Bt_Calc (Linhas 19 a 24):

Escolha uma ou mais opções:

Escolha uma ou mais:

☐ a.

19...	n1=self.__F
20...	n2=self.__F
21...	n3=self.__F
22...	total=(n1+r
23...	self.__Et_h
24...	self.__Et_h

☒ b.

19...	n1=float(se
20...	n2=float(se
21...	n3=float(se
22...	total=(n1+r
23...	self.__Et_h
24...	self.__Et_h

☐ c.

19...	n1=float(se
20...	n1=float(se
21...	n1=float(se
22...	total=(n1+r
23...	self.__Et_h
24...	self.__Et_h

☐ d.

19...	n1=float(Et
20...	n2=float(Et
21...	n3=float(Et
22...	total=(n1+r
23...	self.__Et_h
24...	self.__Et_h

☐ e.

19...	n1=float(se
20...	n2=float(se
21...	n3=float(se
22...	total=(n1+r
23...	Et_Media.de
24...	Et_Media.ir



Questão 13

Correto

Atingiu 1,00 de 1,00

A sequência correta para ler e apresentar uma imagem em um Aplicativo Python é dado por:

Escolha uma ou mais:

- ☐ a.
- **Passo 1:** Importar a bibliot
from PIL import ImageTk, I
 - **Passo 2:** Ler a imagem cor
com o nome do arquivo
 - **Passo 3:** Transformar a im
photo = Transforma(Image
 - **Passo 4:** Colar a foto sobre
Label(self, image=photo)

- ☐ b.
- **Passo 1:** Importar a bibliot
from PIL import ImagemTk
 - **Passo 2:** Ler a imagem cor
com o nome do arquivo
 - **Passo 3:** Transformar a im
photo = ImageTk.PhotoIma
 - **Passo 4:** Colar a foto sobre
Table(self, imagem=photo)

- ☐ c.
- **Passo 1:** Importar a bibliot
from PIL import ImageTk, I
 - **Passo 2:** Ler a imagem cor
com o nome do arquivo
 - **Passo 3:** Transformar a im
photo = TextTk.PhotoImag
 - **Passo 4:** Colar a foto sobre
Text(self, image=photo)

- ☐ d.
- **Passo 1:** Importar a bibliot
from PIL import ImagemTk
 - **Passo 2:** Ler a imagem cor
com o nome do arquivo
 - **Passo 3:** Transformar a im
photo = ImageTkinter.Foto
 - **Passo 4:** Colar a foto sobre
Button(self, image=photo)

- ☒ e.
- **Passo 1:** Importar a bibliot
from PIL import ImageTk, I
 - **Passo 2:** Ler a imagem cor
com o nome do arquivo
 - **Passo 3:** Transformar a im
photo = ImageTk.PhotoIma
 - **Passo 4:** Colar a foto sobre
Label(self, image=photo)



Questão 14

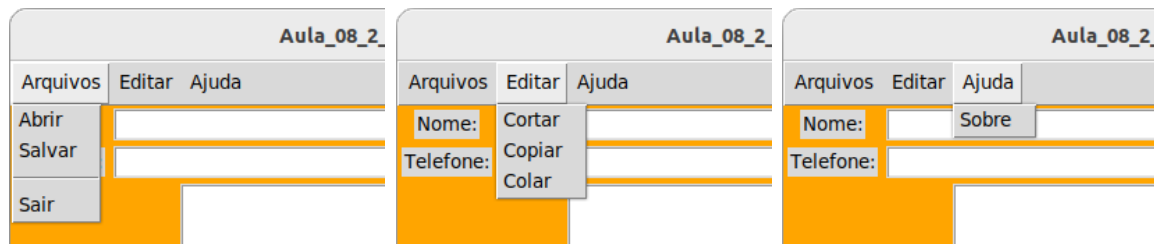
Correto

Atingiu 1,00 de 1,00

Dado o código de alocação:

avo_bar = Menu(self)

Qual a sequência correta para adicionar todos os itens de menu na janela a seguir?



