



# UEA

UNIVERSIDAD

ESTATAL AMAZÓNICA

**UNIVERSIDAD ESTATAL AMAZÓNICA**



## **INGENIERÍA EN TECNOLOGÍAS DE INFORMACIÓN**

### **PROGRAMACIÓN ORIENTADA A OBJETOS**

#### **TEMA:**

**APLICACIÓN GUI DE LISTA DE TAREAS**

#### **ESTUDIANTE:**

**EDWIN FABIÁN NORIEGA BALDEON**

#### **DOCENTE:**

**ING. SANTIAGO ISRAEL NOGALES GUERRERO**

#### **NIVEL:**

**SEGUNDO NIVEL\_PARALELO “A”**

**2025-2025**

**PUYO-ECUADOR**



# UEA

## UNIVERSIDAD ESTATAL AMAZÓNICA

### 1. CÓDIGO DE GITHUB

[https://github.com/EDFANOBA19/PROGRAMACION\\_ORIENTADA\\_A\\_OBJETOS.git](https://github.com/EDFANOBA19/PROGRAMACION_ORIENTADA_A_OBJETOS.git)

### 2. REPOSITORIO DE GITHUB

github.com/EDFANOBA19/PROGRAMACION\_ORIENTADA\_A\_OBJETOS

main 1 Branch 0 Tags

Go to file Add file <> Code About

EDFANOBA Agregar aplicación GUI de lista de tareas con Tkinter 3bf0a6d · 2 minutos ago 42 Commits

.idea feat: Añadir clase Cliente con atributos y método para mostr... 3 months ago

CAPTURAS SEMANA 8 Implementación completa del sistema de gestión de inventa... last month

EJERCICIO EN CLASE SEMANA 07 Implementación de ejercicios de POO en Python: 2 months ago

SEMANA\_10\_POO\_Sistema de Gestión de Inve... Implementación completa del Sistema de Gestión de Invent... last month

SEMANA\_11\_POO\_Sistema Avanzado de Gest... feat: Implementación completa de sistema de inventario en ... last month

SEMANA\_12\_POO\_Sistema de Gestión de Bibli... Implementación completa del Sistema de Gestión de Bibliot... 3 weeks ago

SEMANA\_13\_POO\_Conceptos fundamentales ... git add . 2 weeks ago

SEMANA\_14\_POO\_Creación de una Aplicación ... feat: Implementar aplicación de agenda personal con Tkinter last week

SEMANA\_15\_POO\_Aplicación GUI de Lista de T... Agregar aplicación GUI de lista de tareas con Tkinter 2 minutos ago

SEMANA\_2\_POO\_PILARES DE PROGRAMACIO... Add files via upload 3 months ago

SEMANA\_3\_POO\_Comparación de Programaci... Add files via upload 3 months ago

SEMANA\_4\_POO\_EJEMPLOMUNDOREAL Merge remote-tracking branch 'origin/main' 2 months ago

No description, website, or topics provided

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

EDFANOBA

EDFANOBA19

EDFANOBA19 / PROGRAMACION\_ORIENTADA\_A\_OBJETOS

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main

Go to file

> .idea

> CAPTURAS SEMANA 8

> EJERCICIO EN CLASE SEMANA 07

> SEMANA\_10\_POO\_Sistema de Ge...

> SEMANA\_11\_POO\_Sistema Avanz...

> SEMANA\_12\_POO\_Sistema de Ge...

> SEMANA\_13\_POO\_Conceptos fu...

> SEMANA\_14\_POO\_Creación de una Aplicación GUI de Lista de Tareas

> SEMANA\_15\_POO\_Aplicación GUI de Lista de Tareas

> SEMANA\_2\_POO\_PILARES DE PR...

> SEMANA\_3\_POO\_Comparación d...

> SEMANA\_4\_POO\_EJEMPLOMUN...

> SEMANA\_5\_POO\_Tipos de datos,...

> SEMANA\_6\_POO\_Clasas, objetos,...

PROGRAMACION\_ORIENTADA\_A\_OBJETOS / SEMANA\_15\_POO\_Aplicación GUI de Lista de Tareas / Aplicación GUI de Lista de Tareas

EDFANOBA Agregar aplicación GUI de lista de tareas con Tkinter

Code Blame 107 lines (96 loc) · 5.74 KB

```
1 import tkinter as tk # Importa la biblioteca Tkinter para interfaz gráfica
2 from tkinter import messagebox # Importa el módulo para cuadros de diálogo
3
4 class ListaTareasApp:
5     def __init__(self, root):
6         """
7         Inicializa la ventana principal y todos los widgets de la aplicación.
8         """
9         self.root = root # Guarda la referencia a la ventana principal
10        self.root.title("Lista de Tareas de Edwin Noriega") # Establece el título de la ventana
11
12        # Campo Entry para que el usuario escriba nuevas tareas
13        self.entry = tk.Entry(root, width=40)
14        # Ubica el Entry en la primera fila y primera columna con espacio exterior
15        self.entry.grid(row=0, column=0, padx=10, pady=10)
16        # Vincula la tecla Enter para agregar tareas al presionar mientras está en el Entry
17        self.entry.bind("<Return>", self.agregar_tarea)
18
19        # Botón para añadir tarea, que ejecuta la función agregar_tarea
20        self.btn_add = tk.Button(root, text="Añadir Tarea", command=self.agregar_tarea)
21        # Posición del botón junto al Entry con un pequeño hueco horizontal
22        self.btn_add.grid(row=0, column=1, padx=5)
```



