Documentación: Calculadora TAP con Flet

Este documento detalla el funcionamiento de una interfaz gráfica para una calculadora básica desarrollada en Python utilizando la librería **Flet**. El código actual implementa la visualización, botones numéricos del 1 al 4, un botón de limpieza y la lógica para capturar los clics.

Parte 1: Configuración en Mac

Antes de tocar el código, necesitamos preparar tu computadora. Flet es una librería de Python, por lo que requerimos tener Python instalado.

1. Verificar Python

Abre tu Terminal (puedes buscarla presionando Command + Espacio y escribiendo "Terminal") y escribe:

python3 –version

Si ves algo como Python 3.10.x (o superior), estás listo. Si no, descárgalo desde python.org.

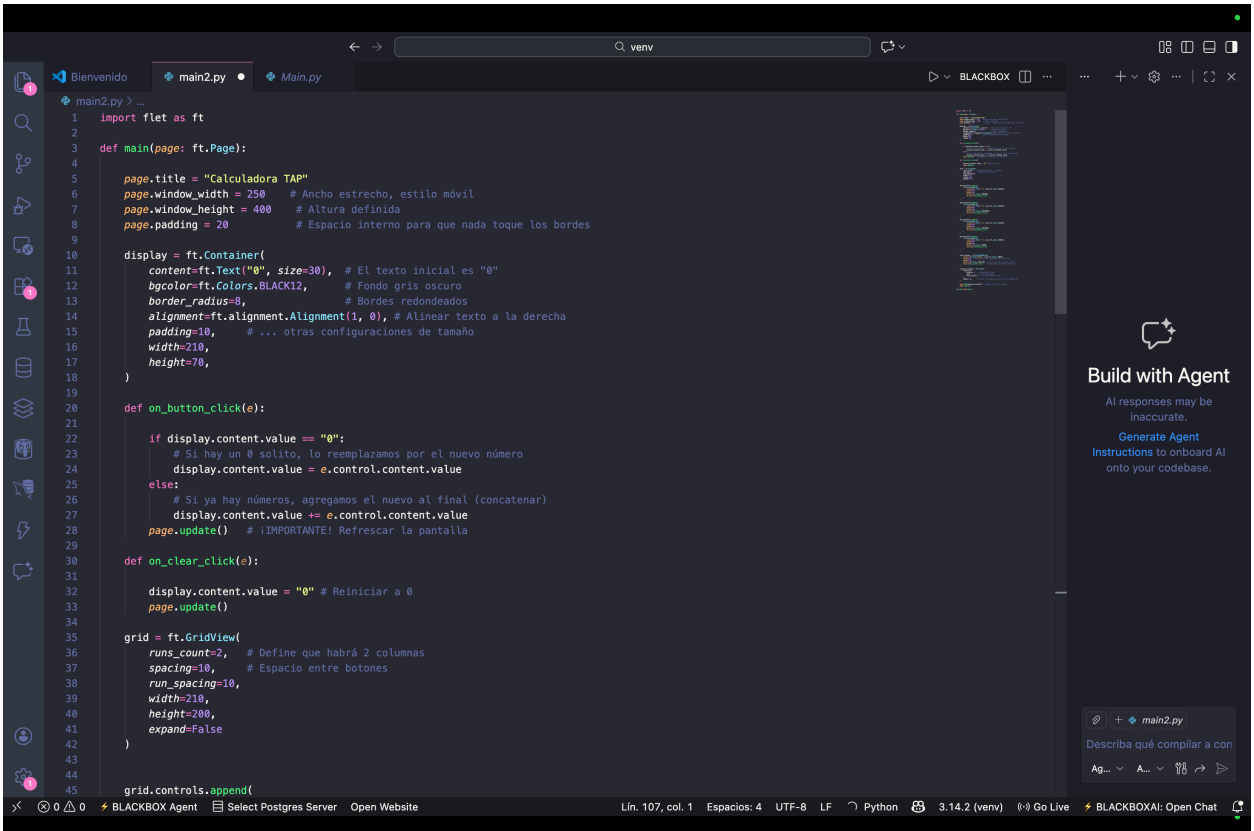
2. Instalar Flet

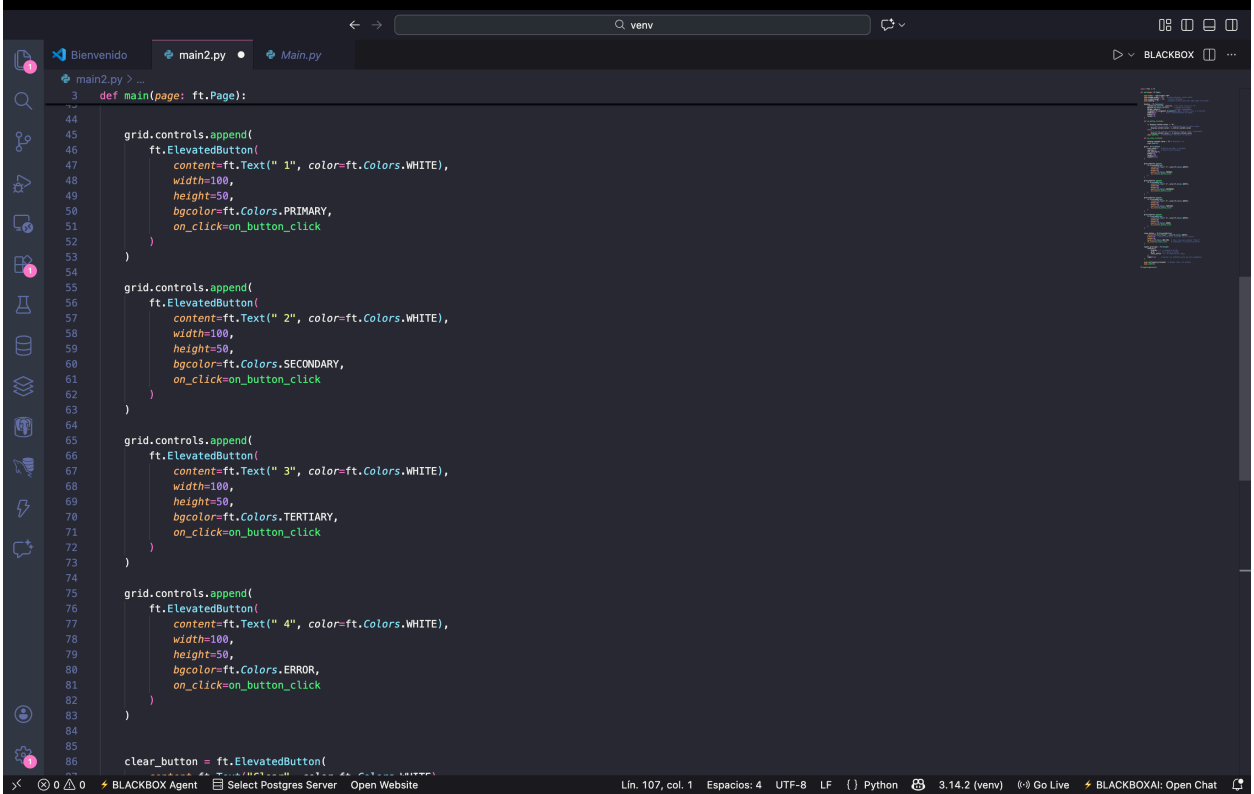
En la misma terminal, ejecuta el siguiente comando para descargar e instalar la librería necesaria:

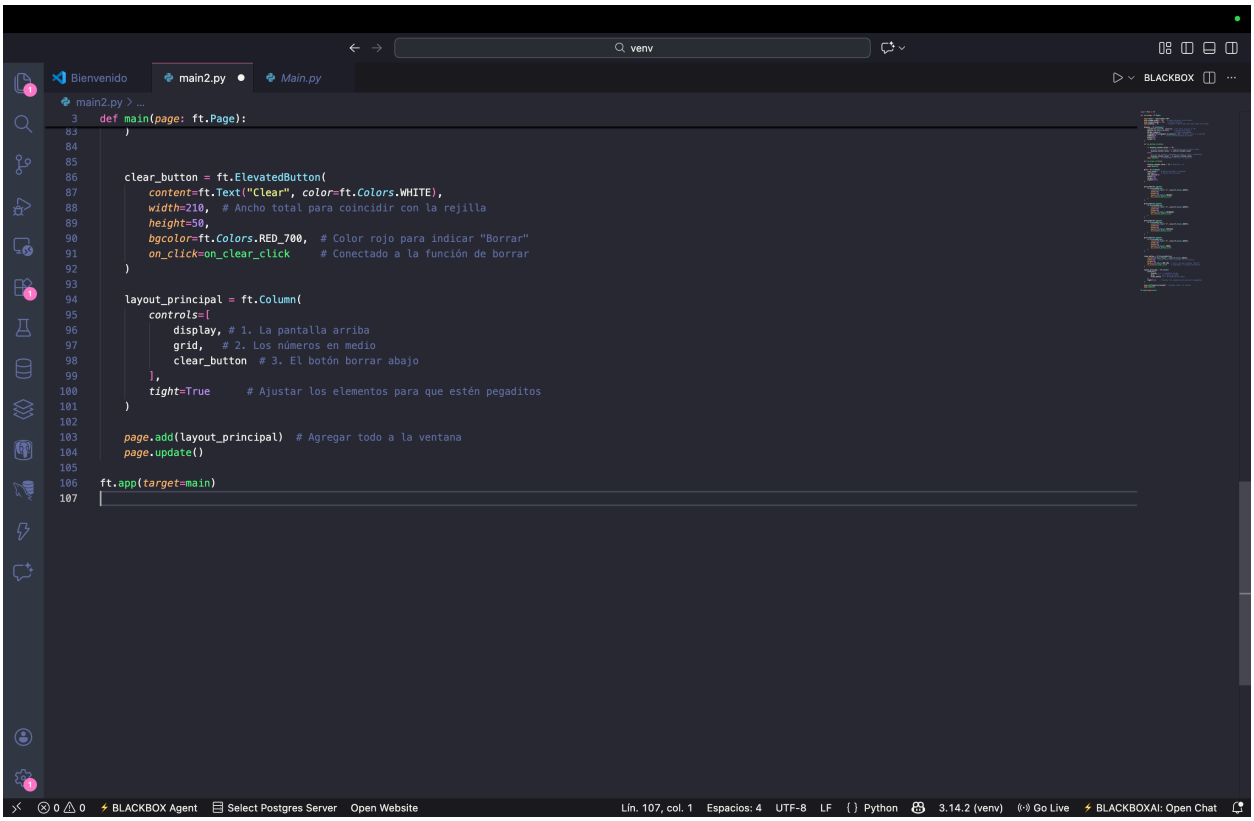
pip3 install flet

Parte 2: Cómo correr el código

1. **Crear el archivo:** Abre tu editor de código favorito (como VS Code) o un editor de texto simple.
2. **Pegar el código:**







3. **Guardar:** Guarda el archivo con el nombre que desees por ejemplo: calculadora.py (es importante que termine en .py).

4. **Ejecutar:** Desde la terminal, navega a la carpeta donde guardaste el archivo y escribe:

python3 calculadora.py

Al hacer esto, se abrirá una ventana nativa en tu Mac con la interfaz de la calculadora.

Parte 3: Explicación del Código

A continuación, desglosamos el código por secciones funcionales para entender qué hace cada bloque.

1. Configuración de la Ventana (main)

Esta sección define cómo se verá la "caja" de la aplicación al abrirse.

