



EXAMEN FINAL DE PROGRAMACION VI

- **Nombre y apellido:** Francisco Ariel Galeano Ruiz Díaz
- **Profesor:** Ricardo Alberto Maidana Giménez
- **Materia:** Programación VI
- **Fecha:** 10/12/2024

**CAACUPE, PARAGUAY
2024**

INDICE

INTRODUCCION	3
OBJETIVOS DE LA INVESTIGACION	4
Objetivo General:.....	4
Objetivo Específico:.....	4
APP PARA VERIFICACION DE LA EDAD.	5
CODIGO FUNCIONAL DEL PROYECTO.....	5
CAPTURAS DE PANTALLA.	10
CONCLUSION.....	13

INTRODUCCION

Este presente proyecto tiene como objetivo desarrollar una aplicación de escritorio que permita realizar la gestión de usuarios y verificar edades, combinando una interfaz amigable con funcionalidades clave como la autenticación mediante contraseña, el almacenamiento de datos en una tabla dinámica, y la exportación del historial. La aplicación está diseñada para ser sencilla, práctica y eficiente, asegurando una experiencia positiva para el usuario.

A medida que la tecnología avanza, las herramientas de software personalizadas para tareas específicas han ganado importancia. Este proyecto se enfoca en ofrecer una forma práctica de recolectar y administrar información de usuarios en tiempo real.

OBJETIVOS DE LA INVESTIGACION

Objetivo General:

Desarrollar una aplicación de escritorio que permita la verificación de usuarios, gestión de edades y exportación de registros en tiempo real, brindando una interfaz gráfica intuitiva y funcional que facilite la experiencia del usuario.

Objetivo Específico:

- Implementar un sistema de autenticación que valide las credenciales del usuario para garantizar la seguridad de los datos.
- Desarrollar una funcionalidad para verificar y registrar la información de los usuarios, categorizándolos según su edad.
- Crear una tabla dinámica que permita la edición, eliminación y visualización de los datos registrados de manera eficiente.
- Incorporar una funcionalidad para exportar el historial de usuarios a un archivo de texto, asegurando la persistencia de los datos.
- Diseñar una interfaz gráfica atractiva y funcional que facilite la interacción del usuario con la aplicación.

APP PARA VERIFICACION DE LA EDAD.

CODIGO FUNCIONAL DEL PROYECTO.

```
import flet as ft
from datetime import datetime

# Contraseña fija
CONTRASENA_CORRECTA = "Fran12345"

# Lista global para almacenar nombres, edades, y fechas
lista_entradas = []

class App:
    def __init__(self, page: ft.Page):
        self.page = page
        self.page.title = "Examen Final"
        self.page.bgcolor = "#EAF6F6"
        self.styles = {
            "titulo": {"size": 30, "color": "#333333"},
            "texto": {"size": 20, "color": "#666666"},
            "boton": {"color": "white", "bgcolor": "#007BFF"}
        }
        self.main_view()

    def main_view(self):
        """Vista principal de inicio de sesión."""
        self.page.clean()
        usuario_input = ft.TextField(label="Usuario", width=250)
        contrasena_input = ft.TextField(label="Contraseña", width=250,
password=True)
        mensaje_error = ft.Text(value="", color="red")

        # Imagen del logo
        logo = ft.Image(
            src="C:\\Users\\TOSHIBA\\P6 Python - Francisco
Galeano\\Logo.JPG",
            width=400,
            height=300,
            fit=ft.ImageFit.CONTAIN,
        )

        # Botón para iniciar sesión
        boton_iniciar = ft.ElevatedButton(
            text="Iniciar",
            style=ft.ButtonStyle(
                color=self.styles["boton"]["color"],
                bgcolor=self.styles["boton"]["bgcolor"]
            ),
        ),
```

```

        on_click=lambda e: self.verificar_contrasena(
            contrasena_input.value, mensaje_error
        ),
    )

    contenido = ft.Column(
        controls=[
            logo,
            ft.Text("Iniciar Sesión", **self.styles["titulo"]),
            usuario_input,
            contrasena_input,
            ft.Row(
                controls=[ft.Checkbox(label="Recordar contraseña")],
                alignment=ft.MainAxisAlignment.CENTER,
            ),
            boton_iniciar,
            mensaje_error,
        ],
        alignment=ft.MainAxisAlignment.CENTER,
        horizontal_alignment=ft.CrossAxisAlignment.CENTER,
        spacing=10,
    )
    self.page.add(contenido)

def verificar_contrasena(self, contrasena, mensaje_error):
    """Verifica la contraseña ingresada por el usuario."""
    if contrasena == CONTRASENA_CORRECTA:
        mensaje_error.value = ""
        self.verificacion_edad_view()
    else:
        mensaje_error.value = "Contraseña incorrecta. Intenta
nuevamente."
        mensaje_error.update()

def verificacion_edad_view(self):
    """Vista para verificar edad."""
    self.page.clean()

    nombre_input = ft.TextField(label="Nombre", width=200)
    edad_input = ft.TextField(label="Edad", width=200)
    resultado = ft.Text(value="", color="black")

    def procesar_datos(e):
        """Procesa y valida los datos ingresados."""
        nombre = nombre_input.value.strip()
        edad = edad_input.value.strip()
        if not nombre or not edad:
            resultado.value = "Por favor, llena todos los campos."
        else:
            try:

```

```

        edad = int(edad)
        if edad < 0:
            resultado.value = "Por favor, ingresa una edad
válida (no negativa)."
        else:
            fecha_hora = datetime.now().strftime("%Y-%m-%d
%H:%M:%S")

            if edad >= 18:
                mensaje = f"Hola {nombre}, eres mayor de
edad."
            else:
                mensaje = f"Hola {nombre}, eres menor de
edad."

            lista_entradas.append(
                {"Nombre": nombre, "Edad": edad, "Fecha y
Hora": fecha_hora}
            )
            actualizar_lista()
            resultado.value = mensaje
    except ValueError:
        resultado.value = "Por favor, ingresa una edad válida
(número)."

    resultado.update()

def actualizar_lista():
    """Actualiza la tabla de entradas."""
    tabla_entradas.controls.clear()
    for i, entrada in enumerate(lista_entradas):
        fila = ft.Row(
            controls=[
                ft.TextField(value=entrada["Nombre"],
on_change=lambda e, i=i: editar_nombre(i, e.control.value)),
                ft.TextField(value=str(entrada["Edad"]),
on_change=lambda e, i=i: editar_edad(i, e.control.value)),
                ft.Text(value=entrada["Fecha y Hora"]),
                ft.IconButton(icon=ft.icons.DELETE,
on_click=lambda e, i=i: eliminar_entrada(i))
            ],
            alignment=ft.MainAxisAlignment.CENTER,
        )
        tabla_entradas.controls.append(fila)
    tabla_entradas.update()

def editar_nombre(indice, nuevo_nombre):
    lista_entradas[indice]["Nombre"] = nuevo_nombre

def editar_edad(indice, nueva_edad):
    try:
        lista_entradas[indice]["Edad"] = int(nueva_edad)
    except ValueError:

```

```

        pass # Ignorar valores inválidos temporalmente

    def eliminar_entrada(indice):
        lista_entradas.pop(indice)
        actualizar_lista()

    def exportar_datos(e):
        with open("historial_usuarios.txt", "w") as archivo:
            archivo.writelines(
                f"{entrada['Nombre']} - {entrada['Edad']} años - {entrada['Fecha y Hora']}\n"
                for entrada in lista_entradas
            )
        resultado.value = "Historial exportado a historial_usuarios.txt"
        resultado.update()

    def limpiar_campos(e):
        nombre_input.value = ""
        edad_input.value = ""
        resultado.value = ""
        nombre_input.update()
        edad_input.update()
        resultado.update()

    tabla_entradas = ft.Column()

    formulario = ft.Column(
        controls=[
            ft.Text("Verificación de Edad", **self.styles["titulo"]),
            nombre_input,
            edad_input,
            ft.Row(controls=[
                ft.ElevatedButton(text="Procesar",
on_click=procesar_datos),
                ft.ElevatedButton(text="Limpiar",
on_click=limpiar_campos),
                ft.ElevatedButton(text="Exportar Historial",
on_click=exportar_datos),
                ft.ElevatedButton(text="Salir", on_click=lambda e:
self.main_view()),
            ]),
            resultado,
        ],
    )

    self.page.add(
        ft.Column(
            controls=[
                formulario,

