



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

EXAMEN FINAL

NOMBRE Y APELLIDO

JESUS MIGUEL PORTILLO LOPEZ

MATERIA

PROGRAMACION VI

SEMESTRE

OCTAVO

PROF. RICARDO ALBERTO MAIDANA

CAACUPÉ PARAGUAY

DICIEMBRE 2024

Introducción

En el contexto actual, el uso de herramientas tecnológicas se ha vuelto esencial para optimizar y automatizar procesos en diferentes áreas. La presente aplicación, desarrollada en Python utilizando el framework Flet, está diseñada como una solución funcional y adaptable que combina gestión de usuarios y un sistema básico de tienda de ropa denominado JP Store. Esta herramienta permite gestionar información básica de usuarios, validar accesos mediante un sistema de login, y ofrecer una experiencia de compra simulada con funcionalidades como agregar productos a un carrito, limpiar el carrito y finalizar compras mediante un enlace directo a WhatsApp.

Este desarrollo tiene como propósito reforzar las habilidades de programación y diseño de interfaces gráficas, empleando tecnologías actuales que se adaptan a necesidades reales. Además, fomenta la integración de bases de datos locales (SQLite) para el almacenamiento y manejo de datos.

Aplicación Tienda De Ropas

Código Funcional

```
import flet as ft
import sqlite3
import webbrowser

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

# Crear la tabla de usuarios 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()

# Guardar los datos del usuario en la base de datos
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()

# Función para verificar las credenciales de inicio de sesión
def verificar_credenciales(usuario, contrasena):
    if usuario == "admin" and contrasena == "1234":
        return True
    return False

# Función para enviar mensaje personalizado a WhatsApp
def enviar_whatsapp(carrito):
    # Crear el mensaje con los detalles del carrito
    mensaje = "\n".join([f"{p['nombre']} - ${p['precio']}" for p in carrito])
    url = f"https://wa.me/595982323178?text={mensaje}"
    webbrowser.open(url) # Abrir la URL de WhatsApp en el navegador

# Función para mostrar la interfaz de la tienda
def mostrar_tienda(page):
    productos = [
```

```

        {"nombre": "Remera", "precio": 35.00, "imagen":
"C:\\Users\\Owner\\OneDrive\\Documents\\SegundaParcial\\camisa.jpg"},
        {"nombre": "Pantalón", "precio": 50.00, "imagen":
"C:\\Users\\Owner\\OneDrive\\Documents\\SegundaParcial\\pantalon.jpg"},
        {"nombre": "Nike Air Force 1", "precio": 120.00, "imagen":
"C:\\Users\\Owner\\OneDrive\\Documents\\SegundaParcial\\zapatos.jpg"},
    ]
    carrito = []
    carrito_container = ft.Column()

    # Agregar producto al carrito
    def agregar_a_carrito(producto):
        carrito.append(producto)
        actualizar_carrito()

    # Actualizar carrito
    def actualizar_carrito():
        carrito_container.controls.clear()
        if carrito:
            carrito_text = "\n".join([f"{p['nombre']} - ${p['precio']}" for p in
carrito])
            carrito_container.controls.append(ft.Text(f"Carrito:\n{carrito_text}",
color=ft.colors.BLACK))
        else:
            carrito_container.controls.append(ft.Text("Carrito vacío.",
color=ft.colors.BLACK))
        page.update()

    # Crear la lista de productos
    productos_container = ft.Row(wrap=True, spacing=10, run_spacing=10)

    for producto in productos:
        productos_container.controls.append(
            ft.Container(
                content=ft.Column(
                    [
                        ft.Image(src=producto["imagen"], width=80, height=80,
fit=ft.ImageFit.CONTAIN),
                        ft.Text(producto["nombre"], color=ft.colors.BLACK),
                        ft.Text(f"${producto['precio']}", color=ft.colors.BLACK),
                        ft.ElevatedButton("Agregar", on_click=lambda e, p=producto:
agregar_a_carrito(p), bgcolor=ft.colors.GREEN_500),
                    ],
                    alignment=ft.MainAxisAlignment.CENTER,
                    horizontal_alignment=ft.CrossAxisAlignment.CENTER,
                ),
                padding=ft.padding.all(10),
                bgcolor=ft.colors.WHITE,
                border_radius=ft.border_radius.all(5),
            )
        )