### 3. DESARROLLO DE CÓDIGO CON SUS DIFERENTES PEDIDOS Y

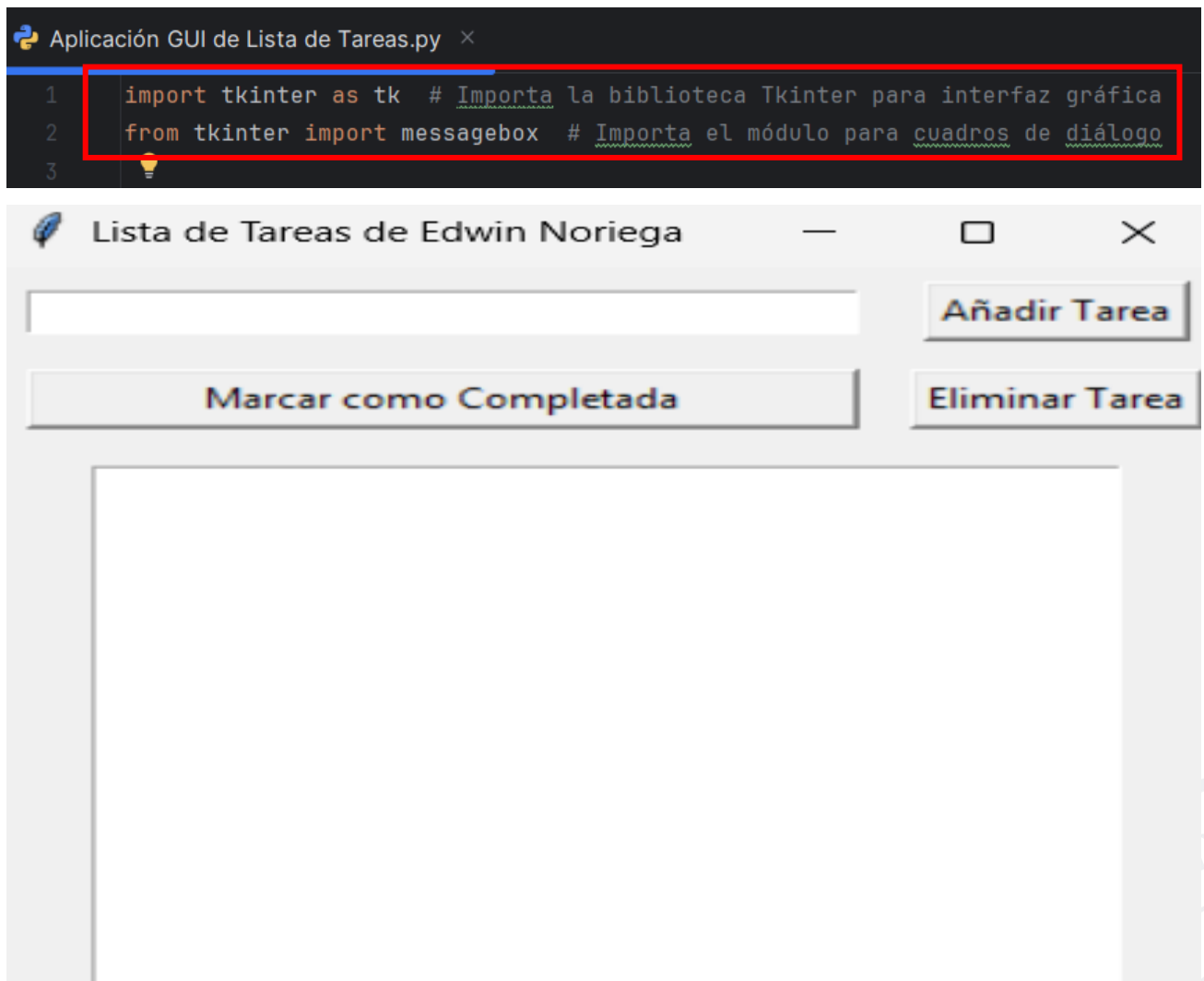
#### REQUISITOS:

**Objetivo:** Desarrollar una aplicación GUI simple para gestionar una lista de tareas, permitiendo al usuario añadir nuevas tareas, marcarlas como completadas y eliminarlas. La aplicación deberá responder adecuadamente a los eventos del usuario, como clics del ratón y pulsaciones del teclado.

#### Requisitos:

##### 1. Interfaz Gráfica:

- Utilizar Tkinter para crear la interfaz de usuario.



