



#ProgramáTuFuturo





Municipalidad de
Tres de Febrero



Programá
tu futuro



PYTHON INTERMEDIO

¡Les damos la bienvenida!





Municipalidad de
Tres de Febrero



Programá
tu futuro



CRUD I

CLASE 11



PARTE GRAFICA



Programá
tu futuro



Municipalidad de
Tres de Febrero

Definimos el archivo principal

Creamos un archivo principal.py

```
import tkinter as tk

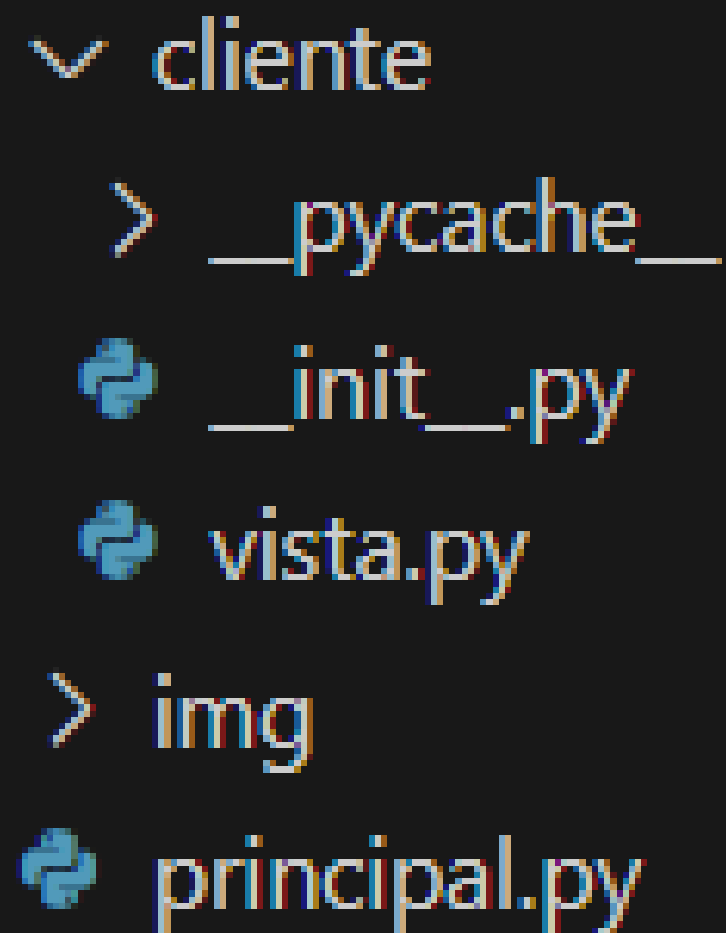
def main():
    ventana = tk.Tk()
    ventana.title('Listado Peliculas')
    ventana.iconbitmap('img/videocamara.ico')
    ventana.resizable(0,0)

    ventana.mainloop()

if __name__ == '__main__':
    main()
```

Haciendo una Vista

Creamos una carpeta donde ubicaremos la vista del usuario



```
✓ cliente
  > __pycache__
  🔄 __init__.py
  🔄 vista.py
  > img
  🔄 principal.py
```

```
import tkinter as tk
```

```
class Frame(tk.Frame):
    def __init__(self, root = None):
        super().__init__(root,width=480,height=320)
        self.root = root
        self.pack()
        self.config(bg='green')
```

Creamos una clase Frame, que herede de la clase Frame de Tkinter

Instanciamos la vista

instanciamos la clase Frame que acabamos de crear en nuestro archivo principal

Para esto importaremos desde nuestro archivo vista

```
import tkinter as tk  
from cliente.vista import Frame
```

```
def main():  
    ventana = tk.Tk()  
    ventana.title('Listado Peliculas')  
    ventana.iconbitmap('img/videocamara.ico')  
    ventana.resizable(0,0)
```

```
app = Frame(root = ventana)
```

```
ventana.mainloop()
```

```
if __name__ == '__main__':  
    main()
```


agregando un menu

vamos a crear una barra de menu

Por el momento no habra funcionalidades, mas que la de Salir, luego de crearlo lo importamos en nuestro main y ejecutamos antes de instanciar

```
from cliente.vista import Frame, barra_menu

def main():
    ventana = tk.Tk()
    ventana.title('Listado Peliculas')
    ventana.iconbitmap('img/videocamara.ico')
    ventana.resizable(0,0)

    barra_menu(ventana)
    app = Frame(root = ventana)
```

```
def barra_menu(root):
    barra = tk.Menu(root)
    root.config(menu = barra, width = 300 , height = 300)
    menu_inicio = tk.Menu(barra, tearoff=0)

    # niveles #

    #principal

    barra.add_cascade(label='Inicio', menu = menu_inicio)
    barra.add_cascade(label='Consultas', menu = menu_inicio)
    barra.add_cascade(label='Acerca de..', menu = menu_inicio)
    barra.add_cascade(label='Ayuda', menu = menu_inicio)

    #submenu
    menu_inicio.add_command(label='Conectar DB')
    menu_inicio.add_command(label='Desconectar DB')
    menu_inicio.add_command(label='Salir', command= root.destroy)
```