```

```

# Cambiar interfaz a la tienda
page.controls.clear()
page.add(
    ft.Stack(
        [
            ft.Image(
                src="C:\\Users\\Owner\\OneDrive\\Documents\\SegundaParcial\\fondo1.
jpg",
                fit=ft.ImageFit.COVER,
                opacity=0.8
            ),
            ft.Container(
                content=ft.Column(
                    controls=[
                        ft.Container(
                            content=ft.Text("JP Store", size=34,
color=ft.colors.BLACK, weight=ft.FontWeight.BOLD),
                            bgcolor=ft.colors.WHITE,
                            opacity=0.7,
                            padding=ft.padding.all(10),
                            border_radius=ft.border_radius.all(5)
                        ),
                        productos_container,
                        ft.Divider(),
                        carrito_container,
                        ft.ElevatedButton("Finalizar compra", on_click=lambda e:
enviar_whatsapp(carrito), bgcolor=ft.colors.BLUE_500),
                        ft.ElevatedButton("Limpiar carrito", on_click=lambda e:
[carrito.clear(), actualizar_carrito()], bgcolor=ft.colors.RED_500),
                    ],
                    alignment=ft.MainAxisAlignment.CENTER,
                    horizontal_alignment=ft.CrossAxisAlignment.CENTER,
                ),
                alignment=ft.alignment.center,
                padding=ft.padding.only(top=100)
            )
        ]
    )
)
actualizar_carrito()

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 los componentes de la interfaz con fondos transparentes
    nombre_input = ft.TextField(
        label="Nombre",

```

```

        width=300,
        bgcolor="#FFFFFFA0",
        border_color="#FFFFFFA0",
        color=ft.colors.BLACK # Cambiar el color del texto del campo a negro
    )
edad_input = ft.TextField(
    label="Edad",
    width=300,
    bgcolor="#FFFFFFA0",
    border_color="#FFFFFFA0",
    color=ft.colors.BLACK # Cambiar el color del texto del campo a negro
)

mensaje_container = ft.Column()

# Guardar datos
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)
        if edad >= 18:
            mostrar_tienda(page) # Mostrar tienda si es mayor de edad
        else:
            mensaje_container.controls.append(ft.Text(f"{nombre} es menor de
edad.", color=ft.colors.BLACK))
            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)
    limpiar_button = ft.ElevatedButton("Limpiar", on_click=lambda e:
[nombre_input.clear(), edad_input.clear(), mensaje_container.controls.clear(),
page.update()], bgcolor=ft.colors.RED_500)

    usuario_input = ft.TextField(
        label="Usuario",
        width=300,
        bgcolor="#FFFFFFA0",
        border_color="#FFFFFFA0",
        color=ft.colors.BLACK
    )
    contrasena_input = ft.TextField(
        label="Contraseña",
        width=300,
        bgcolor="#FFFFFFA0",