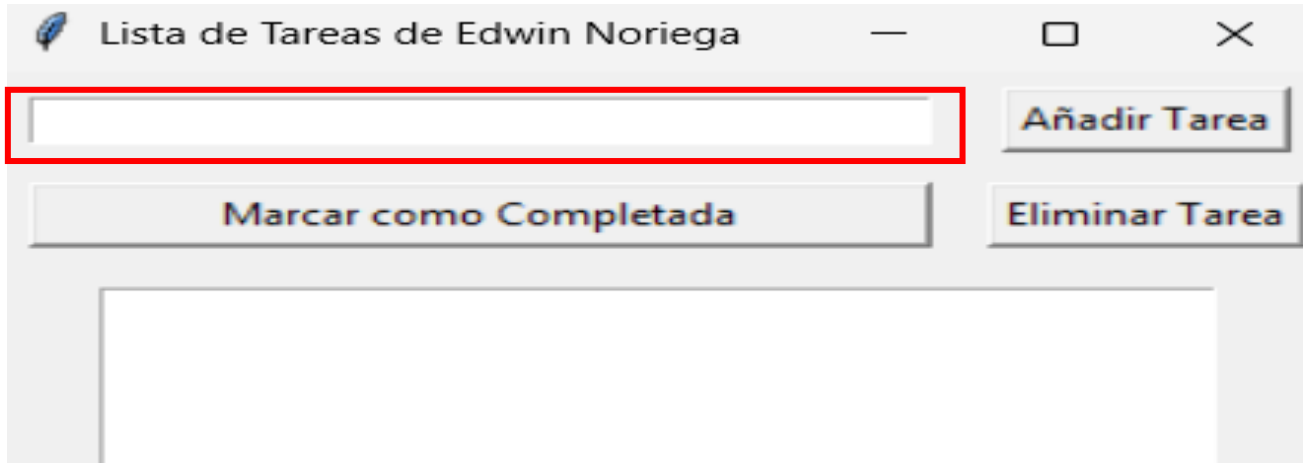
- La aplicación debe tener un campo de entrada (Entry) para escribir nuevas tareas.



# UEA

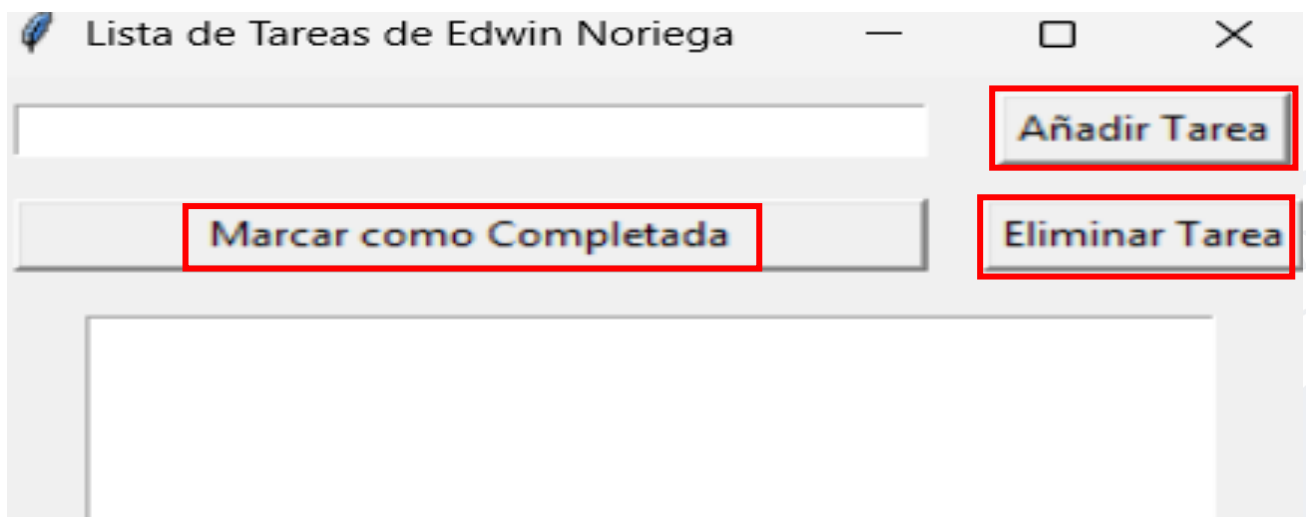
## UNIVERSIDAD ESTATAL AMAZÓNICA

```
12 # Campo Entry para que el usuario escriba nuevas tareas
13 self.entry = tk.Entry(root, width=40)
14 # Ubica el Entry en la primera fila y primera columna con espacio exterior
15 self.entry.grid(row=0, column=0, padx=10, pady=10)
16 # Vincula la tecla Enter para agregar tareas al presionar mientras está en el Entry
17 self.entry.bind("<Return>", self.agregar_tarea)
```



- Incluir botones para "Añadir Tarea", "Marcar como Completada" y "Eliminar Tarea".

```
19 # Botón para añadir tarea, que ejecuta la función agregar_tarea
20 self.btn_add = tk.Button(root, text="Añadir Tarea", command=self.agregar_tarea)
21 # Posición del botón junto al Entry con un pequeño hueco horizontal
22 self.btn_add.grid(row=0, column=1, padx=5)
23
24 # Botón para marcar una tarea seleccionada como completada
25 self.btn_complete = tk.Button(root, text="Marcar como Completada", command=self.completar_tarea)
26 # Se coloca en la segunda fila, primera columna, y ocupa todo el ancho horizontal de la celda
27 self.btn_complete.grid(row=1, column=0, padx=10, pady=5, sticky="ew")
28
29 # Botón para eliminar la tarea seleccionada
30 self.btn_delete = tk.Button(root, text="Eliminar Tarea", command=self.eliminar_tarea)
31 # Ubicación: segunda fila, segunda columna, también con estiramiento horizontal
32 self.btn_delete.grid(row=1, column=1, padx=5, pady=5, sticky="ew")
```





