



Universidad Nacional Autónoma de  
México

Facultad de Ingeniería

Semestre: 2020 1

Proyecto: Pumi agenda

Asignatura: Estructura de datos y  
algoritmos I

Profesor: Marco Antonio Quintana

Alumno: pascual Hernández Daniel  
Alfonso

# OBJETIVOS

Lograr crea una agenda de fácil acceso

Uso intuitivo

Disponible sin conexión a internet

Uso rápido

Disponible en diferentes dispositivos

Sin límite de espacio

# INTRIDUCCION

Es una agenda electrónica donde podemos agregar diferente tipo de información, de manera fácil, con ayuda de la interfaz grafica logrando interacción intuitiva. La importancia en este proyecto radica que no existen muchas agendas de este tipo la gran mayoría necesita de internet o alguna cuneta para su uso.

Es aplicable alguna empresa que quiera usar su propio software unitariamente, uso personal o domestico para casos de agencia.

# DESARROLLO

Primera fase: decidí iniciar con la interfaz grafica usando la biblioteca de “Tkinder”, ya que teniendo ese esqueleto me ayudaría a saber como quería estructurar todo, decidí usar el lenguaje de programación python puesto que para la lista que quería usar me daba ciertas ventajas con otros lenguajes que suelo usar, además que gracias al profesor me oriento bastante con los cursos que nos dio.

Segunda fase: en esta parte es donde utilice una estructura vista en las clases (listas), de esta forma logre almacenar los distintos tipos de datos que podemos agregar, tarde varios días en hacer a veces ya que la inicio tenía bastantes errores por lo consecuente no copilaba el programa. Cuando logre “terminarlo” tenía unos errores ya que no tenía un funcionamiento adecuado de algunas funciones nuevamente me tarde en encontrar el ‘problema pero lo logre resolverlo.

# RESULTADOS

Algoritmo:

Crear una interfaz grafica donde se pueda pedir información y posteriormente mostrarla

