



**FACULTAD DE INGENIERIA
INGENIERIA INFORMATICA
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

El desarrollo de aplicaciones modernas exige herramientas que simplifiquen la creación ágil y efectiva de interfaces gráficas de usuario (GUI). En este escenario específico -- Flet emerge como una opción flexible que agiliza el diseño tanto de aplicaciones web como de escritorio al fusionarse naturalmente con Python -- un lenguaje predominantemente elegido por su simplicidad y potencia informática. El proyecto se enfoca en diseñar una aplicación para gestionar la lista de compras donde los usuarios puedan añadir modificar o borrar ítems de manera instintiva y fluidamente.

Usando Flet se creó una interfaz gráfica que responde de forma dinámica a las acciones del usuario: añadir nuevos elementos y editar o eliminar las tareas no necesarias. Las funcionalidades se incorporaron siguiendo prácticas de desarrollo recomendadas para lograr una aplicación funcional que cumple los requisitos establecidos. En este informe se describen los cambios realizados en el código base y se explica cómo contribuyen a mejorar la facilidad de uso de la aplicación.

```

import flet as ft

def main(page: ft.Page):
    page.window_width = 600
    page.window_height = 500
    page.title = "Lista de compras"

    # Crear un campo de texto para la entrada
    new_task = ft.TextField(hint_text="¿Qué necesitas comprar?",
width=300)
    editing_task = None # Variable para saber si estamos editando una
tarea

    # Función para agregar o modificar una tarea
    def add_clicked(e):
        nonlocal editing_task
        if new_task.value: # Asegúrate de que el campo no esté vacío
            if editing_task: # Si estamos editando, actualizamos la
tarea

                editing_task.label = new_task.value
                editing_task = None # Reiniciamos el estado de edición
            else: # Si no, creamos una nueva tarea
                task = ft.Checkbox(label=new_task.value)

                # Crear botones de modificar y eliminar para cada tarea
                modify_button = ft.IconButton(
                    icon=ft.icons.CREATE, # Icono de modificar
                    on_click=lambda e, t=task: modify_task(t)
                )

                delete_button = ft.IconButton(
                    icon=ft.icons.DELETE, # Icono de eliminar
                    on_click=lambda e, t=task: delete_task(task_row)
                )

                # Añadir la tarea y los botones en una fila (Row)
                task_row = ft.Row([task, modify_button, delete_button])
                page.add(task_row)

                # Limpiar el campo de entrada y enfocar nuevamente
                new_task.value = ""
                new_task.focus()
                page.update()

    # Función para modificar una tarea
    def modify_task(task):
        nonlocal editing_task

```

```

        new_task.value = task.label # Pasar el valor de la tarea al
campo de texto
        editing_task = task # Guardar referencia a la tarea que se está
editando
        new_task.focus()
        page.update()

# Función para eliminar una tarea
def delete_task(task_row):
    page.controls.remove(task_row) # Eliminar la fila completa que
contiene la tarea
    page.update()

# Logo y texto del encabezado
logo = ft.Image(src="C:\\Users\\Owner\\OneDrive\\Documents\\TP
PROGRAMACIONV1\\jp.jpg", width=500, height=400)

header_text = ft.Text("Bienvenidos a la App de Lista de Compras",
size=20, weight=ft.FontWeight.BOLD)

# Organizar el encabezado en una columna
header = ft.Column([
    logo,
    header_text
], alignment="center")

# Agregar elementos a la aplicación
page.add(
    header,
    ft.Divider(height=20), # Agrega un divisor
    ft.Row([
        new_task,
        ft.IconButton(icon=ft.icons.ADD, on_click=add_clicked) #
Botón agregar con icono
    ])
)

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

```

Se realizó las siguientes modificaciones:

1. Logo

```
logo = ft.Image(src="C:\\Users\\Owner\\OneDrive\\Documents\\TP  
PROGRAMACIONV1\\jp.jpg", width=500, height=400)
```

- ✓ La imagen se carga desde una ruta local especificada en src.
- ✓ Los atributos **width=500 y height=400** establecen el tamaño del logo en 500 píxeles de ancho y 400 píxeles de alto.

2. Mensaje de bienvenida

```
header_text = ft.Text("Bienvenidos a la App de Lista de Compras",  
size=20, weight=ft.FontWeight.BOLD)
```

- ✓ El parámetro "Bienvenidos a la App de Lista de Compras" es el texto que se mostrará.
- ✓ El atributo **size=20** define el tamaño de la fuente.
- ✓ **weight=ft.FontWeight.BOLD** hace que el texto aparezca en negrita.

3. Ancho de 600px Alto de 400px

```
page.window_width = 600  
page.window_height = 500
```

- ✓ Estas dos líneas de código definen el ancho y alto de la ventana de la aplicación en 600 píxeles de ancho y 500 píxeles de alto

4. Botón de agregar

```
ft.IconButton(icon=ft.icons.ADD, on_click=add_clicked)
```

Este código crea un botón con ícono que permite al usuario agregar elementos a la lista de compras.

- ✓ El ícono **ft.icons.ADD** es un símbolo estándar de "más" (+), representando la acción de agregar.
- ✓ El atributo **on_click=add_clicked** vincula este botón con la función **add_clicked**, que es la encargada de agregar la tarea escrita en el campo de texto.