Escolha uma ou mais:

- ☐ a.
- ```

pai_arquivos = MenuLinea
pai_arquivos.add_command
pai_arquivos.add_command
pai_arquivos.add_separat
pai_arquivos.add_command
avo_bar.add_cascade (Text

pai_editar = MenuLinear (
pai_editar.add_command(T
pai_editar.add_command(T
pai_editar.add_command(T
avo_bar.add_cascade (Text

pai_ajuda = MenuLinear(a
pai_ajuda.add_command(Te
avo_bar.add_cascade (Text

```
- ☒ b.
- ```

pai_arquivos = Menu(avo_
pai_arquivos.add_command
pai_arquivos.add_command
pai_arquivos.add_separat
pai_arquivos.add_command
avo_bar.add_cascade (labe

pai_editar = Menu(avo_ba
pai_editar.add_command(l
pai_editar.add_command(l
pai_editar.add_command(l
avo_bar.add_cascade (labe

pai_ajuda = Menu(avo_bar
pai_ajuda.add_command(la
avo_bar.add_cascade (labe

```



☐ c.

```
avo_arquivos = Menu(avo_
avo_arquivos.add_command
avo_arquivos.add_command
avo_arquivos.add_separat
avo_arquivos.add_command
pai_bar.add_cascade(labe

avo_editar = Menu(avo_ba
avo_editar.add_command(1
avo_editar.add_command(1
avo_editar.add_command(1
pai_bar.add_cascade(labe

avo_ajuda = Menu(avo_bar
avo_ajuda.add_command(1a
pai_bar.add_cascade(labe
```

☒ d.

```
pai_arquivos = Menu(avo_
pai_arquivos.add_command
pai_arquivos.add_command
pai_arquivos.add_separat
pai_arquivos.add_command
avo_bar.add_cascade(menu

pai_editar = Menu(avo_ba
pai_editar.add_command(1
pai_editar.add_command(1
pai_editar.add_command(1
avo_bar.add_cascade(labe

pai_ajuda = Menu(avo_bar
pai_ajuda.add_command(1a
avo_bar.add_cascade(labe
```

☐ e.

```
pai_arquivos = Menu(avo_
pai_arquivos.adicionar_c
pai_arquivos.adicionar_c
pai_arquivos.adicionar_s
pai_arquivos.adicionar_c
avo_bar.adicionar_cascad

pai_editar = Menu(avo_ba
pai_editar.adicionar_com
pai_editar.adicionar_com
pai_editar.adicionar_com
avo_bar.adicionar_cascad

pai_ajuda = Menu(avo_bar
pai_ajuda.adicionar_comm
avo_bar.adicionar_cascad
```



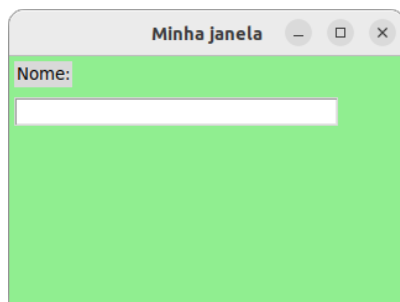
Questão 15

Correto

Atingiu 2,00 de 2,00

Complete o código abaixo:

(Aula_07_4a_App_Toplevel_OOP.py)



Arraste quais são as alternativas corretas.

```

Janela_Top.py

class Janela_Top(  ✓ ):
    def  ✓ (self, Master, Str="Janela", px=0, py=0,
        dx=640, dy=480, cor= ✓ ):
        super().__init__(Master)
         ✓ .title(Str)
        super(). ✓ ("%dx%d+%d+%d" % (dx, dy, px, py))
        super(). ✓ (bg=cor)
        super(). ✓ ()
        self. ✓ ()

    def action_exit(self):
        print("Destruindo a janela 2...")
        self. ✓

    def initialize( ✓ ):
        Lbl=Label(self, text="Nome")
        Et1=Entry(self,  ✓ )
        Lbl.grid(row=0, column=0, sticky=W, padx=2,  ✓ )
        Et1.grid(row=1, column=0, sticky=W, padx=4,  ✓ )
        self.protocol("WM_DELETE_WINDOW",  ✓ )

