



**FACULTAD DE INGENIERIA  
INGENIERÍA INFORMÁTICA  
COMUNITARIA DE CAACUPE**

## **TRABAJO PRÁCTICO**

### **NOMBRE Y APELLIDO**

JESUS MIGUEL PORTILLO LOPEZ

### **MATERIA**

PROGRAMACION VI

### **SEMESTRE**

OCTAVO

PROF. RICARDO ALBERTO MAIDANA

CAACUPÉ PARAGUAY

OCTUBRE 2024

# Introducción

En la actualidad, la organización y manejo eficaz de la información es fundamental para optimizar diversos procesos en entornos educativos, laborales y personales. El presente proyecto presenta el desarrollo de una aplicación de gestión de usuarios, diseñada para facilitar el registro de datos básicos, implementada con Python, el framework Flet y SQLite. La aplicación permite a los usuarios introducir datos personales básicos, como nombre y edad, ofrece opciones para guardar y limpiar la información ingresada.

La aplicación tiene como objetivo ofrecer una interfaz intuitiva y funcional que permita realizar las operaciones CRUD (Crear, Leer, Actualizar y Borrar) de forma eficaz, asegurando también una experiencia de usuario fácil y organizada. Además, se integran elementos visuales y lógicos que permiten al usuario verificar y gestionar la información ingresada, donde evalúa la edad del usuario y despliega un mensaje si es mayor o menor de edad.

# Aplicación Gestión de Usuarios

## Código Funcional

```
import flet as ft
import sqlite3

# Conexión a la base de datos
def crear_conexion():
    conexion = sqlite3.connect('usuarios.db')
    return conexion

# Crear la tabla si no existe
def crear_tabla():
    conexion = crear_conexion()
    cursor = conexion.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS usuarios (
            id INTEGER PRIMARY KEY,
            nombre TEXT NOT NULL,
            edad INTEGER NOT NULL
        )
    """)
    conexion.commit()
    conexion.close()

def guardar_datos(nombre, edad):
    conexion = crear_conexion()
    cursor = conexion.cursor()
    cursor.execute("INSERT INTO usuarios (nombre, edad) VALUES (?, ?)", (nombre, edad))
    conexion.commit()
    conexion.close()

def main(page: ft.Page):
    page.title = "Gestión de Usuarios"
    page.horizontal_alignment = ft.CrossAxisAlignment.CENTER
    page.vertical_alignment = ft.MainAxisAlignment.CENTER

    # Crear la tabla si no existe
    crear_tabla()

    # Crear el título con fondo blanco translúcido
    title_text = ft.Container(
        content=ft.Text("Gestión de Usuarios", size=30, color=ft.colors.BLACK),
        bgcolor=ft.colors.WHITE,
        opacity=0.7,
        padding=ft.padding.all(10),
        border_radius=ft.border_radius.all(5)
    )

    # Crear los componentes de la interfaz con fondos transparentes
    nombre_input = ft.TextField(
```

```

        label="Nombre",
        width=300,
        bgcolor="#FFFFFFA0",
        border_color="#FFFFFFA0"
    )
    edad_input = ft.TextField(
        label="Edad",
        width=300,
        bgcolor="#FFFFFFA0",
        border_color="#FFFFFFA0"
    )

    mensaje_container = ft.Column()

    def guardar_click(e):
        mensaje_container.controls.clear()
        nombre = nombre_input.value
        edad = edad_input.value

        if nombre and edad.isdigit():
            edad = int(edad)
            guardar_datos(nombre, edad)
            # Mensaje de resultado con estilo
            mensaje_text = ft.Container(
                content=ft.Text(
                    f"{nombre} es {'mayor' if edad >= 18 else 'menor'} de edad.",
                    size=20, # Tamaño de texto más grande
                    color=ft.colors.BLACK
                ),
                bgcolor=ft.colors.WHITE,
                opacity=0.7,
                padding=ft.padding.all(10),
                border_radius=ft.border_radius.all(5)
            )
            mensaje_container.controls.append(mensaje_text)
            nombre_input.value = ""
            edad_input.value = ""
        else:
            mensaje_container.controls.append(ft.Text("Por favor, ingrese un nombre
válido y una edad.", color=ft.colors.RED))

        page.update()

    guardar_button = ft.ElevatedButton("Guardar", on_click=guardar_click,
bgcolor=ft.colors.BLUE_500, color=ft.colors.WHITE)

    def limpiar_click(e):
        nombre_input.value = ""
        edad_input.value = ""
        mensaje_container.controls.clear()
        page.update()