5. Botón de modificar

```
modify_button = ft.IconButton(  
    icon=ft.icons.CREATE, # Icono de modificar  
    on_click=lambda e, t=task: modify_task(t)
```

Este botón se genera junto a cada tarea agregada. Es un botón con una acción vinculada para editar una tarea ya realizada.

- ✓ El ícono **ft.icons.CREATE** representa un lápiz, indicando que es para modificar o editar.
- ✓ La función **on_click=lambda e, t=task: modify_task(t)** llama a la función **modify_task** cuando se hace clic, permitiendo que el texto de la tarea actual se cargue en el campo de entrada para su modificación.

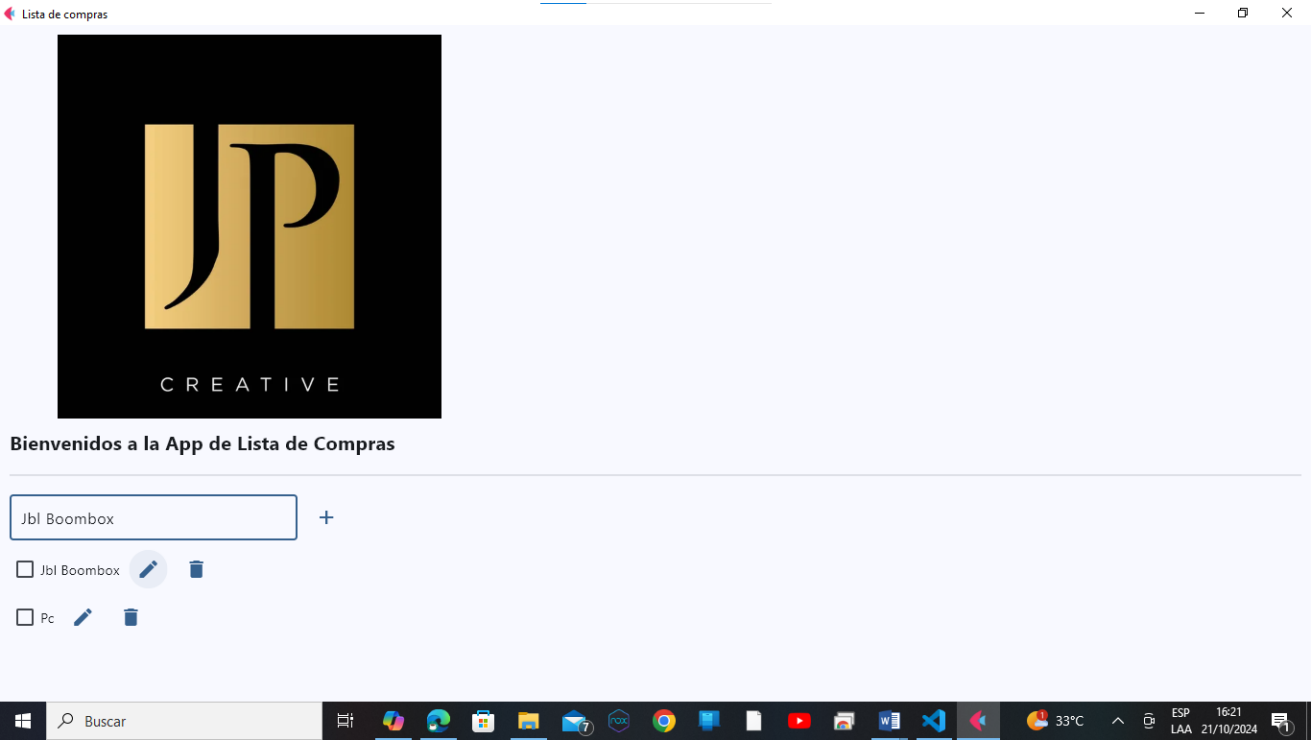
6. Botón de eliminar

```
delete_button = ft.IconButton(  
    icon=ft.icons.DELETE, # Icono de eliminar  
    on_click=lambda e, t=task: delete_task(task_row)  
)
```

- ✓ El ícono **ft.icons.DELETE** es un basurero, simbolizando la acción de eliminar.

- ✓ La función **on_click=lambda e, t=task: delete_task(task_row)** llama a **delete_task**, la cual elimina la fila completa que contiene tanto la tarea como los botones asociados (modificar y eliminar).

Interfaz de la Aplicación



Conclusión

El uso de Flet en conjunto a Python permitió el desarrollo de una aplicación para crear listas de compras que resultó ser funcional y sencilla de utilizar en gran medida. Se logró implementar de forma exitosa las funciones para añadir nuevos elementos a la lista y modificar o eliminar los existentes; garantizando así una experiencia para el usuario sumamente fluida. Además se destacó la flexibilidad que brinda Flet en cuanto al manejo de eventos y la actualización dinámica de la interfaz gráfica.

Este proyecto muestra la viabilidad de combinar la lógica empresarial y el diseño de interfaces sin necesidad de utilizar tecnologías adicionales complejas gracias a Flet. Los conocimientos obtenidos en este proceso son fundamentales para futuros proyectos que necesiten desarrollar aplicaciones interactivas; ofreciendo un método práctico y eficiente para crear software moderno.

LINK DE GITHUB

<https://github.com/JesusPortillo19/Jes-sPortillo-PVI.git>