```

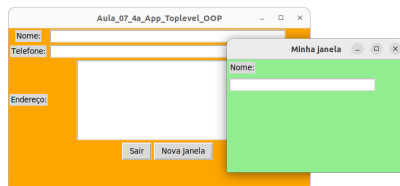




Questão 16

Correto

Atingiu 2,00 de 2,00

Complete o código abaixo:**O programa é continuação do programa apresentado anteriormente.****Arraste quais são as alternativas corretas.**`Janela.py`

```

from Janela_Top import Janela_Top

class Janela(  ✓ ):
    def __init__(self, Str="Janela",  ✓ , cor="lightgray"):
        super().  ✓
        super().title( ✓ )
        super().geometry("%dx%d+%d+%d" % (dx, dy, px, py))
        super().configure(bg=cor)
        self.  ✓ ()

    def action_new_window(self):
        Jan2 = Janela_Top(self, "Minha janela",  ✓ )

    def action_exit(self):
        self.destroy()
        sys.exit(0)

    def inicialize(self):
        Lb1=Label(self, text="Nome:")
        Lb2= ✓
        Lb3=Label(self, text="Endereço:")

        Et1=Entry(self, width=52)
        Et2= ✓

        Txt1= ✓

        Bt1=Button(self, text='Sair', command=self.action_exit)
        Bt2=Button(self, text='Nova Janela', command= ✓ )

        Lb1.grid(row=0, column=0, padx=2, pady=2)
        Lb2.grid(row=1, column=0, padx=2, pady=2)
        Lb3. ✓
        Et1.grid(row=0, column=1, columnspan=2, padx=2, pady=2)
        Et2.grid(row=1, column=1, columnspan=2, padx=2, pady=2)
        Txt1.grid(row=3, column=1, columnspan=2, padx=2, pady=2)
        Bt1.grid(row=4, column=1, sticky=E, padx=2, pady=2)
         ✓ (row=4, column=2, sticky=W, padx=2, pady=2)
        self.protocol("WM_DELETE_WINDOW",  ✓ )

```





Questão 17

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo:

Aula_09_4_App_TkTable			
Cod.	Nome	Idade	Ingresso
01	Antonio	18	2019
02	Beatriz	19	2018
03	Carlos	20	2017
04	Darlan	17	2020
05	Eduardo	21	2016
06	Fernando	20	2015
07	Guilherme	19	2017
08	Humberto	18	2018
09	Isabel	17	2021
Print Exit			

Programa: Aula_09_4_App_TkTable

Dados.py

```
001... #####
002...
003... Dados = [
004...     ['01', 'Antonio', '18', '2019'],
005...     ['02', 'Beatriz', '19', '2018'],
006...     ['03', 'Carlos', '20', '2017'],
007...     ['04', 'Darlan', '17', '2020'],
008...     ['05', 'Eduardo', '21', '2016'],
009...     ['06', 'Fernando', '20', '2015'],
010...     ['07', 'Guilherme', '19', '2017'],
011...     ['08', 'Humberto', '18', '2018'],
012...     ['09', 'Isabel', '17', '2021'],
013...     ['10', 'Joao', '21', '2016'],
014...     ['11', 'Karen', '23', '2015'],
015...     ['12', 'Luciana', '19', '2018'],
016...     ['13', 'Maria', '18', '2020'],
017...     ['14', 'Nicole', '19', '2019']
018... ]
019...
020... #####
```