# UEA

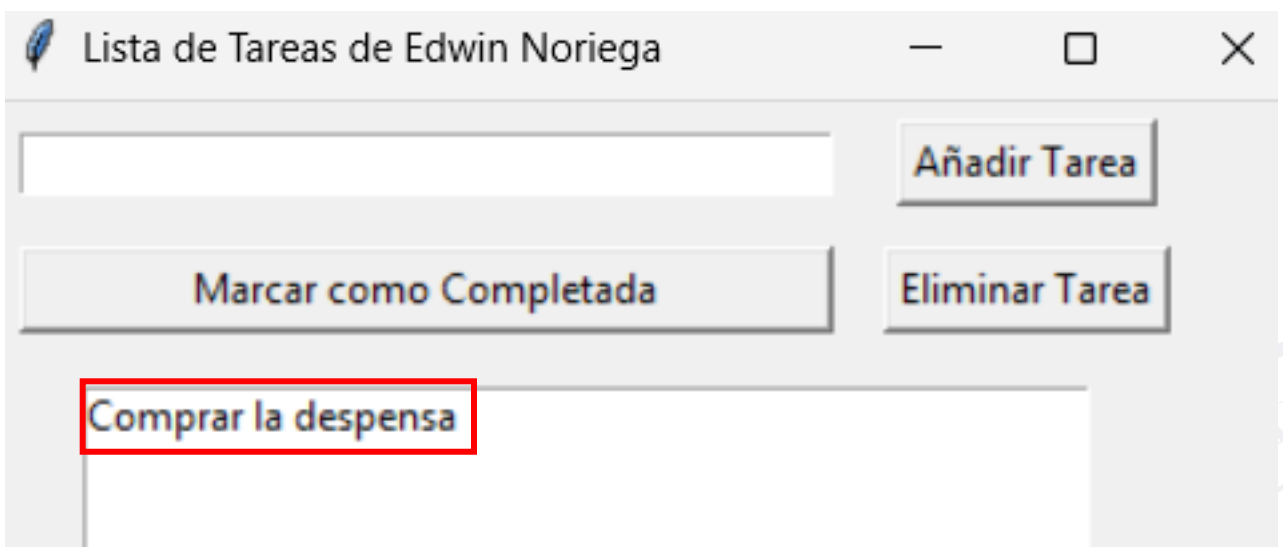
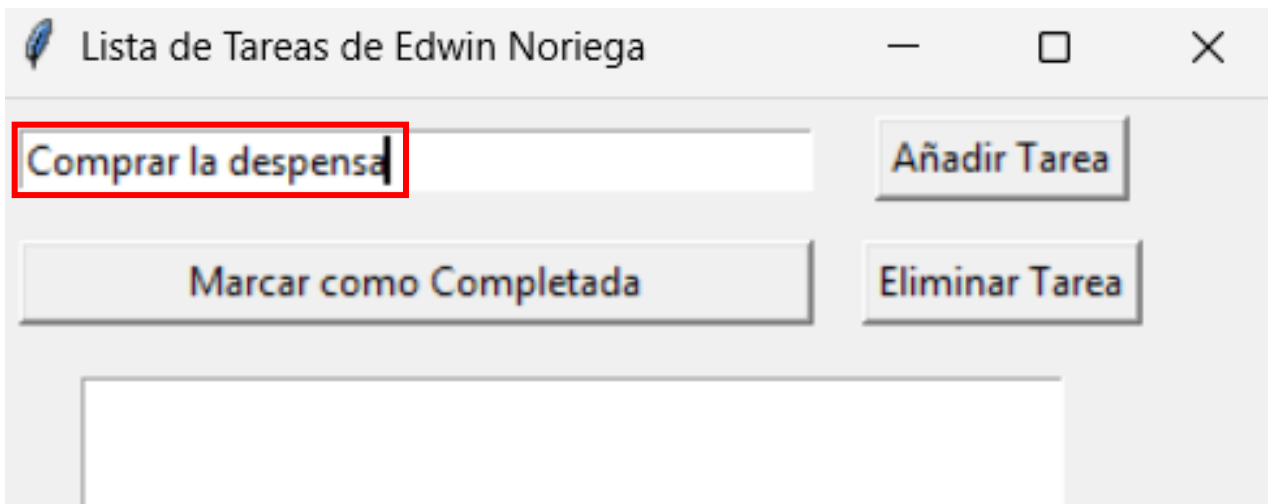
## UNIVERSIDAD ESTATAL AMAZÓNICA

- Mostrar las tareas actuales en un componente de lista (p. ej., Listbox o Treeview).

```
34 # Listbox para mostrar la lista visible de tareas con selección simple
35 self.lista = tk.Listbox(root, width=50, height=15, selectmode=tk.SINGLE)
36 # Se extiende a lo ancho ocupando dos columnas en la tercera fila
37 self.lista.grid(row=2, column=0, columnspan=2, padx=10, pady=10)
38 # Asigna evento para que al hacer doble clic en una tarea se marque como completada
39 self.lista.bind("<Double-1>", self.completar_tarea)
40
```

## 2. Manejo de Eventos:

- Permitir la adición de tareas presionando la tecla Enter después de escribir una tarea.