```

```

        border_color="#FFFFFFA0",
        color=ft.colors.BLACK,
        password=True # Campo para contraseña
    )

mensaje_login = ft.Column()

def login_click(e):
    mensaje_login.controls.clear()
    usuario = usuario_input.value
    contrasena = contrasena_input.value

    if verificar_credenciales(usuario, contrasena):
        page.controls.clear()
        page.add(
            ft.Stack(
                [
                    ft.Image(
                        src="C:\\Users\\Owner\\OneDrive\\Documents\\SegundaParcial\\fondo.jpg",
                        fit=ft.ImageFit.COVER,
                        opacity=0.8
                    ),
                    ft.Container(
                        content=ft.Column(
                            controls=[
                                ft.Container(
                                    content=ft.Text("Gestión de Usuarios", size=30,
color=ft.colors.BLACK, weight=ft.FontWeight.BOLD),
                                    bgcolor=ft.colors.WHITE,
                                    opacity=0.7,
                                    padding=ft.padding.all(10),
                                    border_radius=ft.border_radius.all(5)
                                ),
                                nombre_input,
                                edad_input,
                                guardar_button,
                                limpiar_button,
                                mensaje_container,
                            ]
                        ),
                        alignment=ft.alignment.center,
                        padding=ft.padding.only(top=100)
                    )
                ]
            )
        )
    else:
        # Si las credenciales son incorrectas, mostramos un mensaje de error
        mensaje_login.controls.append(ft.Text("Usuario o contraseña incorrectos.",
color=ft.colors.RED))
        page.update()

```

```

    login_button = ft.ElevatedButton("Iniciar sesión", on_click=login_click,
bgcolor=ft.colors.GREEN_500, color=ft.colors.WHITE)

# Contenedor principal de contenido para login con el título actualizado
login_content = ft.Column(
    controls=[
        ft.Container(
            content=ft.Text("Iniciar sesión", 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)
        ),
        usuario_input,
        contrasena_input,
        login_button,
        mensaje_login
    ],
    alignment=ft.MainAxisAlignment.CENTER,
    horizontal_alignment=ft.CrossAxisAlignment.CENTER,
)

# Estructura de la página con imagen de fondo y contenido centrado más abajo
page.add(
    ft.Stack(
        [
            ft.Image(
                src="C:\\Users\\Owner\\OneDrive\\Documents\\SegundaParcial\\fondo.j
pg",

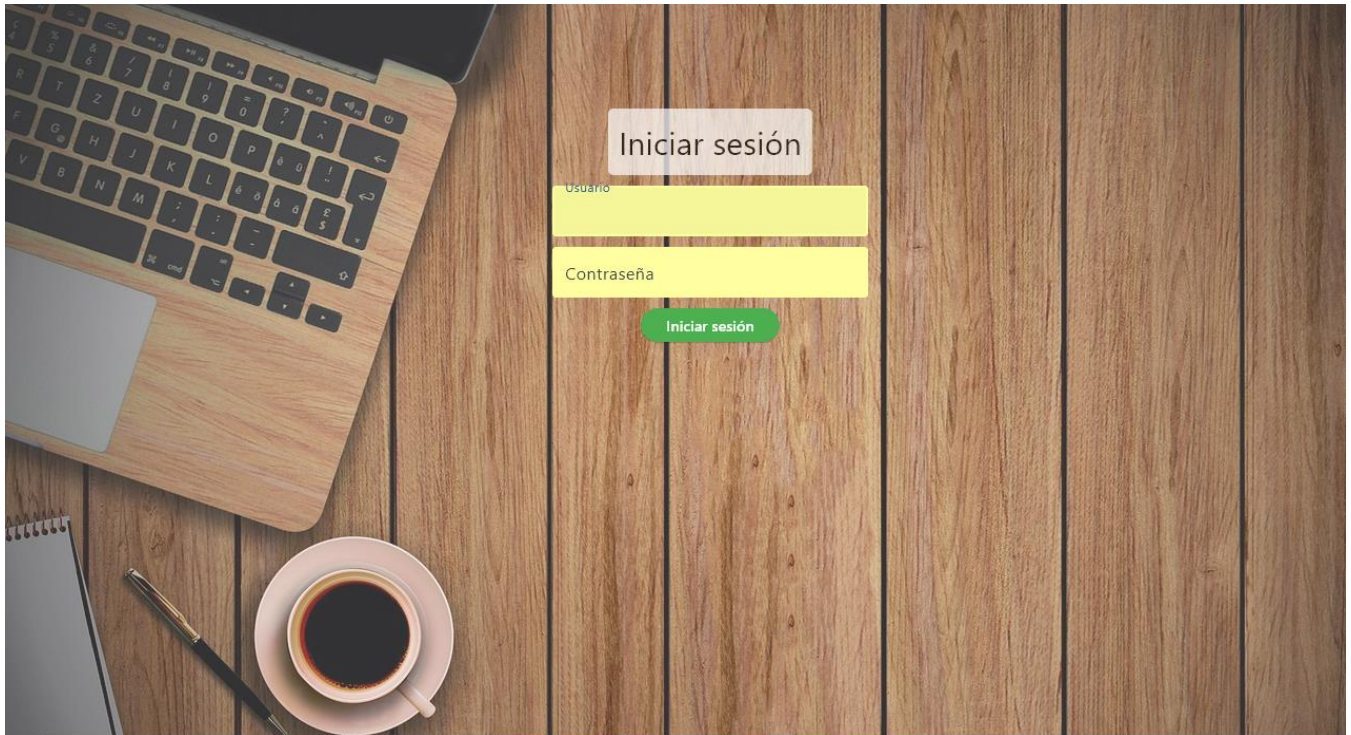
                fit=ft.ImageFit.COVER,
                opacity=0.8
            ),
            ft.Container(
                content=login_content,
                alignment=ft.alignment.center,
                padding=ft.padding.only(top=100)
            )
        ]
    )
)

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

