



**TECNOLÓGICO
NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO
DE CUAUTLA**

Código ejercicio 1

Gálvez Cadenas Oscar | Tópicos Avanzados de Programación | 16/02/2026
24680111 | Cuarto Semestre - Grupo 1 | Ingeniería en Sistemas Computacionales
Unidad 1 | Profesor. Caballero Alfaro Arístides

Reporte de Funcionamiento: Calculadora TAP (Flet)

El código analizado construye una aplicación de escritorio funcional que emula una calculadora estándar, aplicando el paradigma de **Programación Orientada a Eventos** mediante el framework Flet.

Estructura de la Interfaz (Layout)

El programa utiliza una jerarquía de contenedores para organizar los elementos visuales (Componentes de Control):

- `ft. Page`: Es la ventana principal. Se configura con un tamaño fijo (300x500 px) y se desactiva el cambio de tamaño para mantener la integridad del diseño.
- `ft. Container`: Actúa como la pantalla de la calculadora, con un fondo gris tenue (`BLACK12`) y bordes redondeados.
- `ft. Column` y `ft. Row`: El código utiliza un sistema de rejilla (grid) basado en filas y columnas para alinear los 20 botones de forma simétrica.

Manejo de Estado

Para gestionar qué números se muestran, el código emplea una técnica común en Python:

- `expression = []`: Se usa una lista con un solo string para almacenar la operación actual. Al ser una lista (objeto mutable), permite que la función interna `button_click` modifique su contenido sin necesidad de usar la palabra clave `nonlocal`.

Lógica de Eventos (Event Handling)

El corazón del programa es la función `button_click(e)`, que implementa la teoría de manejo de eventos:

- **Captura de datos**: Cuando se hace clic en cualquier botón, el evento `e` recupera el valor almacenado en la propiedad `data` del botón.
- **Procesamiento condicional**:
 - **Limpieza (C)**: Reinicia la cadena de texto.

- **Cálculo (=):** Utiliza la función `eval()` de Python para resolver la operación matemática. Sustituye el símbolo `^` por `**` para que sea compatible con la sintaxis de Python.
- **Funciones Especiales:** Implementa la raíz cuadrada (`math.sqrt()`) y el cambio de signo (`+/-`).
- **Manejo de Errores:** Utiliza bloques `try/except` para evitar que la aplicación se cierre si el usuario intenta una operación inválida (como dividir por cero), mostrando en su lugar la palabra "Error".

4. Actualización de la GUI

Siguiendo la arquitectura de Flet, los cambios en las variables de Python no se reflejan automáticamente en la pantalla por razones de rendimiento. El código utiliza:

- `update_display()`: Una función personalizada que actualiza el texto del control `display` y ejecuta `page.update()`. Este último comando es vital, ya que envía los cambios realizados en el backend de Python hacia el motor de renderizado (Flutter) para que el usuario vea el resultado.

Resumen de Componentes Técnicos

Concepto	Implementación en el código
Componente de entrada	<code>ft.Button</code> con propiedad <code>on_click</code> .
Componente de salida	<code>ft.Text</code> dentro de un <code>ft.Container</code> .
Distribución	Uso de <code>expand=1</code> en botones para que ocupen el mismo ancho en la fila.
Lógica matemática	Librería <code>math</code> y función <code>eval()</code> .