# UEA

## UNIVERSIDAD ESTATAL AMAZÓNICA

- Opcional: Implementar eventos adicionales de tu elección para mejorar la funcionalidad (p. ej., doble clic en una tarea para marcarla como completada).

### 3. Lógica de la Aplicación:

- Las tareas añadidas deben aparecer en la lista de tareas.



Lista de Tareas de Edwin Noriega

Cobrar el arriendo

Añadir Tarea

Marcar como Completada

Eliminar Tarea

- ✓ Comprar la despensa
- Asistir a la cita medica
- Retirar las correspondenciales
- Asistir a tutotias de POO
- Dar de comer a los perros
- Comprar la Isita de utiles

- Al marcar una tarea como completada, cambia su estado visualmente para reflejar su nuevo estado.

Lista de Tareas de Edwin Noriega

Añadir Tarea

Marcar como Completada

Eliminar Tarea

- ✓ Comprar la despensa
- ✓ Asistir a la cita medica
- ✓ Retirar las correspondenciales
- ✓ Asistir a tutotias de POO
- Dar de comer a los perros
- Comprar la Isita de utiles
- Cobrar el arriendo

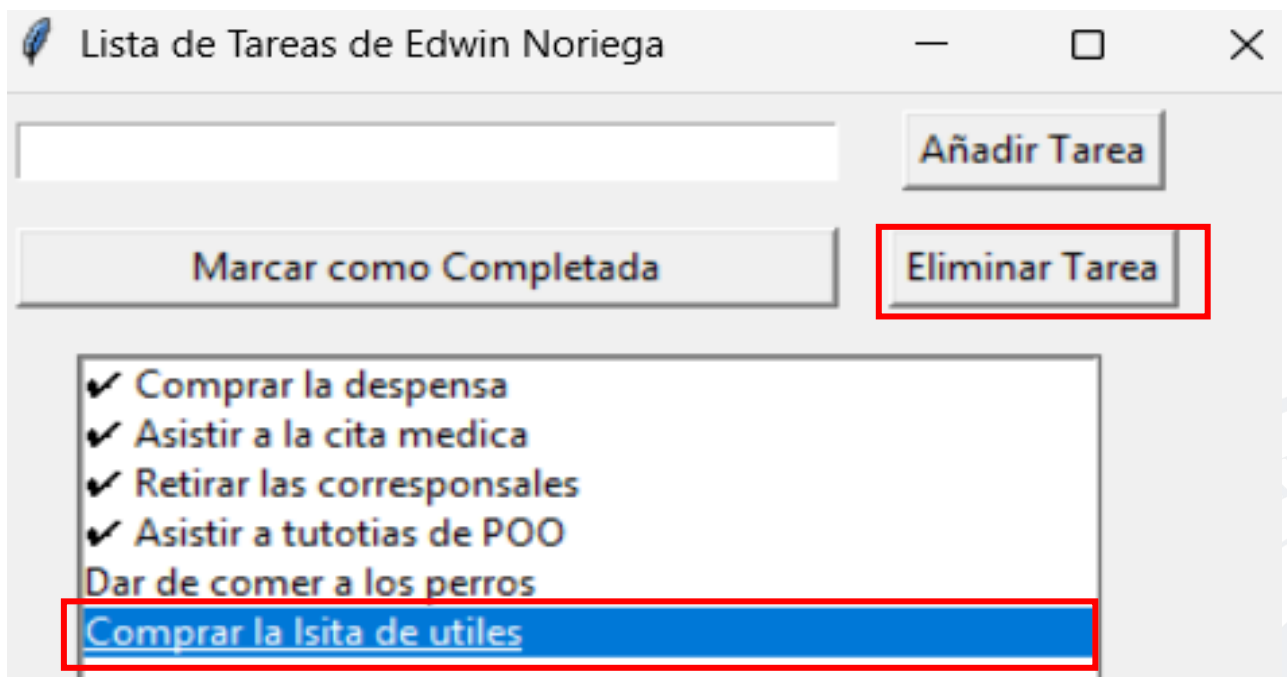
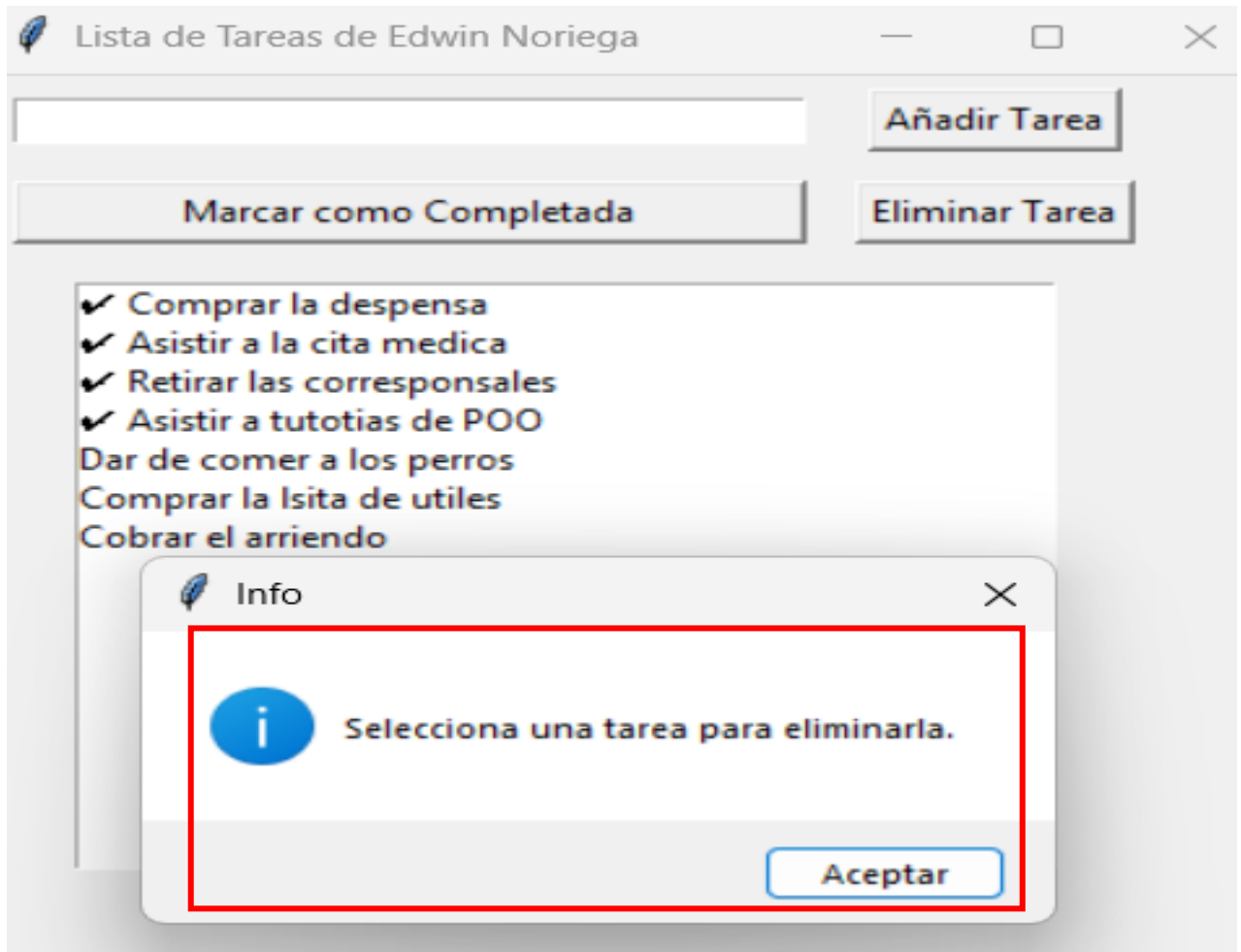