```


Interfaz Login

La interfaz de login permite verificar la edad del usuario para determinar si puede acceder a las funciones avanzadas de la aplicación. Incluye un campo para ingresar la edad, un botón de inicio de sesión, y mensajes de validación en caso de que no se cumpla el requisito de ser mayor de edad.



Interfaz de Gestión de Usuarios

The image shows a top-down view of a wooden desk. On the left, a portion of a laptop with a light-colored keyboard is visible. In the bottom left corner, there is a white cup of coffee on a saucer, a black pen, and a spiral-bound notebook. Overlaid on the right side of the desk is a semi-transparent white rectangular box titled "Gestión de Usuarios". Inside this box, there are two yellow input fields: the first is labeled "Nombre" and the second is labeled "Edad". Below these fields are two buttons: a blue one labeled "Guardar" and a red one labeled "Limpiar".

Gestión de Usuarios

Nombre

Edad

Guardar

Limpiar

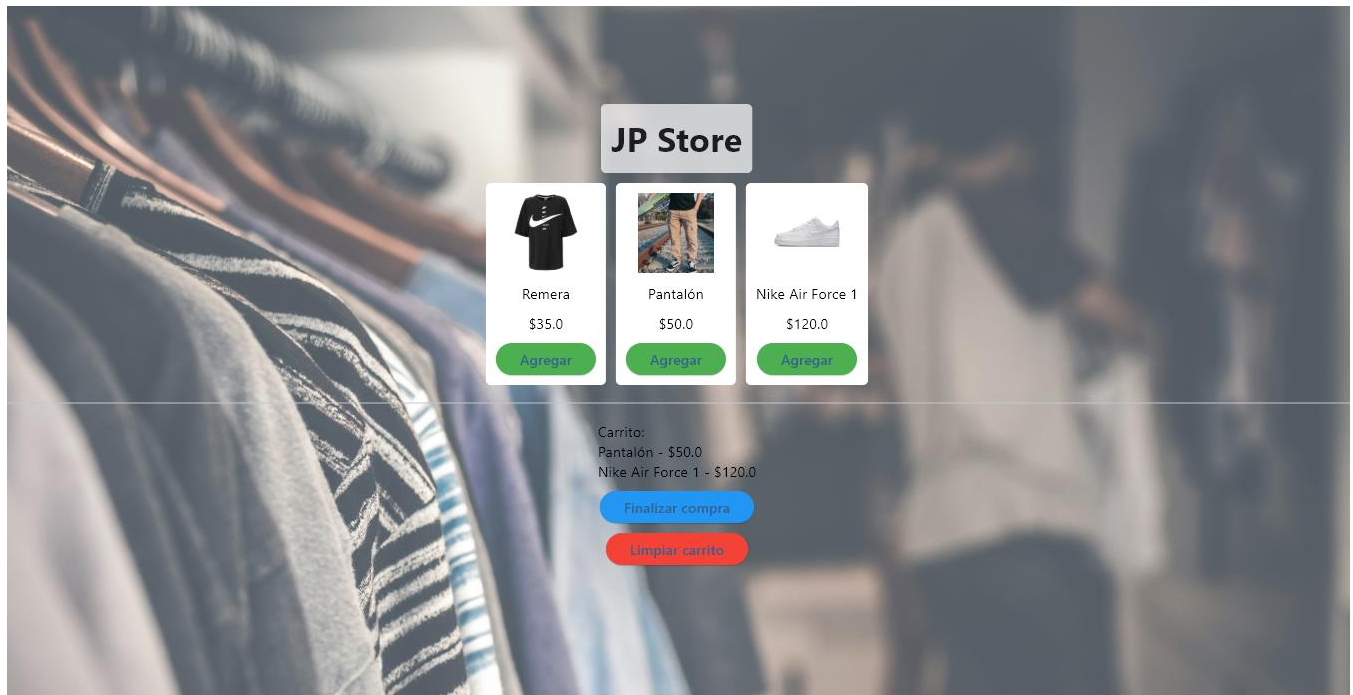
La sección de gestión de usuarios está diseñada para ingresar información básica de usuarios, como el nombre y la edad. Esta interfaz incluye:

- Campos para ingresar el **nombre y la edad** del usuario.
- Dos botones:

Guardar: Al guardar, se registran los datos del usuario.

Limpiar: Borra los campos de entrada para una nueva operación. El sistema verifica automáticamente si el usuario es mayor de edad. Solo aquellos usuarios que cumplen este requisito pueden proceder para ingresar a la tienda.

Interfaz de Tienda de Ropas - JP Store



La tienda JP Store está diseñada para brindar una experiencia de compra personalizada y eficiente con las siguientes funcionalidades:

Exploración de Productos:

- Visualización de los productos disponibles con información relevante: nombre, precio, descripción e imágenes.

Carrito de Compras Personalizado:

- Los usuarios pueden seleccionar productos y agregarlos al carrito para revisar su selección antes de finalizar la compra.

Finalización de Compra a través de WhatsApp:

- Al momento de finalizar una compra, el sistema ofrece la opción de contactar con el proveedor para cerrar la operación directamente por WhatsApp.
- Se facilita una experiencia de compra directa y personalizada a través de mensajería para brindar una comunicación cercana con los usuarios.

Confirmación de Pedido:

- Al elegir finalizar la compra, el usuario envía su solicitud a través de WhatsApp para confirmar detalles como dirección de envío, tallas, método de pago y otros parámetros importantes.

Gestión de Consultas:

- Los usuarios pueden enviar consultas sobre productos, disponibilidad, métodos de envío o cualquier otro tema mediante WhatsApp.

Conclusión

El desarrollo de esta aplicación ha permitido consolidar competencias clave en el ámbito de la programación y el diseño de interfaces gráficas, destacando la integración de bases de datos y la implementación de funcionalidades prácticas y relevantes. La herramienta no solo cumple con los requerimientos de gestión de usuarios y simulación de tienda, sino que también establece un marco adaptable para futuros proyectos más complejos.

Además, la aplicación refleja cómo la tecnología puede simplificar procesos cotidianos y proporcionar experiencias interactivas mediante la integración de plataformas como WhatsApp. Este proyecto representa un paso importante en la formación como ingeniero en informática, orientado a resolver problemas prácticos mediante soluciones tecnológicas innovadoras y funcionales.

LINK DE GITHUB

[JesusPortillo19/Examen-Final](https://github.com/JesusPortillo19/Examen-Final)