```

```

limpiar_button = ft.ElevatedButton("Limpiar", on_click=limpiar_click,
bgcolor=ft.colors.RED_500, color=ft.colors.WHITE)

# Contenedor principal de contenido
content_container = ft.Column(
    controls=[
        title_text,
        nombre_input,
        edad_input,
        guardar_button,
        limpiar_button,
        mensaje_container
    ],
    alignment=ft.MainAxisAlignment.CENTER,
    horizontal_alignment=ft.CrossAxisAlignment.CENTER,
)

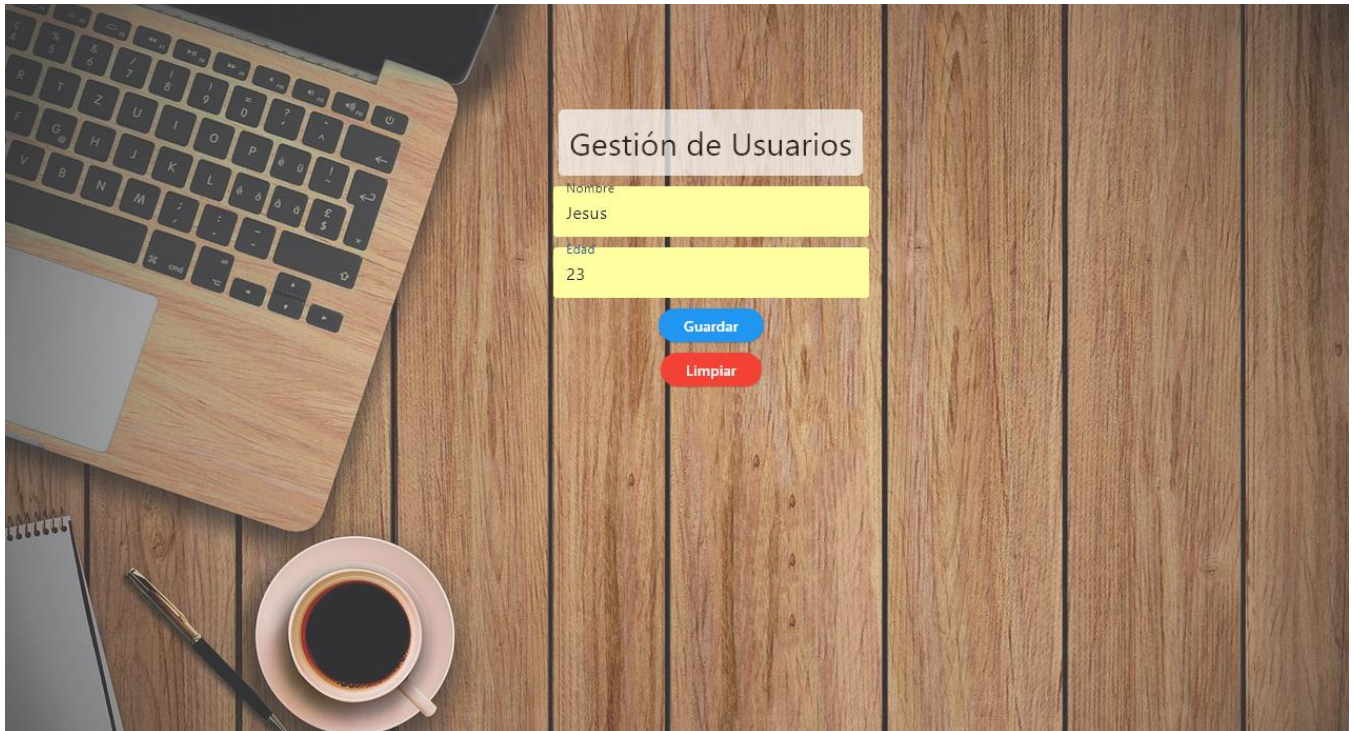
# Estructura de la página
page.add(
    ft.Stack(
        [
            ft.Image(
                src="C:\\Users\\Owner\\OneDrive\\Documents\\SegundaParcial\\fondo.j
pg",
                fit=ft.ImageFit.COVER,
                opacity=0.8
            ),
            ft.Container(
                content=content_container,
                alignment=ft.alignment.center,
                padding=ft.padding.only(top=100)
            )
        ]
    )
)

# Iniciar la aplicación
ft.app(target=main)