Crear una lista para poder almacenar la información

Poder agregar información a dicha lista

Poder mostrar lo agregado

Finalizar hasta cerrar la ventana

Pseudocódigo

1) inicio

2) función guardar y crear lista

3) asignar variables para poder guardar la información

4) ordenamos la lista

5) limpiamos las casillas al ser guardadas y notificar que se guardo

6) fin función guardar

7) función eliminar

8) crear variable para verificar existencia de contacto

9) decimos en que parte de la lista buscar

10) notificación que se borro

11) fin función borrar

12) función consultar

13) crear lista

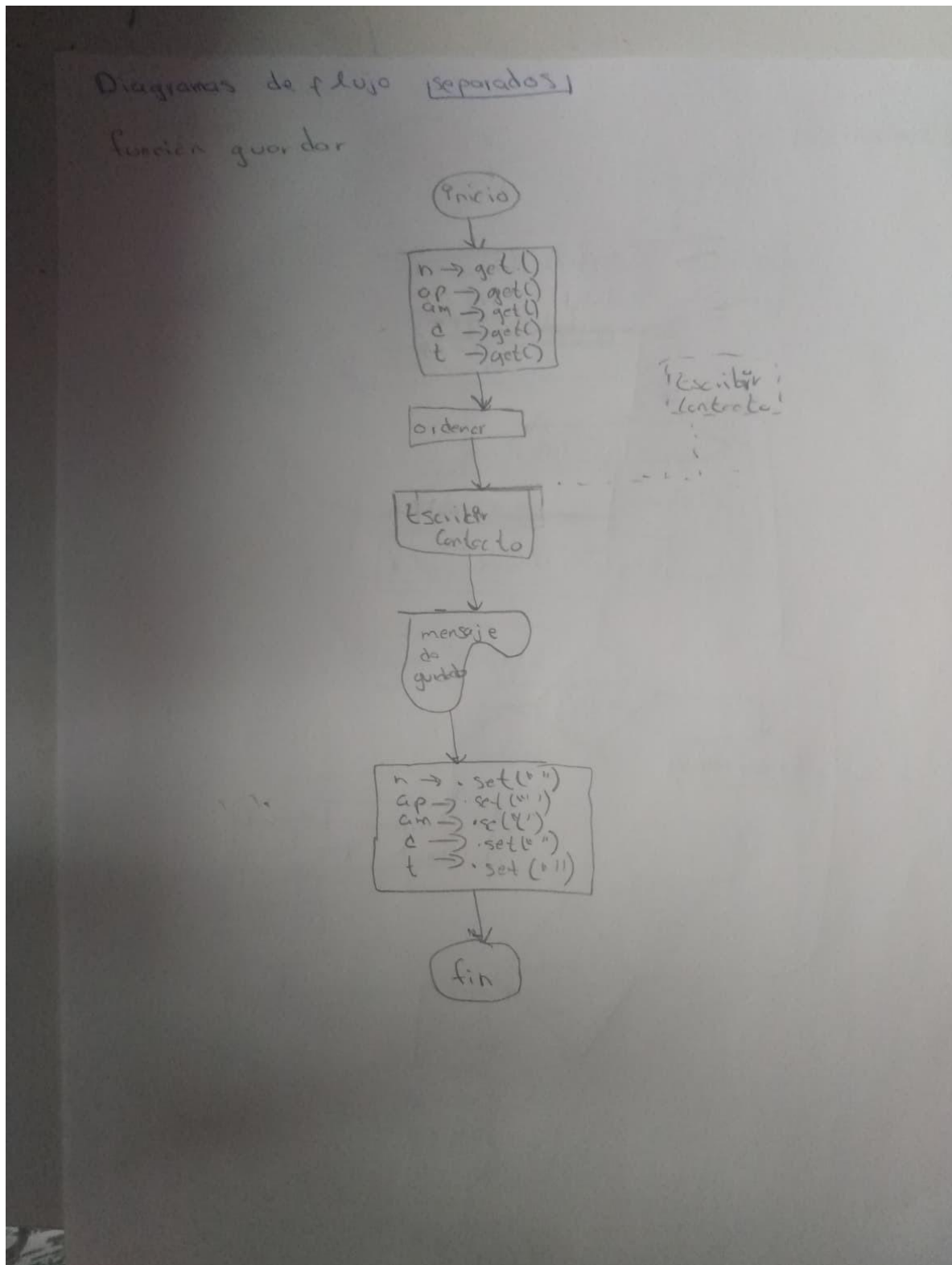
14) asignamos valores en la caja donde se verán los contactos guardados