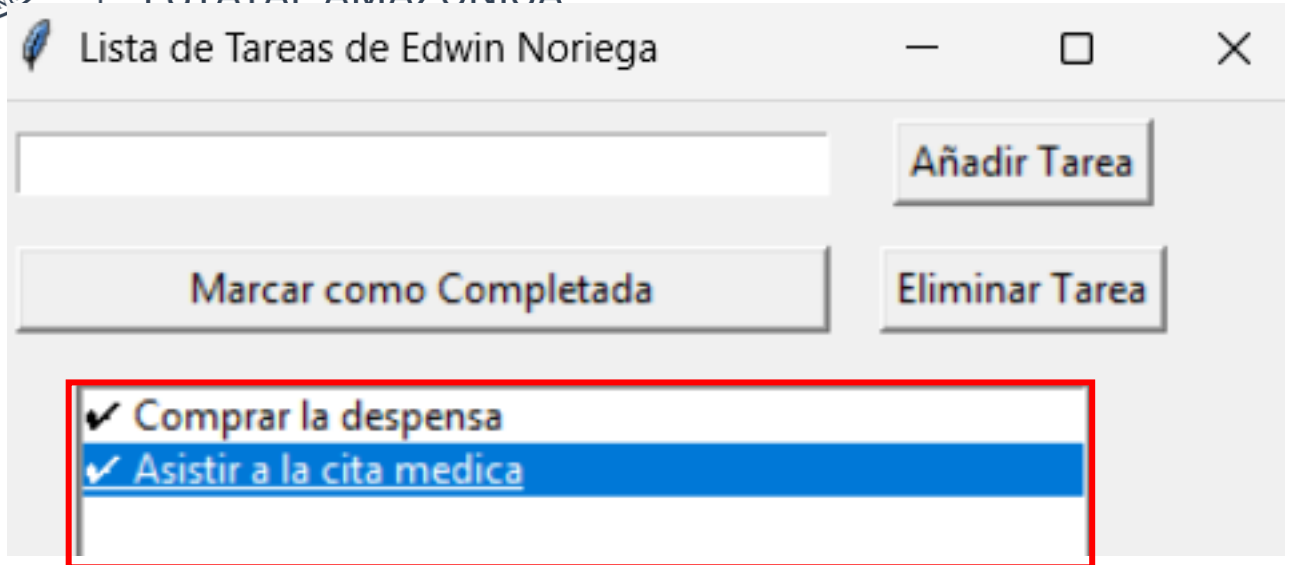
# UEA

## UNIVERSIDAD ESTATAL AMAZÓNICA

- Al eliminar una tarea, esta debe ser removida de la lista.