Janela.py

```
021... import sys
022... from tkinter import *
023... from tkinter import messagebox
024... from Dados import Dados
025...
026... #####
027...
028... class Janela(Tk):
029...     Cnv1 = None
030...     inter = None
031...     inter_id = None
032...     Matriz = None
033...
034...     def __init__(self, Str="Janela", px=0, py=0, dx=640, dy=480, cor="lighthgray"):
035...         super().__init__()
036...         super().title(Str)
037...         super().geometry("%dx%d+%d+%d" % (dx, dy, px, py))
038...         super().configure(bg=cor)
039...         self.inicialize()
040...
041...     def action_exit(self):
042...         print("Destruindo a janela...")
043...         self.destroy()
044...         sys.exit(0)
045...
046...     def action_print(self):
047...         #####
048...         ##                                     ##
049...         ##             Complete o código             ##
050...         ##             do evento                       ##
051...         ##                                     ##
052...         #####
053...
054...     def calculo_percentual(self):
055...         total=0
056...         Lb_height = self.Lb_cod.wininfo_height() + 2 + 2
057...         print("Label height: %f" % Lb_height)
058...         total += Lb_height
059...         for row in self.Matriz:
060...             if (len(row)>0):
061...                 cell = row[0]
062...                 Et_height = cell.wininfo_height()+2+2
063...                 total+=Et_height
064...
065...         Cnv_height = self.Cnv1.wininfo_height()
066...         perc = (total-Cnv_height) / total
067...
068...         print("Canvas height: %f" % Cnv_height)
069...         print("Total: %f" % total)
070...         print("Percentual: %f %" % perc)
071...         return(perc)
072...
073...     # track changes to the canvas and frame width and sync them,
074...     # also updating the scrollbar
075...     def _configure_interior(self, event):
076...         # update the scrollbars to match the size of the inner frame
077...         print("_configure_interior: %s" % event)
078...         posy = -event.y / event.height
079...         print("moveTo: %f %" % posy)
080...
081...         perc = self.calculo_percentual()
082...
083...         if (posy < 0.0):
084...             self.Cnv1.yview_moveto(0.0)
```



```

085...         if (posy > perc):
086...             self.Cnv1.yview_moveto(perc)
087...
088...         size = (self.inter.wininfo_reqwidth(), self.inter.wininfo_reqheight())
089...         self.Cnv1.config(scrollregion="0 0 %s %s" % size)
090...         if self.inter.wininfo_reqwidth() != self.Cnv1.wininfo_width():
091...             # update the canvas's width to fit the inner frame
092...             self.Cnv1.config(width=self.inter.wininfo_reqwidth())
093...
094...     def _configure_canvas(self, event):
095...         if self.inter.wininfo_reqwidth() != self.Cnv1.wininfo_width():
096...             # update the inner frame's width to fill the canvas
097...             self.Cnv1.itemconfigure(self.inter_id, width=self.Cnv1.wininfo_width())
098...
099...     def initialize(self):
100...         self.Frm1 = Frame(self, width=100, height=80, bd=4)
101...         self.Frm1.configure(bg='cyan')
102...
103...         self.Cnv1 = Canvas(self.Frm1, width=300, height=260, confine=False,
104...                             yscrollincrement=10, scrollregion=(0, 0, 300, 260))
105...         self.Cnv1.configure(bg='yellow', scrollregion="0 0 200 160")
106...         self.Cnv1.xview_moveto(0)
107...         self.Cnv1.yview_moveto(0)
108...
109...         self.inter = Frame(self.Cnv1)
110...         self.inter_id = self.Cnv1.create_window(0, 0, window=self.inter, anchor=NW);
111...
112...         self.inter.bind('<Configure>', self._configure_interior)
113...
114...         self.Cnv1.bind('<Configure>', self._configure_canvas)
115...
116...         Sb1 = Scrollbar(self.Frm1, orient="vertical")
117...
118...         self.Frm2 = Frame(self, width=100, bd=4)
119...         self.Frm2.configure(bg='red')
120...
121...         self.Lb_cod = Label(self.inter, text="Cod.")
122...         #####
123...         ##      Alocar os demais componentes      ##
124...         #####
125...
126...         self.Lb_cod.configure(bg='yellow', anchor=W)
127...         Lb_nome.configure(bg='yellow')
128...         Lb_idade.configure(bg='yellow')
129...         Lb_ingresso.configure(bg='yellow')
130...
131...         self.Lb_cod.grid(row=0, column=0, sticky=W, padx=2, pady=2)
132...         Lb_nome.grid(row=0, column=1, sticky=W, padx=2, pady=2)
133...         Lb_idade.grid(row=0, column=2, sticky=W, padx=2, pady=2)
134...         Lb_ingresso.grid(row=0, column=3, sticky=W, padx=2, pady=2)
135...
136...         #####
137...         ##                                     ##
138...         ##                                     ##
139...         ##                                     ##
140...         ##                                     ##
141...         ##      Preencha os campos da tabela      ##
142...         ##                                     ##
143...         ##                                     ##
144...         ##                                     ##
145...         ##                                     ##
146...         #####
147...
148...         Sb1.set(0, 10)