```



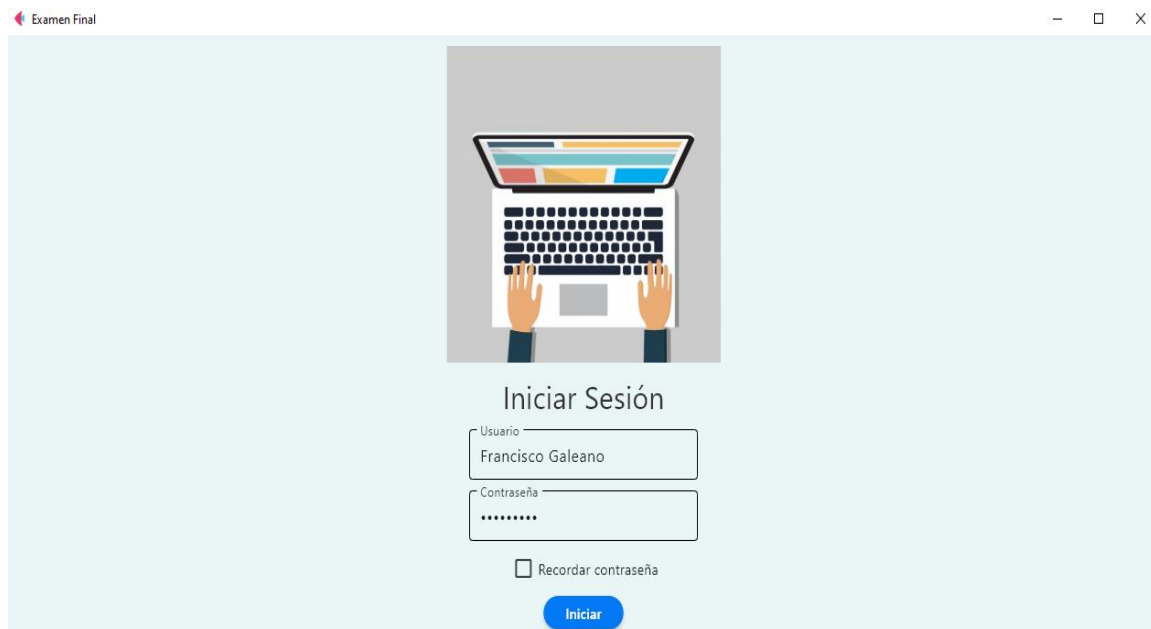
```
        ft.Text("Historial de Entradas:",
**self.styles["texto"]),
        tabla_entradas,
    ]
    )
)

# Ejecutar la aplicación
ft.app(target=lambda page: App(page))
```

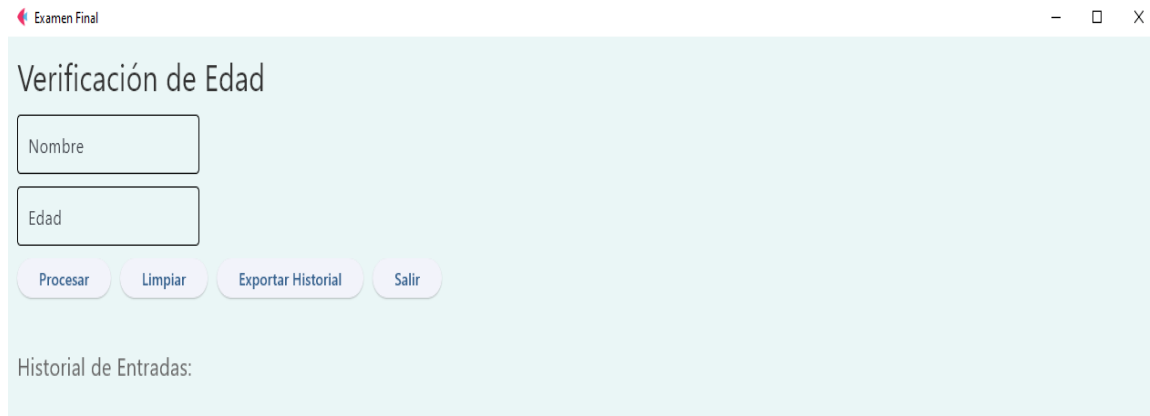
CAPTURAS DE PANTALLA.

APP PARA LA VERIFICACION DE LA EDAD.