15) fin función consultar

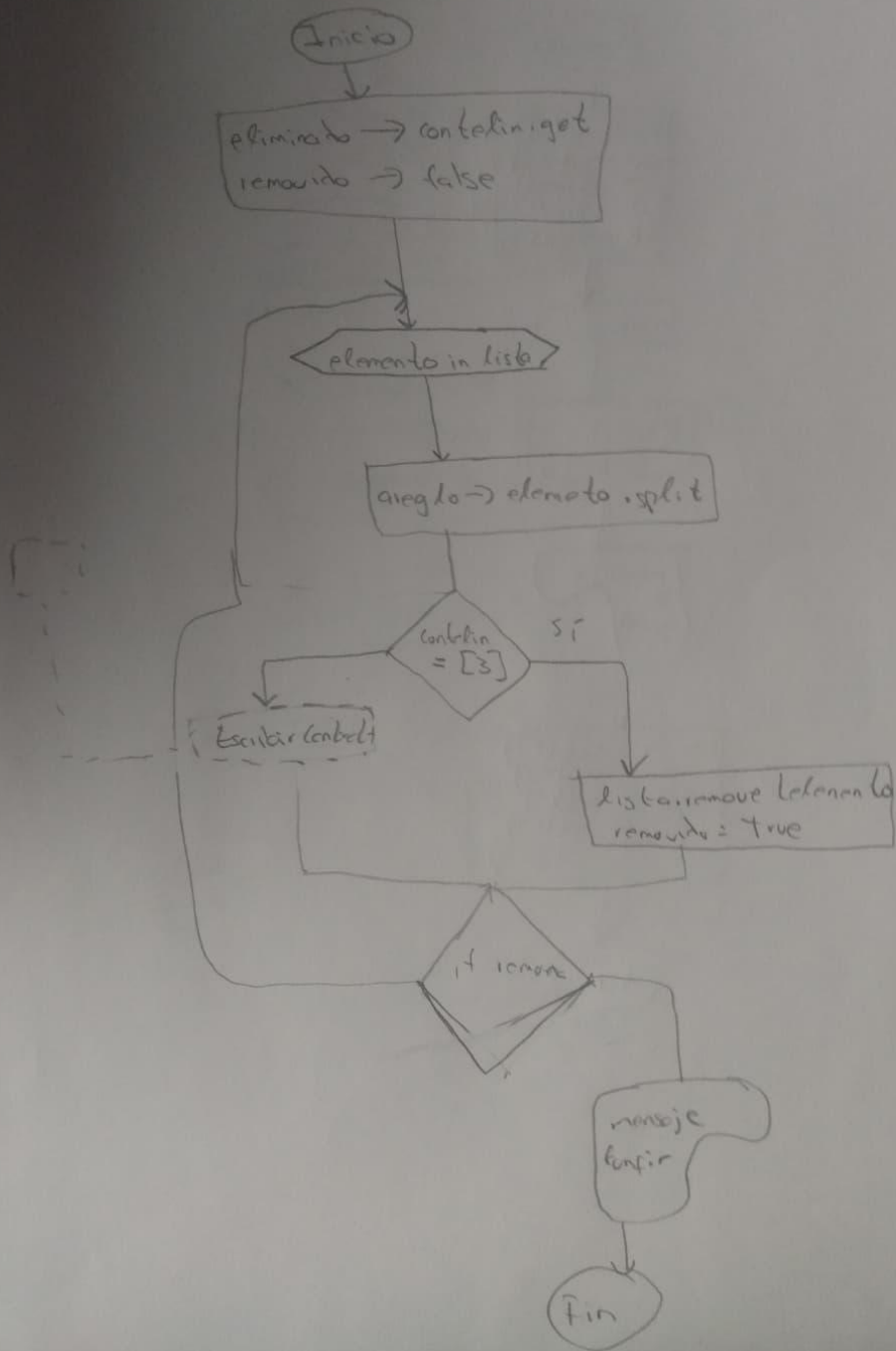
16) función cargar

- 17) creamos un archivo en block de notas
- 18) almacenamos todo lo dado
- 19) cerrar archivo y fin función cargar
- 20) función escribir contacto
- 21) vamos copiando toda la información del archivo creado
- 22) mostramos en pantalla)
- 23) fin función escribir contacto
- 24) estructura la interfaz grafica
- 25) hacer que se abra otra ventana y escoger el color
- 26) colocar los spinbox junto con su nombre
- 27) ir dando las funciones que se deben estar ejecutando
- 28) dar un mainloop para que se ejecutó constantemente
- 29) fin de la interfaz
- 30) fin