```



149...	Sb1.config(command=self.Cnv1.yview)
150...	self.Cnv1.config(yscrollcommand=Sb1.set)
151...	
152...	Bt_print = Button(self.Frm2, text='Print', anchor='e', command=self.action_print)
153...	Bt_exit = Button(self.Frm2, text='Exit', anchor='w', command=self.action_exit)
154...	
155...	self.Frm1.grid(row=0, column=0)
156...	self.Frm2.grid(row=1, column=0)
157...	
158...	self.Cnv1.grid(row=0, column=0)
159...	Sb1.grid(row=0, column=ncol, rowspan=nlin, sticky=NS)
160...	Bt_print.grid(row=0, column=0, sticky=E, padx=8, pady=2)
161...	Bt_exit.grid(row=0, column=1, sticky=W, padx=8, pady=2)
162...	
163...	self.grid_anchor("center")
164...	
165...	self.protocol("WM_DELETE_WINDOW", self.action_exit)
166...	
167...	#####

Aula_09_4_App_TkTable.py	
168...	from Janela import Janela
169...	
170...	#####
171...	
172...	Jan1=Janela("Aula_09_4_App_TkTable", 400, 200, 720, 340, "orange")
173...	Jan1.mainloop()
174...	
175...	#####

Quais as alternativas corretas em relação ao código das linhas 047 à 052 para imprimir os dados apresentados na figura em uma janela Messagebox ?



Escolha uma ou mais:

- ☐ a.

047...	Texto = ""
048...	for row in
049...	for ce
050...	Te
051...	Texto
052...	messagebox
- ☐ b.

047...	Texto = ""
048...	for row in
049...	for ce
050...	Te
051...	Texto
052...	messagebox
- ☒ c.

047...	Texto = ""
048...	for rw in
049...	for cl
050...	Te
051...	Texto
052...	messagebox

☐ d.

047...	Texto = ""
048...	for i in r
049...	for j
050...	Te
051...	Texto
052...	messagebox

☒ e.

047...	Nome = ""
048...	for rw in
049...	for cl
050...	No
051...	Nome +
052...	messagebox



Questão 18

Parcialmente correto

Atingiu 0,50 de 1,00

Considere o código fornecido anteriormente:

Quais as alternativas corretas em relação ao código das linhas 136 à 146 para preencher os campos da tabela com os dados fornecidos no arquivo Dados.py ?

Escolha uma ou mais:

☒ a.

136...	nlin = len
137...	ncol = len
138...	self.Matri
139...	for i in r
140...	cols =
141...	for j
142...	Et
143...	Et
144...	Et
145...	cc
146...	self.M

☐ b.

136...	nlin = len
137...	ncol = len
138...	self.Matri
139...	for i in r
140...	cols =
141...	for j
142...	Et
143...	Et
144...	Et
145...	cc
146...	self.M

☐ c.

136...	n = len(Da
137...	m = len(Da
138...	self.Matri
139...	for i in r
140...	cols =
141...	for j
142...	Et
143...	Et
144...	Et
145...	cc
146...	self.M

☐ d.

136...	nlin = len
137...	ncol = len
138...	self.Matri
139...	for i in r
140...	cols =
141...	for j
142...	Et
143...	Et
144...	Et
145...	cc
146...	self.M



☐ e.

136...	nlin = len
137...	self.Matri
138...	for i in r
139...	cols =
140...	for j
141...	Et
142...	Et
143...	cc
144...	self.M
145...	
146...	