### Entregables:

- Código fuente de la aplicación.

```
Aplicación GUI de Lista de Tareas.py ×
1 import tkinter as tk # Importa la biblioteca Tkinter para interfaz gráfica
2 from tkinter import messagebox # Importa el módulo para cuadros de diálogo
3
4 class ListaTareasApp: 1 usage
5     def __init__(self, root):
6         """
7         Inicializa la ventana principal y todos los widgets de la aplicación.
8         """
9         self.root = root # Guarda la referencia a la ventana principal
10        self.root.title("Lista de Tareas de Edwin Noriega") # Establece el título de la ventana
11
12        # Campo Entry para que el usuario escriba nuevas tareas
13        self.entry = tk.Entry(root, width=40)
14        # Ubica el Entry en la primera fila y primera columna con espacio exterior
15        self.entry.grid(row=0, column=0, padx=10, pady=10)
16        # Vincula la tecla Enter para agregar tareas al presionar mientras está en el Entry
17        self.entry.bind("<Return>", self.agregar_tarea)
18
19        # Botón para añadir tarea, que ejecuta la función agregar_tarea
20        self.btn_add = tk.Button(root, text="Añadir Tarea", command=self.agregar_tarea)
21        # Posición del botón junto al Entry con un pequeño hueco horizontal
22        self.btn_add.grid(row=0, column=1, padx=5)
23
24        # Botón para marcar una tarea seleccionada como completada
25        self.btn_complete = tk.Button(root, text="Marcar como Completada", command=self.completar_tarea)
26        # Se coloca en la segunda fila, primera columna, y ocupa todo el ancho horizontal de la celda
27        self.btn_complete.grid(row=1, column=0, padx=10, pady=5, sticky="ew")
28
29        # Botón para eliminar la tarea seleccionada
30        self.btn_delete = tk.Button(root, text="Eliminar Tarea", command=self.eliminar_tarea)
31        # Ubicación: segunda fila, segunda columna, también con estiramiento horizontal
32        self.btn_delete.grid(row=1, column=1, padx=5, pady=5, sticky="ew")
```



# UEA

## UNIVERSIDAD ESTATAL AMAZÓNICA

```
33
34 # Listbox para mostrar la lista visible de tareas con selección simple
35 self.lista = tk.Listbox(root, width=50, height=15, selectmode=tk.SINGLE)
36 # Se extiende a lo ancho ocupando dos columnas en la tercera fila
37 self.lista.grid(row=2, column=0, columnspan=2, padx=10, pady=10)
38 # Asigna evento para que al hacer doble clic en una tarea se marque como completada
39 self.lista.bind("<Double-1>", self.completar_tarea)
40
41 # Diccionario para almacenar el estado de cada tarea (True = completada, False = pendiente)
42 self.tareas = {}
43
44 def agregar_tarea(self, event=None): 2 usages
45     """
46     Añade la tarea desde el campo Entry a la lista si el texto no está vacío.
47     Limpia el campo después de añadir y avisa si el campo está vacío.
48     """
49     tarea = self.entry.get().strip() # Extraer el texto y eliminar espacios laterales
50     if tarea: # Solo si la tarea no está vacía
51         self.lista.insert(tk.END, *elements: tarea) # Insertar al final del Listbox
52         self.tareas[tarea] = False # Registrar tarea como no completada
53         self.entry.delete(first=0, tk.END) # Limpiar el campo de entrada
54     else:
55         # Mostrar advertencia si el usuario intenta añadir tarea vacía
56         messagebox.showwarning( title: "Advertencia", message: "No puedes añadir una tarea vacía.")
57
58 def completar_tarea(self, event=None): 2 usages
59     """
60     Marca o desmarca la tarea seleccionada como completada.
61     Cambia el símbolo y el estilo visual (color gris y texto tachado).
62     Si no hay tarea seleccionada informa al usuario.
63     """
64     seleccion = self.lista.curselection() # Obtener selección actual en el Listbox
65     if seleccion:
66         index = seleccion[0] # Índice seleccionado
67         tarea = self.lista.get(index) # Texto de la tarea
68         if not self.tareas.get(tarea, False):
69             # Si no está completada, marcarla como completada
70             self.tareas[tarea] = True # Actualiza estado a True
71             self.lista.delete(index) # Borra elemento del Listbox
72             texto_completado = f"✓ {tarea}" # Agrega prefijo de completado
73             self.lista.insert(index, *elements: texto_completado) # Reinsertar con símbolo
74             # Cambiar estilo visual a gris y texto tachado para claridad visual
75             self.lista.itemconfig(index, fg="gray", font=("Arial", 10, "overstrike"))
76         else:
77             # Si ya estaba completada, revertir a no completada
78             self.tareas[tarea] = False # Estado a False
79             self.lista.delete(index)
80             texto_normal = tarea.replace("✓ ", "") # Quitar símbolo
81             self.lista.insert(index, *elements: texto_normal)
82             # Retornar a estilo visual normal (negro y sin tachado)
83             self.lista.itemconfig(index, fg="black", font=("Arial", 10, "normal"))
84     else:
85         # Mensaje al usuario si no seleccionó ninguna tarea
86         messagebox.showinfo( title: "Info", message: "Selecciona una tarea para marcarla.")
87
```