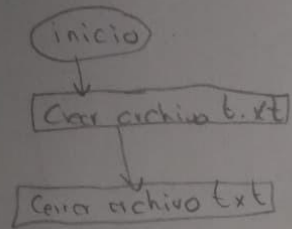
## Diagramas de flujo



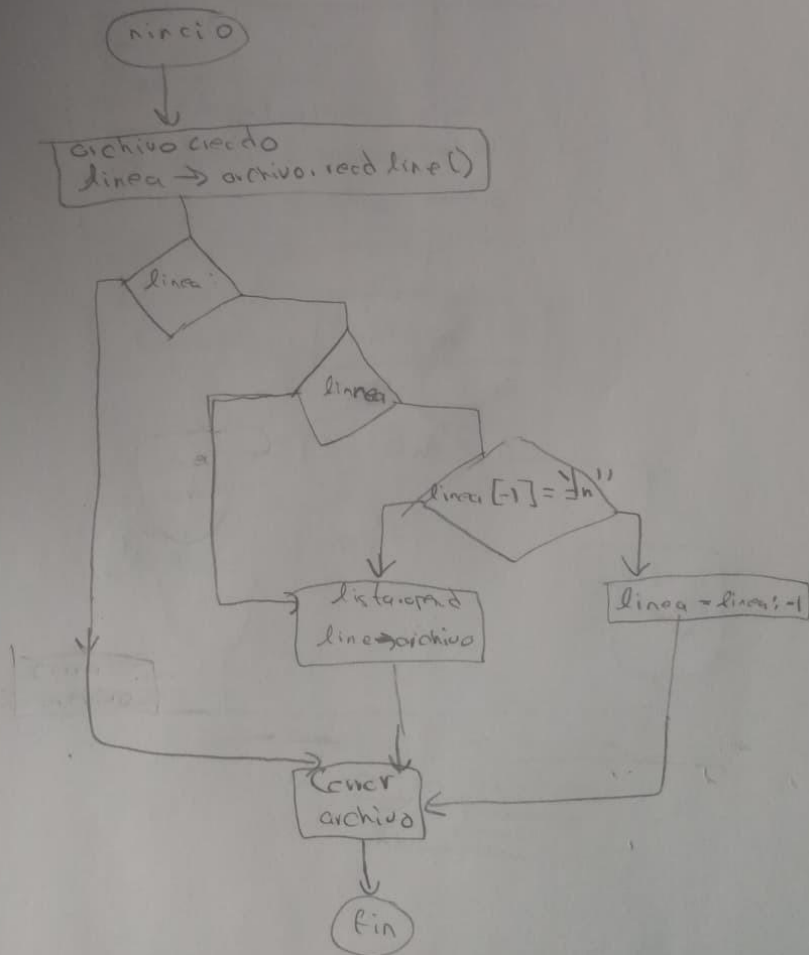
Eliminar



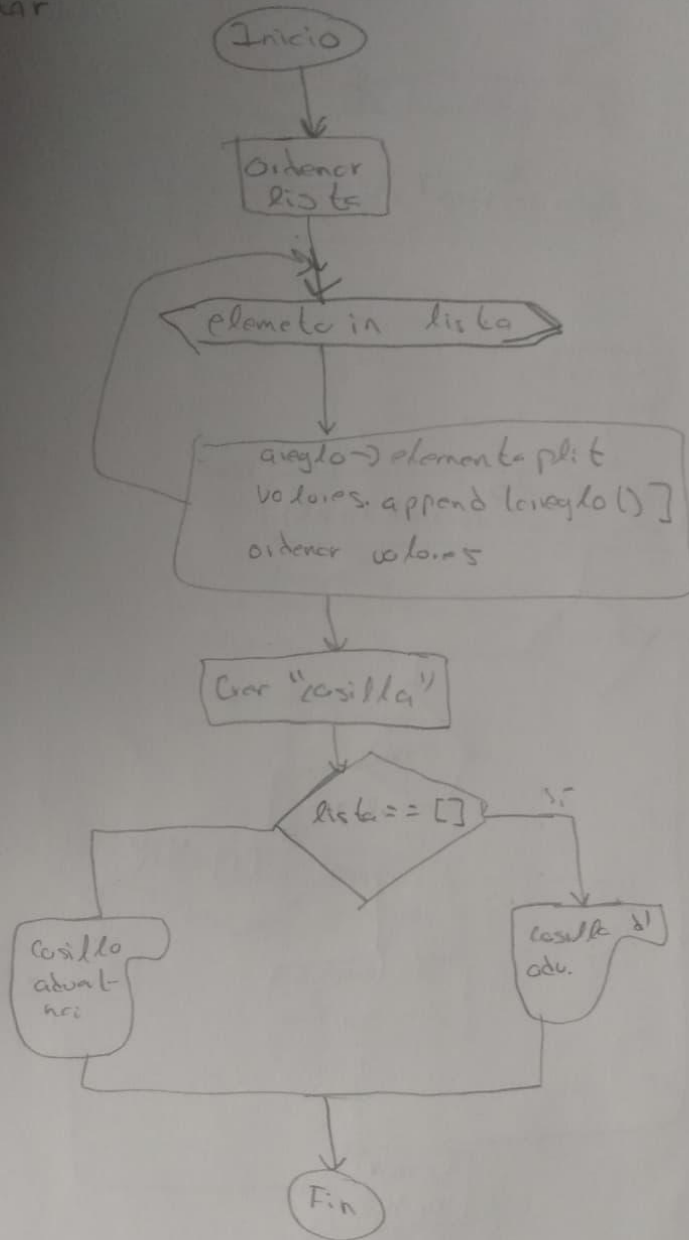
Primer archivo



Segundo

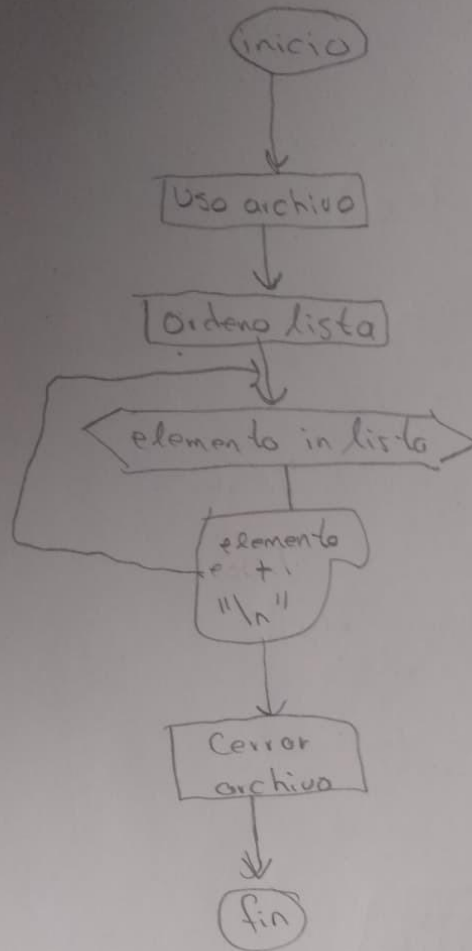


Consultar

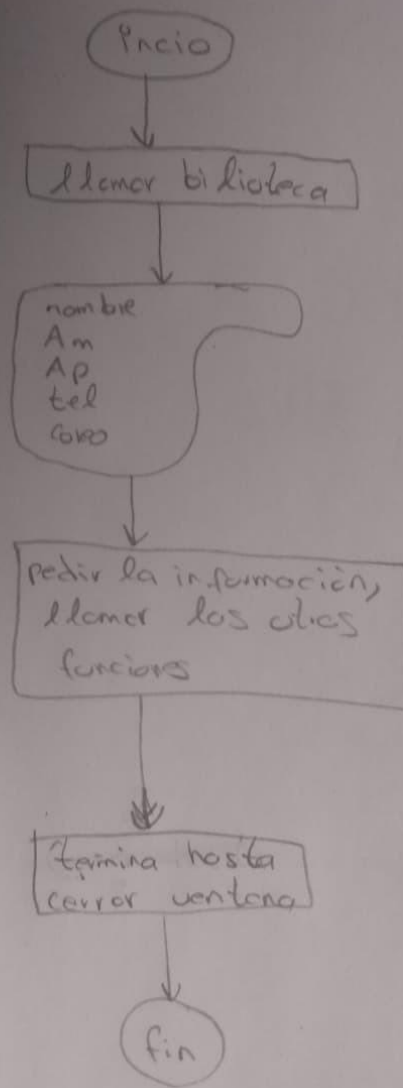




Escribir contactos



in lafca2



## Código

```
File Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda
ers\danielalfonso\Downloads\Sin título1.py
Agenda1_2.py Sin título0.py* _preuba.py Sin título1.py*

1 from tkinter import * #bibliotecas a usar
2 from tkinter import messagebox
3 lista = [] #en esta parte se almacenan los contactos
4 def guardar():
5     n = nombre.get() #para poder extraer la informacion
6     ap = app.get()
7     am = apm.get()
8     c = correo.get()
9     t = telefono.get()
10    lista.append(n+"$"+ap+"$"+am+"$"+c+"$"+t)
11    escribirContacto()
12    messagebox.showinfo("guardando","el contacto ha sido guardado correctamente")
13    nombre.set("") #se limpian las cajas cuando agregamos
14    app.set("")
15    apm.set("")
16    telefono.set("")
17    correo.set("")
18    consultar()
19
20 def Eliminar():
