



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

# Unidad 1

## Interfaz gráfica de usuario.

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

## Tabla de contenido

Interfaz Gráfica de Usuario (GUI) .....	3
Creación de Interfaz Gráfica para Usuarios .....	3
Tipos de Eventos .....	3
Manejo de Eventos .....	4
Manejo de Componentes Gráficos de Control .....	4
Referencias .....	5

## Interfaz Gráfica de Usuario (GUI)

La Interfaz Gráfica de Usuario es el entorno visual que permite la interacción entre el usuario y el sistema mediante elementos como ventanas, botones e iconos . En el enfoque moderno propuesto, se utiliza **Flet**, un framework que permite construir aplicaciones multiplataforma (web, móvil y escritorio) utilizando **Python puro** .

## Creación de Interfaz Gráfica para Usuarios

La creación de interfaces bajo este esquema se basa en una estructura de **objetos** en lugar de diseño web tradicional (HTML/CSS) .

- ❑ **Motor de Renderizado:** Flet utiliza **Flutter** para dibujar los componentes, lo que garantiza que la interfaz sea visualmente atractiva y consistente en diferentes sistemas operativos como iOS, Android, Windows, Linux y macOS .
- ❑ **Base de Código Única:** Se escribe un solo código en Python que se adapta automáticamente a diferentes tamaños de pantalla y plataformas .
- ❑ **Sin Barrera de Frontend:** No se requiere conocimiento previo de lenguajes como Dart, JavaScript o HTML

## Tipos de Eventos

Los eventos son acciones detectadas por el sistema que ocurren sobre la interfaz. En el desarrollo con Flet y Python, los tipos de eventos comunes incluyen:

- ❑ **Interacciones de clic:** Como las generadas por botones ( `ElevatedButton` , `IconButton` ).
- ❑ **Cambios de estado:** Ajustes en controles de selección o deslizamiento (ej. mover un control de volumen o seleccionar un color) .
- ❑ **Eventos de ciclo de vida:** Acciones que ocurren al cargar la página o actualizar la aplicación .

## Manejo de Eventos

El manejo de eventos consiste en definir qué código se ejecutará cuando ocurra una acción específica.

- **Funciones Estándar:** La lógica de eventos se gestiona directamente con funciones estándar de Python o funciones lambda .
- **Ejemplo de Sintaxis:** Para un botón de actualización, se utiliza la propiedad `on_click` vinculada a una función de actualización de la página:
  - `ft.ElevatedButton("Actualizar", on_click=lambda e: page.update())` .
- **Simplificación:** Esto reduce drásticamente la curva de aprendizaje al no requerir sintaxis compleja de otros lenguajes para la gestión de interactividad .

## Manejo de Componentes Gráficos de Control

Los componentes de control (o widgets) son los bloques de construcción de la interfaz.

- **Controles Integrados:** Existen más de **150 controles listos para usar** , basados en los lenguajes de diseño **Material Design** y **Cupertino** (Apple).
- **Ejemplos de Componentes:**
  - **Contenedores y Filas:** Para organizar el diseño ( `ft.Container` , `ft.Row` ).
  - **Entrada y Selección:** Deslizadores de volumen ( `volume` ), menús desplegables (`Seleccionar color`) y botones de acción flotantes .
  - **Visualización de Datos:** Gráficos de líneas ( `ft.LineChart` ) y tablas de datos para reportes .
- **Foco en la Lógica:** Al usar componentes pre -diseñados, el estudiante de ingeniería puede enfocarse en la lógica de negocio y los algoritmos del backend en lugar de los detalles estéticos del diseño visual .

## Referencias

1. Tecnológico Nacional de México. (2010). Programa de estudio de la asignatura: Tópicos Avanzados de Programación. Clave: SCD -1027. Secretaría de Educación Pública. [Este es el documento base que define los temas 1.1 al 1.4]
2. Flet Dev. (s.f.). Introduction to Flet: User interface. Events and event handlers. Recuperado el 22 de mayo de 2024, de <https://flet.dev/docs/> [Fuente principal para el manejo de componentes y eventos en Python].
3. Guzmán, M. A. (2023). Desarrollo de interfaces gráficas con Python y arquitecturas multiplataforma. Editorial Universitaria.
4. Summerfield, M. (2018). Python GUI Programming: A complete guide to building desktop applications. Addison-Wesley.
5. Google. (2023). Material Design 3: The guide to adaptive design and components. Material.io. <https://m3.material.io/> [Referencia esencial para el punto 1.4 sobre el manejo de componentes gráficos].