```

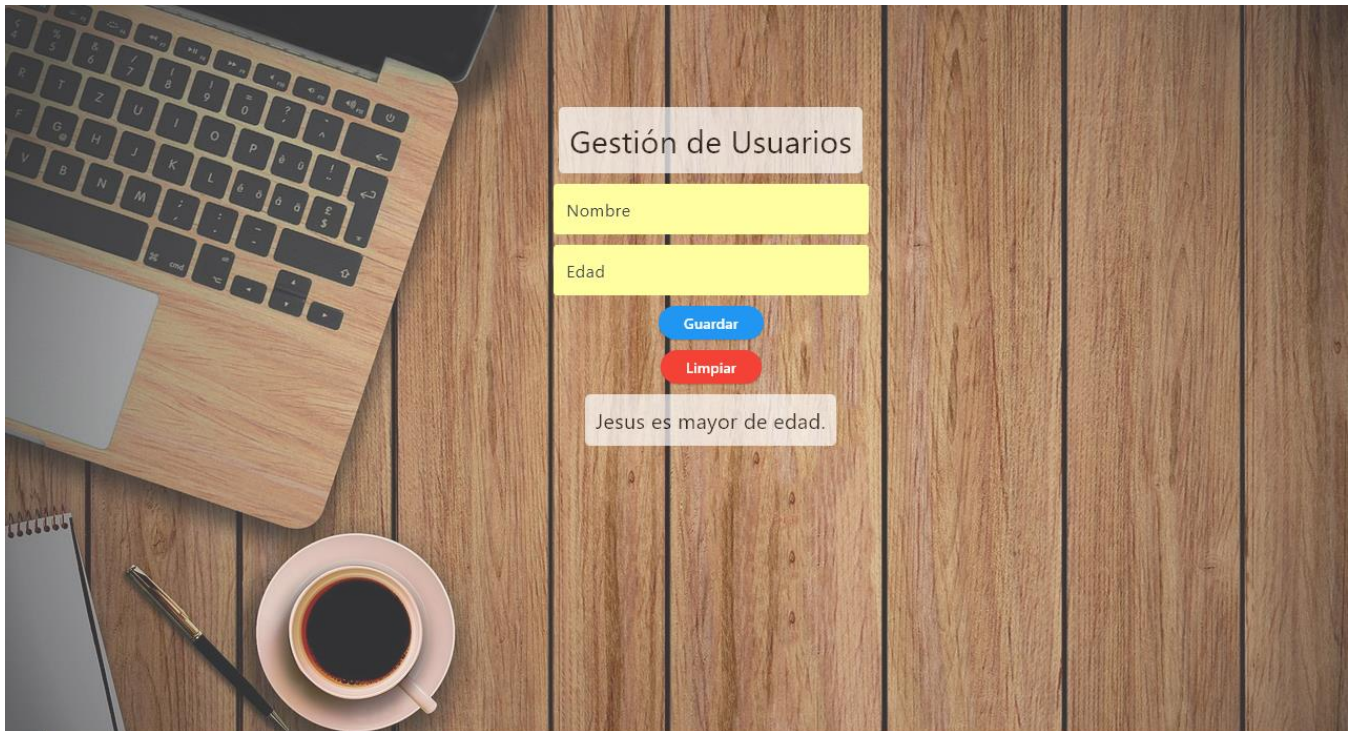
## Interfaz de la Aplicación

En la ventana principal, el usuario puede ver un título con el texto "Gestión de Usuarios", estilizado con fondo blanco translúcido para destacarse sobre la imagen de fondo, con dos campos de entrada principales para "Nombre" y "Edad," los cuales permiten el ingreso de la información del usuario. Ambos campos tienen un fondo y bordes translúcidos, lo que brinda un diseño limpio y sencillo.





Debajo de los campos de entrada, hay dos botones principales: uno para "Guardar," que procesa y almacena los datos ingresados en una base de datos SQLite, y otro botón de "Limpiar" que restablece los campos de entrada a su estado inicial. Adicionalmente, un área de mensajes muestra resultados y confirma si el usuario es mayor o menor de edad, con un diseño similar al título: texto sobre fondo blanco translúcido. La imagen de fondo, aplicada con transparencia, añade un toque visual sin distraer del contenido principal.



## Conclusión

El desarrollo de esta aplicación de gestión de usuarios representa una solución factible para la organización y visualización de datos personales en un entorno controlado. A través de esta herramienta, se ha logrado implementar una interfaz básica que incorpora validación de entradas, manejo de errores y una lógica de procesamiento sencilla para evaluar la mayoría de edad. La utilización de la biblioteca Flet para el diseño de la interfaz gráfica y SQLite para el almacenamiento de datos ha permitido crear una estructura robusta y funcional. En conclusión, esta aplicación cumple con los requisitos fundamentales de accesibilidad, funcionalidad y simplicidad, y su diseño modular permite futuras expansiones y adaptaciones para entornos más complejos, mejorando así su aplicabilidad y alcance en el contexto de gestión de información.

### LINK DE GITHUB

<https://github.com/JesusPortillo19/Segunda-Parcial.git>