21     eliminado = conteliminar.get()
22     removido = False
23     for elemento in lista: #verificamos si existe un contacto en la lista
24         arreglo=elemento.split("$")
25         if conteliminar.get() == arreglo[3]:
26             lista.remove(elemento)
27             removido=True #confirmamos si esta borrando
28     escribirContacto()
29     consultar()
30     if removido:
31         messagebox.showinfo("Eliminar","Contacto eliminado"+eliminado)
32
33 def consultar():
34     r = Text (ventana,width=80,height=15)
35     lista.sort()
36     valores=[] #creamos lista
37     for i in range(len(lista)):
38         valores.append(lista[i])
39     r.insert(1, "\n".join(valores))
39
40 def escribirContacto():
41     f = open("agenda.txt","a")
42     for i in range(len(lista)):
43         f.write(lista[i]+"\n")
44     f.close()
45
46 def limpiar():
47     nombre.set("")
48     app.set("")
49     apm.set("")
50     telefono.set("")
51     correo.set("")
52
53 def __init__(self):
54     self.ventana = Tk()
55     self.ventana.title("Agenda")
56     self.ventana.geometry("400x400")
57     self.ventana.configure(bg="lightblue")
58
59     self.nombre = Entry(self.ventana)
60     self.ap = Entry(self.ventana)
61     self.apm = Entry(self.ventana)
62     self.telefono = Entry(self.ventana)
63     self.correo = Entry(self.ventana)
64     self.conteliminar = Entry(self.ventana)
65
66     self.nombre.grid(row=1,column=1)
67     self.ap.grid(row=1,column=2)
68     self.apm.grid(row=1,column=3)
69     self.telefono.grid(row=1,column=4)
70     self.correo.grid(row=1,column=5)
71     self.conteliminar.grid(row=1,column=6)
72
73     self.bguardar = Button(self.ventana, text="Guardar", command=self.guardar)
74     self.beliminar = Button(self.ventana, text="Eliminar", command=self.Eliminar)
75     self.bconsultar = Button(self.ventana, text="Consultar", command=self.consultar)
76     self.blimpiar = Button(self.ventana, text="Limpiar", command=self.limpiar)
77
78     self.bguardar.grid(row=2,column=1)
79     self.beliminar.grid(row=2,column=2)
80     self.bconsultar.grid(row=2,column=3)
81     self.blimpiar.grid(row=2,column=4)
82
83     self.ventana.mainloop()
84
85 if __name__ == '__main__':
86     agenda = Agenda()
87     agenda.__init__()
```

Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

(danielalfonso)\Downloads\Sin título1.py

enda1\_2.py

Sin título0.py\*

\_preuba.py

Sin título1.py\*

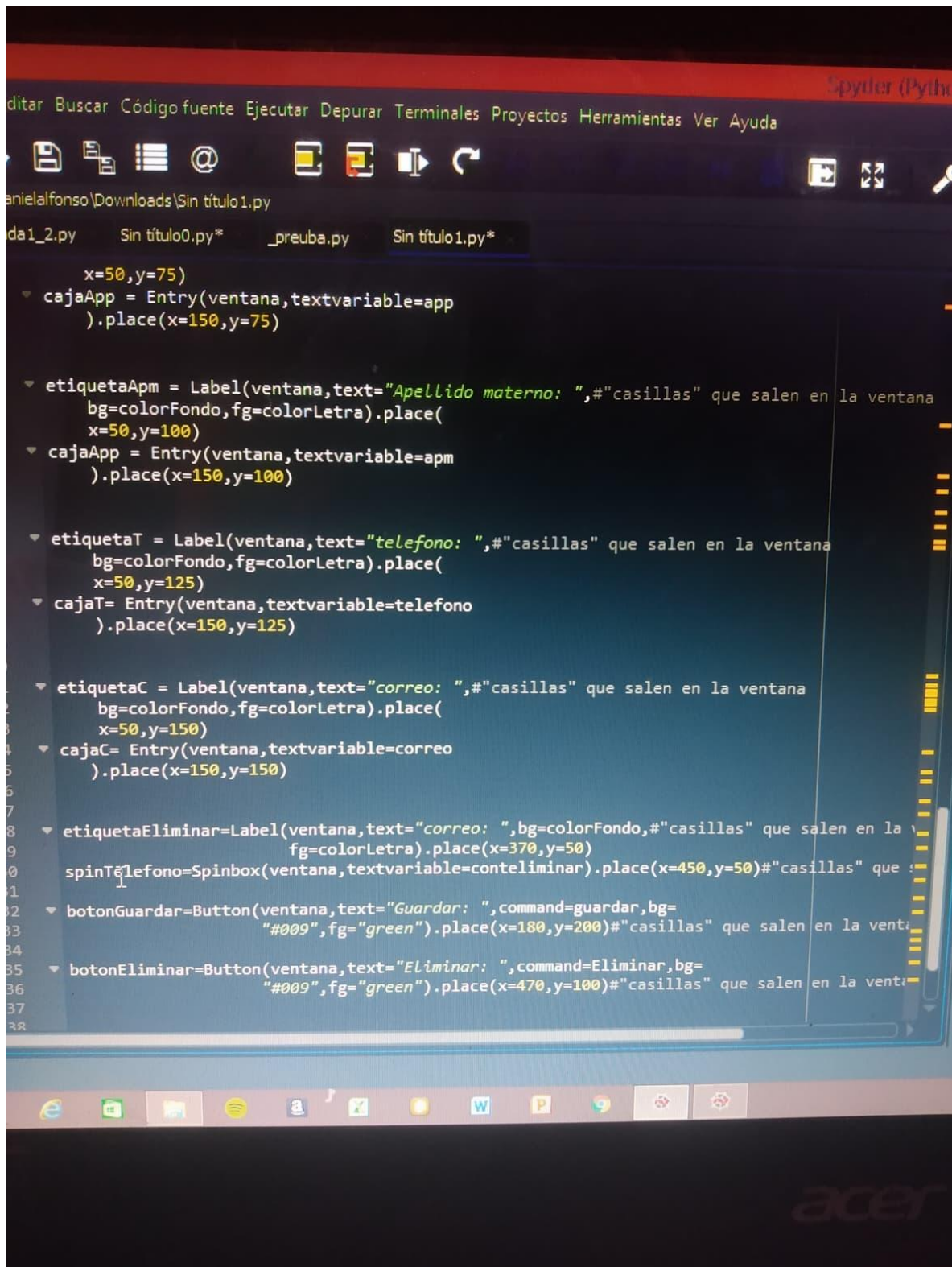
```
r = Text (ventana,width=80,height=15)
lista.sort()
valores=[] #creamos lista
r.insert(INSERT,"nombre \t\tapellido_p \t\tapellido_m \t\tcorreo \t\ttel")
for elemento in lista:
    arreglo=elemento.split("$")
    valores.append(arreglo[3])
    r.insert(INSERT,arreglo[0]+" \t\t"+arreglo[1]+" \t\t"+arreglo[2]+" \t\t"+
        arreglo[4]+" \t\t\n") #asignamos los valores de como se iran
r.place(x=20,y=250)
spinTelefono=Spinbox(ventana,value=(valores),textvariable=
    conteliminar).place(x=450,y=50)
if lista == []:
    spinTelefono=Spinbox(ventana,value=(valores)).place(x=450,y=50)
r.config(state=DISABLED) #para que no se pueda reconfigurar el spinbox
def inciararchivo(): #crear archivo de texto paa tener almacena en otro lugar
    archivo=open("agt.txt","a")
    archivo.close()
def cargar():
    archivo=open("agt.txt","r") #vamos agregando info al archivo
    linea = archivo.readline()
    if linea:
        while linea:
            if linea[-1] == '\n':
                linea=linea[:-1]
                lista.append(linea)
                linea=archivo.readline()
            else:
                linea=archivo.readline()
        archivo.close()
def escribirContacto(): #la info se va agregando en la pantalla y en archivo
    archivo=open("agt.txt","w")
    lista.sort() #se ordenan
    for elemento in lista:
        archivo.write(elemento + "\n")
    archivo.close()
```



```
Editor  Ejecutar  Depurar  Terminales  Proyectos  Herramientas  Ver A...
danielalfonso\Downloads\Sin título1.py
Sin título1_2.py  Sin título0.py*  _preuba.py  Sin título1.py*

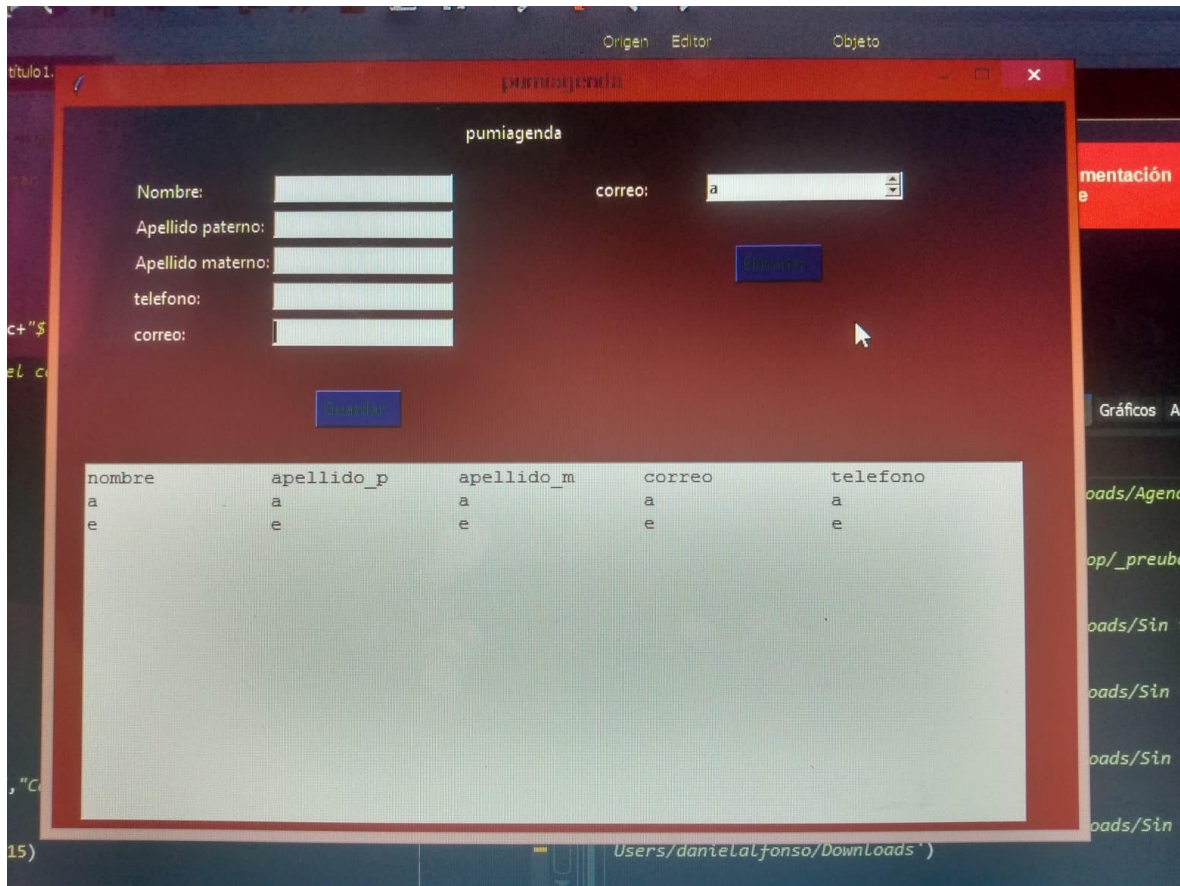
ventana = Tk()