Identificando

Vamos a crear los labels

```
class Frame(tk.Frame):  
    def __init__(self, root = None):  
        super().__init__(root,width=480,height=320)  
        self.root = root  
        self.pack()  
  
        self.label_form()
```

crearemos un metodo dentro de nuestra clase, y luego lo ejecutaremos en el constructor del mismo, comentamos o borramos la linea que da el bg

```
def label_form(self):  
    self.label_nombre = tk.Label(self, text="Nombre: ")  
    self.label_nombre.config(font=('Arial',12,'bold'))  
    self.label_nombre.grid(row= 0, column=0,padx=10,pady=10)  
  
    self.label_nombre = tk.Label(self, text="Duración: ")  
    self.label_nombre.config(font=('Arial',12,'bold'))  
    self.label_nombre.grid(row= 1, column=0,padx=10,pady=10)  
  
    self.label_nombre = tk.Label(self, text="Genero: ")  
    self.label_nombre.config(font=('Arial',12,'bold'))  
    self.label_nombre.grid(row= 2, column=0,padx=10,pady=10)
```

para pasar la informacion

Vamos a crear los input

```
def input_form(self):
    self.entry_nombre = tk.Entry(self)
    self.entry_nombre.config(width=50, state='disabled')
    self.entry_nombre.grid(row= 0, column=1,padx=10,pady=10)

    self.entry_duracion = tk.Entry(self)
    self.entry_duracion.config(width=50, state='disabled')
    self.entry_duracion.grid(row= 1, column=1,padx=10,pady=10)

    self.entry_genero = ttk.Combobox(self, state="readonly")
    self.entry_genero.config(width=25, state='disabled')
    self.entry_genero.bind("<<ComboboxSelected>>")
    self.entry_genero.grid(row= 2, column=1,padx=10,pady=10)
```

Tanto para el nombre , como para la duración usaremos Entry, para el Gebero usaremos un combobox, que se llenara con la info en la base de datos

```
class Frame(tk.Frame):
    def __init__(self, root = None):
        super().__init__(root,width=480,height=320)
        self.root = root
        self.pack()

        self.label_form()
        self.input_form()
```

un btn salvaje ha aparecido

Vamos a crear los botones

```
def botones_principales(self):
    self.btn_alta = tk.Button(self, text='Nuevo')
    self.btn_alta.config(width= 20,font=('Arial', 12,'bold'),fg = '#FFFFFF' ,
bg='#1C500B',cursor='hand2',activebackground='#3FD83F',activeforeground='#000000')
    self.btn_alta.grid(row= 3, column=0,padx=10,pady=10)

    self.btn_modi = tk.Button(self, text='Guardar')
    self.btn_modi.config(width= 20,font=('Arial', 12,'bold'),fg = '#FFFFFF' ,
bg='#0D2A83',cursor='hand2',activebackground='#7594F5',activeforeground='#000000')
    self.btn_modi.grid(row= 3, column=1,padx=10,pady=10)

    self.btn_cance = tk.Button(self, text='Cancelar')
    self.btn_cance.config(width= 20,font=('Arial', 12,'bold'),fg = '#FFFFFF' ,
bg='#A90A0A',cursor='hand2',activebackground='#F35B5B',activeforeground='#000000')
    self.btn_cance.grid(row= 3, column=2,padx=10,pady=10)
```

Crearemos 3 botones principales

```
class Frame(tk.Frame):
    def __init__(self, root = None):
        super().__init__(root,width=480,height=320)
        self.root = root
        self.pack()

        self.label_form()
        self.input_form()
        self.botones_principales()
```


Ajustando los input

Ajustaremos los input

aagregaremos un , columnspan='2' al final del grif

```
self.entry_nombre.grid(row= 0, column=1,padx=10,pady=10, columnspan='2')
```



Listado Peliculas

Inicio Consultas Acerca de.. Ayuda

Nombre:

Duración:

Genero:

Nuevo Guardar Cancelar

Con esto ajustaremos la expansion de los input y se acomodaran mejor los botones

primera funcionalidad.. bueno la segunda

Vamos a dale vida a los botones

```
text='Nuevo', command=self.habilitar_campos)
```

```
text='Cancelar', command=self.bloquear_campos)
```

Todos los botones y campos (salvo nuevo) , deben aparecer disabled

```
def habilitar_campos(self):  
    self.entry_nombre.config(state='normal')  
    self.entry_duracion.config(state='normal')  
    self.entry_genero.config(state='normal')  
    self.btn_modi.config(state='normal')  
    self.btn_cance.config(state='normal')  
    self.btn_alta.config(state='disabled')
```

```
def bloquear_campos(self):  
    self.entry_nombre.config(state='disabled')  
    self.entry_duracion.config(state='disabled')  
    self.entry_genero.config(state='disabled')  
    self.btn_modi.config(state='disabled')  
    self.btn_cance.config(state='disabled')  
    self.btn_alta.config(state='normal')
```

¡MUCHAS GRACIAS!



Programá
tu futuro



Municipalidad de
Tres de Febrero