Questão 19

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo:

```
Mat=[[1.0, 2.0, 3.0],
      [4.0, 5.0, 6.0],
      [7.0, 8.0, 9.0]]

valor=0
for i in range(0,3,1):
    for j in range(0,3,1):
        if (i >= j):
            valor=valor+Mat[i][j]
    print("%4.1f" % valor)
```

**Qual a sequência de valores impressos?**

Escolha uma opção:

- ☐ a.

2.0
11.0
33.0
- ☐ b.

3.0
12.0
31.0
- ☒ c.

1.0
10.0
34.0
- ☐ d.

34.0
10.0
4.0
- ☐ e.

3.0
15.0
30.0

Questão 20

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo:

```
Mat=[[1.0, 2.0, 3.0],  
      [4.0, 5.0, 6.0],  
      [7.0, 8.0, 9.0]]  
  
valor=1  
for i in range(0,3,1):  
    for j in range(0,3,1):  
        if (i <= j):  
            valor=valor*Mat[i][j]  
    print("%6.1f" % valor)
```

Qual a sequência de valores impressos?

Escolha uma opção:

- ☐ a.

8.0
160.0
1820.0
- ☒ b.

6.0
180.0
1620.0
- ☐ c.

160.0
80.0
620.0
- ☐ d.

8.0
142.0
1420.0
- ☐ e.

1620.0
180.0
20.0



Questão 21

Correto

Atingiu 1,00 de 1,00

Suponha que um professor deseja saber quais são os alunos que estão cursando as disciplinas A e B.

O código para resolver esse problema tem as seguintes características:

- Os nomes dos alunos matriculados estão armazenados em duas listas A e B;
- A lista de alunos na disciplina A é um array de 30 posições com o nome dos matriculados;
- A lista de alunos na disciplina B é um array de 40 posições com o nome dos matriculados;
- Os arrays já foram previamente inicializados e seus valores foram digitados pelo usuário.

Parte do código é fornecido, cabe a você completar a parte faltante:

```
for i in range(0,30,1):  
    for j in range(0,40,1):  
        ## Escolha qual o código correto a ser inserido aqui.
```

Qual a resposta correta?



Escolha uma opção:

- ☐ a.

```
if (A[i]!=B[j]):  
    print("%s " % A[i])
```
- ☐ b.

```
if (A[i].compare.(B[j]))  
    print("%s " % A[i])
```
- ☒ c.

```
if (A[i]==B[j]):  
    print("%s " % A[i])
```
- ☐ d.

```
if (A[j]==B[i]):  
    print("%s " % A[j])
```
- ☐ e.

```
if (A[i]==B[j]):  
    print("%s " % A[j])
```
- ☐ f.

```
if (A[i]==B[i]):  
    print("%s " % A[i])
```

Questão 22

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo:

```
01... Vect_Str=[]
02...
03... try:
04...     with open("./texto.txt", "r") as file:
05...         Data=file.readlines()
06...
07...         for line in Data:
08...             Vect_Str.append("%s" % line.replace("\n", ""))
09... except FileNotFoundError as error:
10...     print("Erro1: Arquivo não encontrado: %s" % error)
11... except IOError as error:
12...     print("Erro2: Erro de entrada e saída: %s" % error)
13... except Exception as ex:
14...     print("Erro3: Erro inesperado: %s" % ex)
15...
16... for i in range(0, len(Vect_Str),1):
17...     print("Vect_Str[%d] --> %s" % (i, Vect_Str[i])) #O comando print já põe um \n
```

Quais as afirmações estão corretas?

Escolha uma ou mais:

- ☐ a. O código serve para carregar um arquivo binário para memória
- ☐ b. O código serve para carregar um vetor de números reais para memória
- ☒ c. O código serve para carregar um arquivo de texto para memória ✓
- ☐ d. A linha 05 abre o arquivo para leitura
- ☐ e. A linha 05 abre o arquivo para escrita



Questão 23

Correto

Atingiu 1,00 de 1,00

Em relação ao código da questão anterior quais as afirmações estão corretas?