nombre = StringVar()  #declaramos variables
app = StringVar()
apm = StringVar()
correo = StringVar()
telefono = StringVar()
conteliminar = StringVar()
colorFondo = "#700"  #eliguimos colores
colorLetra = "#FFF"
inciararchivo()  #mandamos a llamr las funciones
cargar()
consultar()
ventana.title("pumiagenda")
ventana.geometry("700x500")
ventana.configure(background=colorFondo)
▼ etiquetaTitulo = Label(ventana,  #el titulo
    text="pumiagenda",bg=
    colorFondo,fg=colorLetra).place(x=280,
    y=10)
▼ etiquetaN = Label(ventana,text="Nombre: ",  #"casillas" que salen en
    x=50,y=50)
▼ cajaN = Entry(ventana,textvariable=nombre
    ).place(x=150,y=50)
▼ etiquetaApp = Label(ventana,text="Apellido paterno: ",#"casillas" que sa
    bg=colorFondo,fg=colorLetra).place(
    x=50,y=75)
▼ cajaApp = Entry(ventana,textvariable=app
    ).place(x=150,y=75)
▼ etiquetaApm = Label(ventana,text="Apellido materno: ",#"casillas" que sal
    bg=colorFondo,fg=colorLetra).place(
    x=50,y=100)
```



# Resultados

## Capturas de funcionamiento



Link a video

<https://youtu.be/-34qIKTgzbQ>

## Conclusiones

Fue un proyecto que me hizo aprender mucho, mejorado mis habilidades en donde tengo que repasar aun más, me agrado que lo pudiera hacer en el lenguaje que quisiera eso me dio mucho libertad al igual que poder escoger mi propio tema o que quería desarrollar. Anqué tuve algunas complicaciones al inicio ya que es la primera vez que usaba interfaz grafica tuve que ver muchos tutoriales para poder lograrlo. Al crear solo el proyecto te haces me hizo más responsable ya que toda la responsabilidad recae en uno mismo. Cabe resaltar que me base la idea un poco en varios códigos de terceros

# Referencias

<http://lcp02.fi-b.unam.mx/>

<https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/>

<https://www.youtube.com/watch?v=W2kAF9pKPPE>