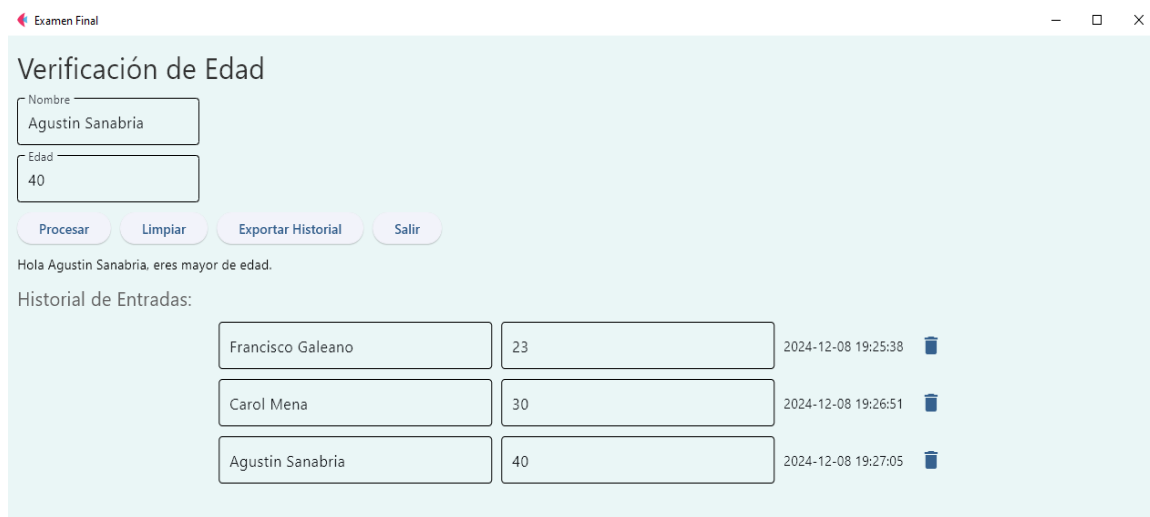
Al ejecutar el código se ve que funciona correctamente acá abajo se visualiza en la imagen que ingresa primeramente a la parte principal en donde se ve el logo que quiere decir que está escribiendo en una Pc para ingresar los datos del usuario, es para Iniciar la Sesión que se pone el nombre del usuario y la contraseña, al dar clic en el botón Iniciar ya ingresa al poner bien la contraseña.






Al ingresar el usuario se ve en la imagen que da la Bienvenida a la Aplicación Para la Verificación de La Edad de los Usuarios, se ve también en la caja de texto Nombre y la Edad más abajo se encuentran los botones de Procesar, Limpiar, Exportar Historial y Salir. Más abajo está la parte del Historial de Entradas.



En esta imagen se observa que ya están los datos cargados como el nombre y la edad de tres personas y damos clic al botón procesar y nos da resultado en el mensaje que es mayor de edad porque son mayores de edad. Abajo me guarda en el historial de entradas lo que he cargado.



Francisco Galeano	23	2024-12-08 19:25:38	
Carol Mena	30	2024-12-08 19:26:51	
Agustin Sanabria	40	2024-12-08 19:27:05	

En esta imagen se observa que cargue los datos de tres personas pero en este caso es menor de edad porque son menores ahí se visualiza en el resultado del mensaje del historial de entradas.

Examen Final

Verificación de Edad

Nombre
Juan Ortega

Edad
17

Procesar Limpiar Exportar Historial Salir

Hola Juan Ortega, eres menor de edad.

Historial de Entradas:

Fernando Martinez	15	2024-12-08 19:31:47	
Sol Medina	16	2024-12-08 19:32:18	
Juan Ortega	17	2024-12-08 19:32:38	

En este Imagen se observa los datos cargados de las personas para la verificación de la edad en total seis cargados, se puede cargar más.

Por ultimo quiero mencionar que al dar clic al botón Limpiar borra lo que cargaste, la Exportación Historial donde se exporta la historial de los usuarios en txt, el botón Salir se cierra y vuelve en la parte de Iniciar Sesión.

Examen Final

Verificación de Edad

Nombre
Agustin Sanabria

Edad
40

Procesar Limpiar Exportar Historial Salir

Historial exportado a historial_usuarios.txt

Historial de Entradas:

Fernando Martinez	15	2024-12-08 19:31:47	
Sol Medina	16	2024-12-08 19:32:18	
Juan Ortega	17	2024-12-08 19:32:38	
Francisco Galeano	23	2024-12-08 19:36:28	
Carol Mena	30	2024-12-08 19:36:50	
Agustin Sanabria	40	2024-12-08 19:37:09	

Link de Acceso Github.

https://github.com/FranciscoGaleano551/Examen_Final_ProgramacionVI.git

CONCLUSION

El desarrollo de esta aplicación para la verificación de edades demuestra cómo una herramienta tecnológica puede simplificar y optimizar la gestión de datos de usuarios mediante un diseño funcional e intuitivo. La implementación de funcionalidades como la autenticación, el manejo dinámico de datos y la exportación de registros no solo atiende a necesidades específicas, sino que también refleja buenas prácticas en programación orientada a la usabilidad.

Este proyecto ofrece un modelo práctico que puede ser extendido o adaptado a diferentes contextos, abriendo la posibilidad de futuras mejoras y personalizaciones. Así, cumple con su propósito de proporcionar una solución efectiva y accesible, demostrando la importancia de las aplicaciones orientadas al usuario en el ámbito tecnológico.