Universidad Nacional de Colombia
Sede Medellín

Facultad de Ciencias

Actividad 6

Estudiante

Maria Fernanda Valencia Jimenez

Docente

Walter Hugo Arboleda Mazo

Julio 2025

Índice

1. Ejercicio

1.1. Código fuente

```
import tkinter as tk
from tkinter import messagebox
import os

class AgendaContactosApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Agenda de Contactos")
        self.geometry("580x370")
        self.resizable(False, False)
        self.centrar_ventana()
        self.construir_interfaz()

    def centrar_ventana(self):
        self.update_idletasks()
        ancho = self.winfo_width()
        alto = self.winfo_height()
        pos_x = (self.winfo_screenwidth() // 2) - (ancho
            // 2)
        pos_y = (self.winfo_screenheight() // 2) - (alto
            // 2)
        self.geometry(f"{ancho}x{alto}+{pos_x}+{pos_y}")

    def construir_interfaz(self):
        # Campos de entrada
        tk.Label(self, text="Nombre:").pack(pady=4)
        self.entrada_nombre = tk.Entry(self, width=58)
        self.entrada_nombre.pack()

        tk.Label(self, text="Teléfono:").pack(pady=4)
        self.entrada_telefono = tk.Entry(self, width=58)
        self.entrada_telefono.pack()

        # Botones
```

```
        botones_info = [
            ("Añadir", self.agregar_contacto),
            ("Ver Todos", self.ver_contactos),
            ("Editar", self.editar_contacto),
            ("Quitar", self.quitar_contacto)
        ]
        for texto, comando in botones_info:
            tk.Button(self, text=texto, command=comando).
                pack(pady=3)

        # Área de texto
        self.panel_contactos = tk.Text(self, width=58,
            height=14, state=tk.DISABLED)
        self.panel_contactos.pack(pady=8)

    def agregar_contacto(self):
        nombre = self.entrada_nombre.get().strip()
        telefono_raw = self.entrada_telefono.get().strip()

        if not nombre or not telefono_raw:
            messagebox.showwarning("Faltan Datos", "Debe
                completar ambos campos.")
            return

        try:
            telefono = int(telefono_raw)
        except ValueError:
            messagebox.showerror("Nú'mero Inválido", "
                Ingrese solo dígitos en el mú'mero.")
            return

        archivo = "friendsContact.txt"
        if not os.path.exists(archivo):
            open(archivo, 'a', encoding="utf-8").close()

        duplicado = False
        with open(archivo, 'r', encoding="utf-8") as f:
            for linea in f:
                datos = linea.strip().split("!")
                if len(datos) == 2 and (datos[0] == nombre
                     or datos[1] == str(telefono)):
                    duplicado = True
                    break

        if duplicado:
            messagebox.showwarning("Duplicado", "Ya existe
                 un contacto con este nombre o mú'mero.")
        else:
```

```
            with open(archivo, 'a', encoding="utf-8") as f
                :
                f.write(f"{nombre}!{telefono}\n")
            messagebox.showinfo("Guardado", "Contacto añ
                adido correctamente.")

        self.limpiar_campos()
        self.ver_contactos()

    def ver_contactos(self):
        self.panel_contactos.config(state=tk.NORMAL)
        self.panel_contactos.delete(1.0, tk.END)

        archivo = "friendsContact.txt"
        if not os.path.exists(archivo) or os.path.getsize(
            archivo) == 0:
            self.panel_contactos.insert(tk.END, "No hay
                contactos registrados.")
        else:
            con_datos = False
            with open(archivo, 'r', encoding="utf-8") as f
                :
                for linea in f:
                    datos = linea.strip().split("!")
                    if len(datos) == 2:
                        self.panel_contactos.insert(tk.END
                            , f"Nombre: {datos[0]}, Telé
                            fono: {datos[1]}\n")
                        con_datos = True
            if not con_datos:
                self.panel_contactos.insert(tk.END, "Lista
                     vacía de contactos.")

        self.panel_contactos.config(state=tk.DISABLED)

    def editar_contacto(self):
        nombre = self.entrada_nombre.get().strip()
        telefono_raw = self.entrada_telefono.get().strip()

        if not nombre or not telefono_raw:
            messagebox.showwarning("Faltan Datos", "Debe
                completar ambos campos.")
            return

        try:
            telefono = int(telefono_raw)
        except ValueError:
            messagebox.showerror("Nú'mero Inválido", "
```

```
                Ingrese solo dígitos en el mí'mero.")
            return

        archivo = "friendsContact.txt"
        if not os.path.exists(archivo):
            messagebox.showwarning("Sin Datos", "No hay
                contactos para actualizar.")
            return

        actualizado = False
        nuevos_datos = []
        with open(archivo, 'r', encoding="utf-8") as f:
            for linea in f:
                datos = linea.strip().split("!")
                if len(datos) == 2:
                    if datos[0] == nombre:
                        nuevos_datos.append(f"{nombre}!{
                            telefono}")
                        actualizado = True
                    else:
                        nuevos_datos.append(linea.strip())

        if actualizado:
            with open(archivo, 'w', encoding="utf-8") as f
                :
                f.write("\n".join(nuevos_datos) + "\n")
            messagebox.showinfo("Actualizado", "El
                contacto ha sido modificado.")
        else:
            messagebox.showwarning("No Encontrado", "No se
                 encontró' un contacto con ese nombre.")

        self.limpiar_campos()
        self.ver_contactos()

    def quitar_contacto(self):
        nombre = self.entrada_nombre.get().strip()

        if not nombre:
            messagebox.showwarning("Falta Nombre", "
                Indique el nombre del contacto a eliminar
                .")
            return

        archivo = "friendsContact.txt"
        if not os.path.exists(archivo):
            messagebox.showwarning("Sin Datos", "No hay
                contactos para eliminar.")
```

```
            return

        eliminado = False
        restantes = []
        with open(archivo, 'r', encoding="utf-8") as f:
            for linea in f:
                datos = linea.strip().split("!")
                if len(datos) == 2:
                    if datos[0] == nombre:
                        eliminado = True
                    else:
                        restantes.append(linea.strip())

        if eliminado:
            with open(archivo, 'w', encoding="utf-8") as f
                :
                f.write("\n".join(restantes) + "\n")
            messagebox.showinfo("Eliminado", "Contacto
                eliminado correctamente.")
        else:
            messagebox.showwarning("No Encontrado", "No
                existe un contacto con ese nombre.")

        self.limpiar_campos()
        self.ver_contactos()

    def limpiar_campos(self):
        self.entrada_nombre.delete(0, tk.END)
        self.entrada_telefono.delete(0, tk.END)

if __name__ == "__main__":
    app = AgendaContactosApp()
    app.mainloop()
```