- **Explicación:** Estamos configurando la aplicación para que tenga dimensiones fijas, similares a las de un teléfono celular antiguo o una calculadora de bolsillo.

2. La Pantalla de Visualización (display)

Aquí definimos el área donde aparecen los números.

```
    display = ft.Container(
        content=ft.Text("0", size=30),  # El texto inicial es "0"
        bgcolor=ft.Colors.BLACK12,      # Fondo gris oscuro
        border_radius=8,                # Bordes redondeados
        alignment=ft.alignment.Alignment(1, 0), # Alinear texto a la derecha
        padding=10,     # ... otras configuraciones de tamaño
        width=210,
        height=70,
    )
```

- **Concepto clave:** Usamos un Container (una caja) para darle color de fondo y bordes redondeados al texto.
- **Alineación (1, 0):** En Flet, 0,0 es el centro. 1 en el eje X significa "todo a la derecha". Esto es típico en calculadoras (los números entran por la derecha).

3. El "Cerebro" de la Calculadora (Eventos)

Estas son las funciones que deciden qué pasa cuando el usuario toca algo.

A. Función on_button_click (Escribir números)

```
    def on_button_click(e):

        if display.content.value == "0":
            # Si hay un 0 solito, lo reemplazamos por el nuevo número
            display.content.value = e.control.content.value
        else:
            # Si ya hay números, agregamos el nuevo al final (concatenar)
            display.content.value += e.control.content.value
        page.update()   # ¡IMPORTANTE! Refrescar la pantalla
```

- **Lógica:** Esta función evita que escribas "01" o "05". Si la pantalla solo muestra el cero inicial, lo borra y pone el número presionado. Si ya tienes un "5" y presionas "2", los une para formar "52".
- **page.update():** Sin esta línea, los datos cambiarían internamente, pero el usuario no vería el cambio en la pantalla.

B. Función on_clear_click (Borrar todo)

```
30      def on_clear_click(e):
31
32          display.content.value = "0" # Reiniciar a 0
33          page.update()
```

- **Función:** Restaura la calculadora a su estado inicial.

4. La Rejilla de Botones (GridView)

En lugar de colocar botones uno debajo del otro, usamos una rejilla para organizarlos.

```
35      grid = ft.GridView(
36          runs_count=2,   # Define que habrá 2 columnas
37          spacing=10,     # Espacio entre botones
38          run_spacing=10,
39          width=210,
40          height=200,
41          expand=False
42      )
```

- **Estructura:** El código añade 4 botones (1, 2, 3, 4). Como definimos runs_count=2, se acomodarán así automáticamente:

  [1] [2]
  [3] [4]

- **Colores:** Cada botón tiene un color distinto (PRIMARY, SECONDARY, etc.) para diferenciarlos visualmente. Todos están conectados a la función on_button_click.

5. El Botón "Clear"

Este botón está fuera de la rejilla numérica porque ocupa todo el ancho.

```
86      clear_button = ft.ElevatedButton(
87          content=ft.Text("Clear", color=ft.Colors.WHITE),
88          width=210,  # Ancho total para coincidir con la rejilla
89          height=50,
90          bgcolor=ft.Colors.RED_700,  # Color rojo para indicar "Borrar"
91          on_click=on_clear_click     # Conectado a la función de borrar
92      )
```

6. El Ensamblaje Final (layout_principal)

Finalmente, tomamos todas las partes sueltas y las metemos en una columna vertical.

```
94      layout_principal = ft.Column(
95          controls=[
96              display, # 1. La pantalla arriba
97              grid,   # 2. Los números en medio
98              clear_button  # 3. El botón borrar abajo
99          ],
100         tight=True      # Ajustar los elementos para que estén pegaditos
101     )
102
103     page.add(layout_principal)  # Agregar todo a la ventana
104     page.update()
105
106 ft.app(target=main)
```

7. Resultado del código

Cuando ejecutes el código se debe de ver algo así.