**DESCRIPCION**

Este programa en Python crea una aplicación de restaurante con interfaz gráfica usando la biblioteca tkinter. A continuación te explico para qué sirve y cómo funciona cada parte del código**:**

**Objetivo del Programa**

La aplicación permite a los usuarios:

* Visualizar un menú de comida organizado por categorías (como SANCOCHOS, EJECUTIVOS, etc.).
* Agregar platos al carrito de compras al hacer clic en los botones del menú.
* Ver los productos seleccionados y el total a pagar.
* Finalmente, confirmar el pedido, lo cual muestra un resumen de los productos seleccionados**.**

**¿Qué librería se utiliza?**

La librería que se utiliza es tkinter

**¿Cómo funciona tkinter?**

tkinter es la biblioteca estándar de Python para crear interfaces gráficas (ventanas, botones, menús, etc.).  
Funciona creando **una ventana principal** y luego agregando **componentes visuales** como:

* Frame: contenedor para organizar otros elementos
* Label: mostrar texto
* Button: botones interactivos
* Listbox: listas de elementos
* messagebox: ventanas emergentes con mensajes

La ventana se mantiene abierta con un bucle principal:

ventana.mainloop()

**¿Cómo se utilizó la librería en el proyecto?**

1. **Se creó la ventana principal** con:

frame\_menu = tk. Frame(ventana)

frame\_menu. pack(side="left")

1. **Se organizaron los componentes** usando Frame:

frame\_menu = tk. Frame(ventana)

frame\_menu. pack(side="left")

1. **Se usaron etiquetas (Label) y botones (Button)** **para mostrar el menú:**

tk. Label (frame\_menu, text="MENU"). pack ()

tk. Button (frame\_menu, text="Costilla - $18000", command=...). pack ()

**4.Se mostró el carrito** con Listbox y se actualizó el total con Label.

**5. Se usó messagebox** para mostrar un mensaje cuando se confirma el pedido:

messagebox. showinfo("Pedido", "Resumen del pedido")

**6.Se ejecutó todo con el bucle final**:

ventana.mainloop()

1. **Datos del Menú**

* Se define un **diccionario menú** con categorías (como "SANCOCHOS", "EJECUTIVOS") y listas de platos (tuplas con el nombre y el precio).
* Ejemplo: ("Costilla", 18000)

2. **Carrito de Compras**

carrito = []

Lista vacía donde se irán agregando los productos que el usuario seleccione.

3. **Función agregar\_al\_carrito (nombre, precio)**

def agregar\_al\_carrito (nombre, precio):

carrito. append ((nombre, precio))

actualizar\_carrito ()

* Añade un producto al carrito.
* Llama a actualizar\_carrito () para refrescar la vista del carrito.

1. **Función actualizar\_carrito ()**

def actualizar\_carrito ():

carrito\_list. delete (0, tk.END)

total = 0

for item in carrito:

carrito\_list.insert(tk.END, f"{item[0]} - ${item[1]:,.0f}")

total += item[1]

total\_label.config(text=f"Total: ${total:,.0f}")

 Limpia y vuelve a llenar la Listbox del carrito.

 Calcula el total de la compra y lo muestra.

1. **Función confirmar\_pedido()**

def confirmar\_pedido():

if carrito:

...

else:

...

* Si hay productos en el carrito, muestra un resumen en un **mensaje emergente**.
* Si no hay productos, lanza una advertencia.

1. **Interfaz gráfica con Tkinter**

ventana = tk.Tk()

ventana.title("App de Restaurante")

Se crea la ventana principal y se le pone un título.

1. **Estructura de la ventana**

frame\_menu = tk.Frame(ventana)

frame\_menu.pack(side="left", ...)

frame\_carrito = tk.Frame(ventana)

frame\_carrito.pack(side="right", ...)

Se crean dos áreas:

* Izquierda: Menú de productos.
* Derecha: Carrito de compras.

1. **Mostrar el Menú**

for categoria, productos in menu.items():

tk.Label(...).pack()

for nombre, precio in productos:

b = tk.Button(..., command=...)

b.pack()

Se recorre el menú por categoría.

Se crea un botón por cada producto.

Al hacer clic, se llama a agregar\_al\_carrito()