1.2. Diagrama de clase

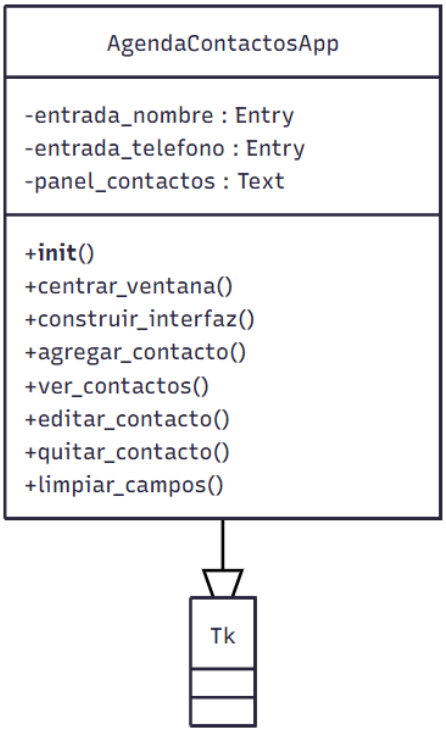

Figura 1: Diagrama de clase

1.3. Casos de uso

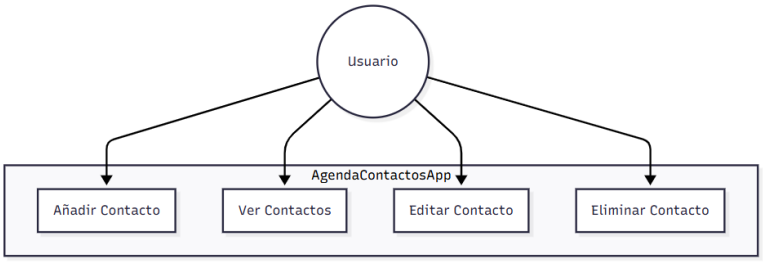

Figura 2: Casos de uso

1.4. Interfaz de usuario

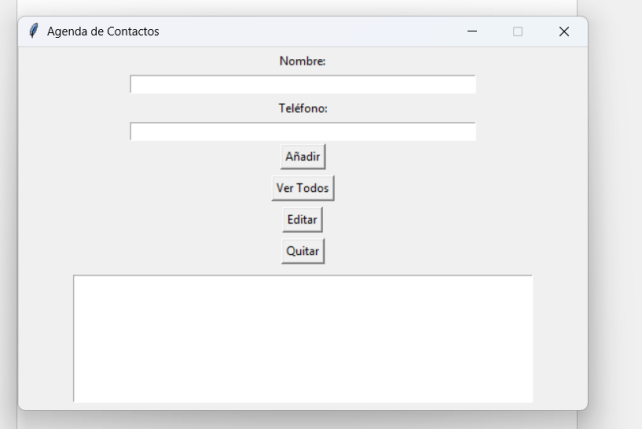

Figura 3: Interfaz de usuario

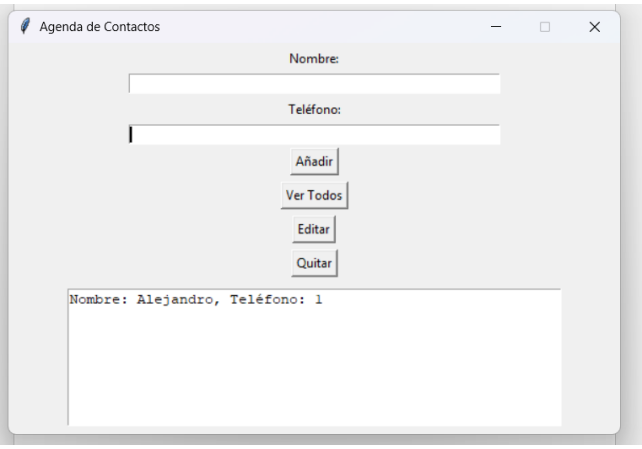