Escolha uma ou mais:

- ☒ a. A linha 05 lê todas as linhas do arquivo "texto.txt" ✓
- ☒ b. A linha 08 transfere todas as linhas lidas para o vetor Vect_Str[] ✓
- ☐ c. Na linha 16 o comando len(Vect_Str) aloca espaço na memória para armazenar os dados
- ☐ d. A linha 17 apresenta um erro de compilação
- ☒ e. O comando da linha 17 serve para imprimir cada linha do vetor Vect_Str juntamente com seu índice i ✓



Questão 24

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo:

```
disciplinas = {  
    'DADII': 'Anderson',  
    'DADI': 'Kobs',  
    'BAN': 'Bilecki',  
    'PROGII': 'Tamanini',  
    'PROGIII': 'Carlos'  
}  
  
for chave in disciplinas:  
    print("Professor(a) de %s é: %s" % (chave, disciplinas[chave]))
```

Qual a sequência de valores impressos?

Escolha uma opção:

- ☐ a. Professor(a) de DADI é:
Professor(a) de BAN é: B
Professor(a) de PROGIII
Professor(a) de DADII é:
Professor(a) de PROGII é
- ☐ b. Professor de DADI é: Kob
Professor de BAN é: Bile
Professor de PROGII é: T
Professor de PROGIII é:
Professor de DADII é: An
- ☐ c. Professor(a) de DADI é:
Professor(a) de BAN é: B
Professor(a) de PROGII é
Professor(a) de PROGIII
Professor(a) de DADII é:
- ☒ d. Professor(a) de DADII é:
Professor(a) de DADI é:
Professor(a) de BAN é: B
Professor(a) de PROGII é
Professor(a) de PROGIII
- ☐ e. Professor de DADII é: An
Professor de DADI é: Kob
Professor de BAN é: Bile
Professor de PROGII é: T
Professor de PROGIII é:



Questão 25

Parcialmente correto

Atingiu 1,40 de 2,00

Assista o vídeo disponível no link abaixo:



Título: 018 - Python tkinter - APLICAÇÃO PARA DESENHO DE INTERFACES
Autor: João Ribeiro
Link: [\(Acessar aqui\)](#)

Agora responda a seguinte questão:

Quais os principais passos descritos no vídeo para utilizar o aplicativo Pygubu para gerar uma interface em Python Tkinter?

Mantenha a mesma ordem apresentada no vídeo!

Ative a legenda no rodapé do video player.



Passo 01:	✓	Localize na internet a página do fabricante do Pygubu
Passo 02:	✓	Digite: "pip install pygubu" para instalar o software
Passo 03:	✓	Chame o aplicativo: "pygubu-designer" para iniciar o software instalado
Passo 04:	✓	Crie um container Toplevel
Passo 05:	✗	Adicione um Label ao container Toplevel
Passo 06:	✗	Altere a geometria do container Toplevel para 500x300
Passo 07:	✗	Acesse o menu Preview para visualizar a interface que está sendo construída
Passo 08:	✓	Adicione um Button ao container Toplevel
Passo 09:	✓	Salve o aplicativo gerado com a extensão .ui
Passo 10:	✓	Crie um aplicativo em Python capaz de ler e apresentar a interface .ui



Questão 26

Correto

Atingiu 1,00 de 1,00

Consulte o link abaixo:

**Título:** pygubu-designer 0.32**Site:** PyPI - Python Package Index**Link:** [\(Acessar aqui\)](#)

Agora responda quais as alternativas estão corretas em relação ao Pygubu

- ☐ a. É um software para auxiliar no desenvolvimento de aplicativos Android
- ☒ b. É um software para auxiliar no desenvolvimento de interface gráfica com a biblioteca Tkinter ✓
- ☐ c. É um software para auxiliar no desenvolvimento de aplicativos em PyQt5
- ☐ d. É um software para auxiliar no diagnóstico de problemas no computador
- ☐ e. É um software para auxiliar no desenvolvimento de interface gráfica com biblioteca matplotlib



Questão 27

Correto

Atingiu 1,00 de 1,00

Consulte o link abaixo:

**Título:** pygubu-designer 0.32**Site:** PyPI - Python Package Index**Link:** [\(Acessar aqui\)](#)

Agora responda quais as alternativas estão corretas em relação ao processo de desenvolvimento utilizando o Pygubu

- ☐ a. Quando você cria uma interface, a especificação da mesma é salva em um arquivo Python, com extensão ".py"
- ☐ b. Quando você cria uma interface, a especificação da mesma é salva em um arquivo xml, com extensão ".xml"
- ☒ c. Quando você cria uma interface, a especificação da mesma é salva em um arquivo xml, com extensão ".ui" ✓
- ☐ d. Quando você cria uma interface, a especificação da mesma é salva em um arquivo txt, com extensão ".txt"
- ☐ e. Quando você cria uma interface, a especificação da mesma roda direto sobre o interpretador do Python



Questão 28

Correto

Atingiu 2,00 de 2,00

Considere que foi montada uma interface com o Pygubu e gravada com o nome: 'HelloWorld_App.ui'. Essa interface é representada pela Figura Q-04:



Figura Q_04: Aplicativo HelloWorld_App

Em seguida foi montado um programa para ler e apresentar essa interface.

Código do programa: HelloWorld_App.py

```
from tkinter import *
import pygubu

class HelloWorld_App :

    def __init__(self):

        #1: Create a builder
        self.builder = pygubu.Builder()

        #2: Load an ui file
        self.builder.add_from_file('HelloWorld_App.ui')

        #3: Create the mainwindow
        self.Jan1 = self.builder.get_object('JanelaPrincipal')

    def run(self):
        self.Jan1.mainloop()

if __name__ == '__main__':
    app = HelloWorld_App()
    app.run()
```



Arraste sobre o código as respostas correspondentes para construir esse aplicativo.

pygubu.compiler



Questão 29

Correto

Atingiu 2,00 de 2,00

Considere que foi montada uma interface com o Pygubu e gravada com o nome: 'Aluno_App.ui'. Essa interface é representada pela Figura Q-05:

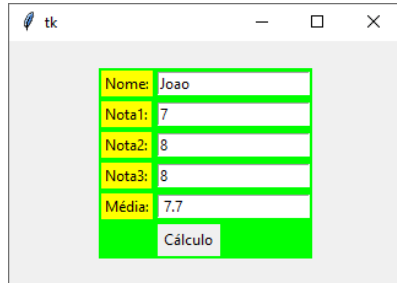


Figura Q_05: Aplicativo Aluno_App

Em seguida foi montado um programa para ler e apresentar essa interface.

Código do programa: Aluno_App.py



```

from tkinter import *
import pygubu

class Aluno_App:

    def __init__(self):

        #1: Create a builder ✓
        self.builder = pygubu.Builder()

        #2: Load an ui file ✓
        self.builder.add_from_file('Aluno_App.ui')

        #3: obtém os objetos da UI
        self.Jan1 = self.builder.get_object('JanelaPrincipal' ✓ )
        self. Et_Nt1 ✓ = self.builder. get_object ✓ ('Et_Nt1')
        self. Et_Nt2 ✓ = self.builder.get_object('Et_Nt2')
        self.Et_Nt3 = self.builder.get_object('Et_Nt3' ✓ )
        self.Et_Media = self.builder ✓ .get_object('Et_Media')

        self. builder ✓ .connect_callbacks(self)

    def run(self):
        self.Jan1. mainloop() ✓

    def action_calcular(self):
        print('Entrei no evento')
        n1=float(self.Et_Nt1. get() ✓ )
        n2=float(self. Et_Nt2.get() ✓ )
        n3= float ✓ (self.Et_Nt3.get())
        total= (n1+n2+n3) ✓ /3
        self.Et_Media.delete(0, END)
        self.Et_Media. insert ✓ (END, "%4.1f" % total)

if __name__ == '__main__' ✓ ':
    app = Aluno_App()
    app. run() ✓

```



Arraste sobre o código as respostas
correspondentes para construir esse
aplicativo.

delete