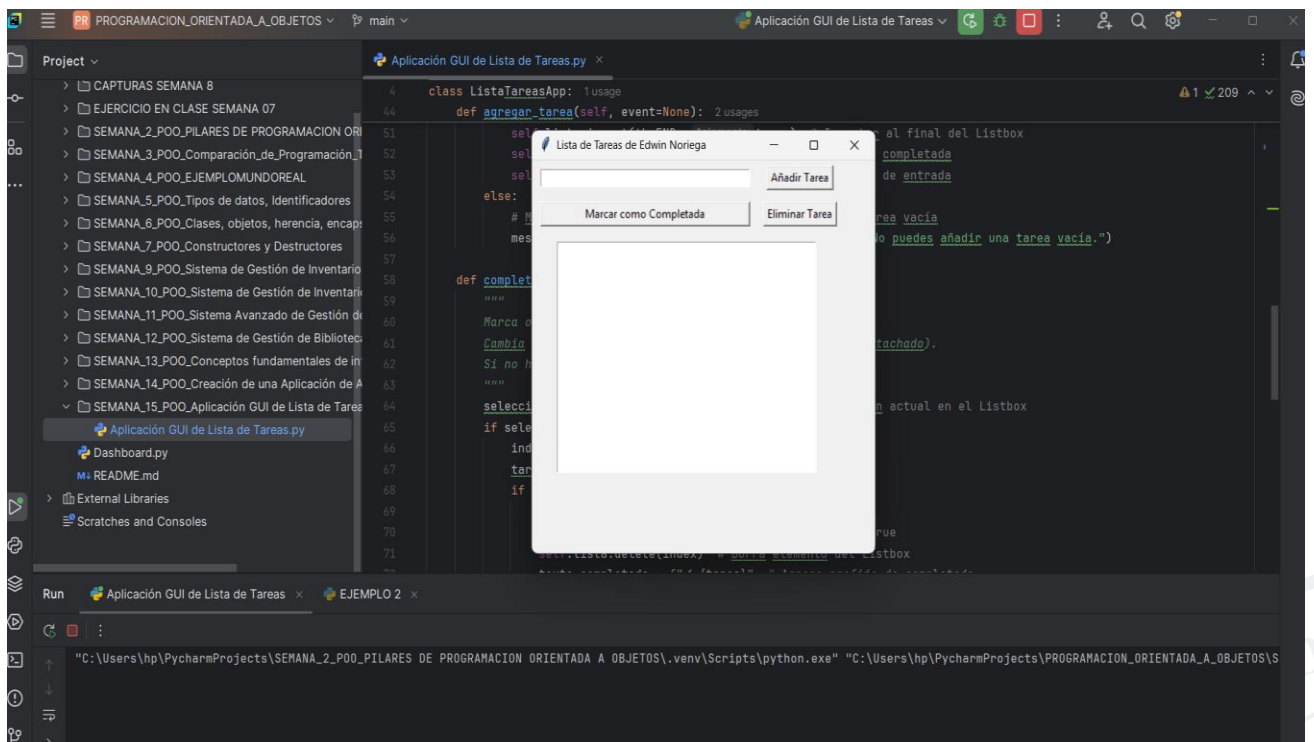


# UEA

## UNIVERSIDAD ESTATAL AMAZÓNICA

```
87
88     def eliminar_tarea(self): 1 usage
89         """
90         Elimina la tarea seleccionada de la lista y del diccionario de estados.
91         Muestra mensaje si no hay una tarea seleccionada.
92         """
93         seleccion = self.lista.curselection()
94         if seleccion:
95             index = seleccion[0]
96             tarea = self.lista.get(index).replace("✓ ", "") # Quitar símbolo si existe
97             if tarea in self.tareas:
98                 del self.tareas[tarea] # Eliminar del diccionario de estados
99                 self.lista.delete(index) # Eliminar del Listbox
100         else:
101             # Informar que debe seleccionar una tarea para eliminar
102             messagebox.showinfo( title= "Info", message= "Selecciona una tarea para eliminarla.")
103
104     if __name__ == "__main__":
105         root = tk.Tk() # Crear ventana principal Tkinter
106         app = ListaTareasApp(root) # Crear instancia de la aplicación
107         root.mainloop() # Ejecutar el bucle principal de eventos
108
```

- Ejecución de código.





Añadir Tarea

Marcar como Completada

Eliminar Tarea