Figura 4: Interfaz de usuario

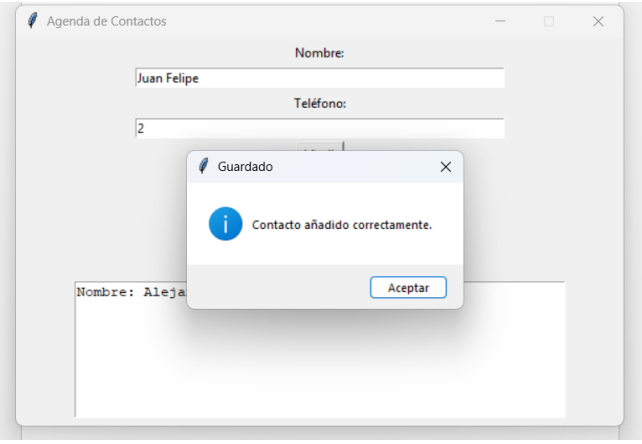

Figura 5: Interfaz de usuario

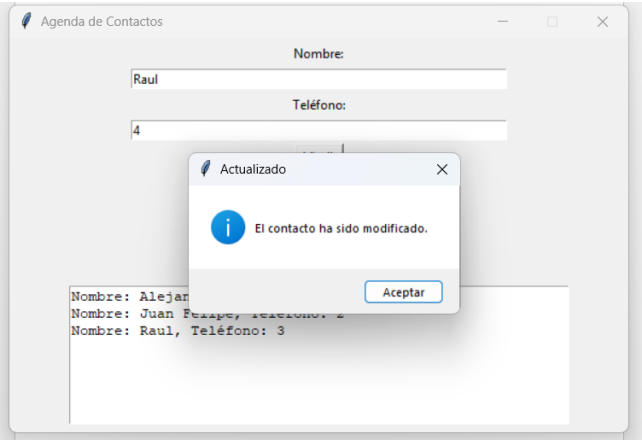

Figura 6: Interfaz de usuario

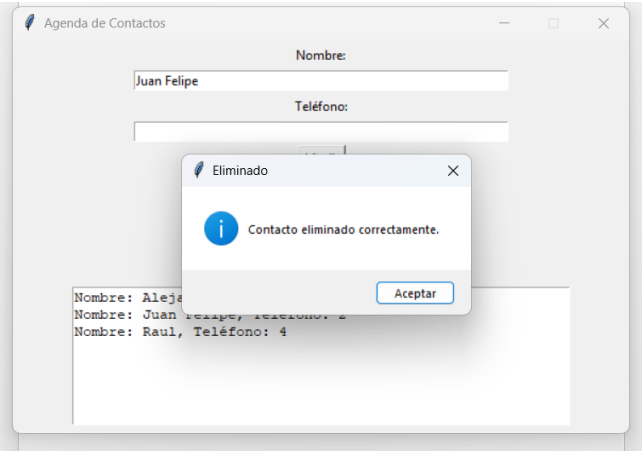

Figura 7: Interfaz de usuario

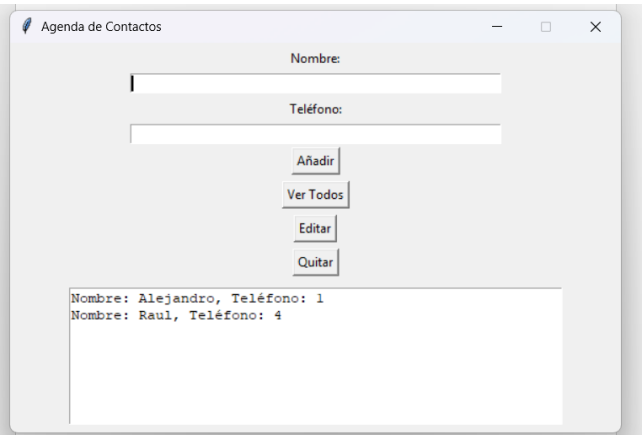

Figura 8: Interfaz de usuario