

# Interfaces Gráficas (GUI) con Tkinter

## Clase 1: "Hola, Mundo" Gráfico

Prof. D.Sc. BARSEKH-ONJI Aboud

Facultad de Ingeniería  
Universidad Anáhuac México

18 de noviembre de 2025

# Agenda

1 Fundamentos de GUI y Tkinter

2 Nuestro "Hola, Mundo" Gráfico

3 Tarea 1

# ¿Qué es una Aplicación Gráfica (GUI)?

## Programa de Consola (Lo que conocemos)

- Se ejecuta en una terminal.
- La interacción es solo texto.
- Flujo lineal: `input()` (espera),  
`procesa, print()` (muestra).

## Aplicación Gráfica (GUI)

- **Graphical User Interface.**
- Se ejecuta en una **ventana**.
- La interacción es visual: botones,  
menús, clics del mouse.
- **Dirigida por eventos:** El programa  
no tiene un flujo lineal, sino que  
**reacciona** a las acciones del usuario.

# ¿Qué es una Aplicación Gráfica (GUI)?

## Nuestra Herramienta: Tkinter

Tkinter es la biblioteca **estándar** de Python para crear interfaces gráficas. ¡Viene incluida, no hay que instalar nada!

# El "Formulario" o Ventana Principal

## El Lienzo de Nuestra Aplicación

Toda aplicación gráfica necesita una **ventana raíz** o principal (también llamada "formulario"). Es el contenedor donde vivirá todo lo demás (botones, texto, etc.).

## El Bucle de Eventos

A diferencia de un script normal, una GUI debe **mantenerse abierta** esperando a que el usuario haga algo (un clic, escribir, etc.).

- Usamos `ventana.mainloop()` al final.
- Este comando inicia el "bucle de eventos" que mantiene la ventana visible y receptiva.

# El "Formulario" o Ventana Principal

## El Código Mínimo para una Ventana

```
1 import tkinter as tk
2
3 # 1. Crear la ventana principal (formulario)
4 ventana = tk.Tk()
5
6 # (Aqui iria el contenido de la ventana)
7
8 # 2. Iniciar el bucle de eventos
9 ventana.mainloop()
10
11 # El programa se pausara aqui hasta que
12 # el usuario cierre la ventana.
13
```

# Agenda

1 Fundamentos de GUI y Tkinter

2 Nuestro "Hola, Mundo" Gráfico

3 Tarea 1

# Objetos o "Widgets"

## ¿Qué es un Widget?

Un "widget"(o control) es cualquier **objeto** o elemento visual que ponemos dentro de nuestra ventana para interactuar con el usuario.

# Objetos o "Widgets"

## Widget: tk.Label

Una **Etiqueta**. Se usa simplemente para mostrar texto (o imágenes) al usuario. No es interactiva.

```
1  
2 etiqueta = tk.Label(ventana, text="Hola  
    Mundo")  
3
```

## Examples

Widget: tk.Button Un **Botón**. El usuario puede hacer clic en él para disparar una acción o **evento**.

```
1 # Sintaxis:  
2 # Button(donde_vive, propiedades...)  
3 boton = tk.Button(ventana, text="Haz  
    Clic")  
4
```

# Crear y Posicionar: El Método .pack()

## Un Proceso de Dos Pasos

Para que un widget aparezca, siempre debemos seguir dos pasos:

- 1** **Crear** el widget (como vimos: `tk.Label(...)`).
- 2** **Posicionarlo** en la ventana.

# Crear y Posicionar: El Método .pack()

## El Posicionador .pack()

.pack() es el administrador de geometría más simple. Simplemente "empaqueta" los widgets uno debajo del otro, en el orden en que los llamas.

```
1 import tkinter as tk
2 ventana = tk.Tk()
3
4 # 1. Crear la etiqueta
5 etiqueta = tk.Label(ventana, text="¡Hola, Tkinter!")
6 # 2. Posicionar la etiqueta
7 etiqueta.pack()
8 # 1. Crear el boton
9 boton = tk.Button(ventana, text="Soy un boton")
10 # 2. Posicionar el boton
11 boton.pack()
12 ventana.mainloop()
13
```



# Agenda

1 Fundamentos de GUI y Tkinter

2 Nuestro "Hola, Mundo" Gráfico

3 Tarea 1

# Tarea 1: Tu Primera Ventana

## Objetivo

Crear tu primera aplicación gráfica funcional, aplicando los conceptos de **ventana, widgets y posicionamiento**.

# Tarea 1: Tu Primera Ventana

## Instrucciones

Escribe un programa de Python que haga lo siguiente:

- 1 Importa la biblioteca tkinter (usa el alias tk).
- 2 Crea la ventana principal (formulario) llamada ventana.
- 3 **(Nuevo)** Dale un título a tu ventana usando: `ventana.title("Mi Aplicación")`
- 4 Crea un `tk.Label` que viva en ventana y muestre el texto: "¡Bienvenido a mi primera GUI!".
- 5 Crea un `tk.Button` que viva en ventana y muestre el texto: "No hago nada (aún)".
- 6 Usa el método `.pack()` en la etiqueta y luego en el botón para que aparezcan.

