



UEA

UNIVERSIDAD

ESTATAL AMAZÓNICA

UNIVERSIDAD ESTATAL AMAZÓNICA



INGENIERÍA EN TECNOLOGÍAS DE INFORMACIÓN

PROGRAMACIÓN ORIENTADA A OBJETOS

TEMA:

CONCEPTOS FUNDAMENTALES DE INTERFACES GRAFICAS
DE USUARIO “CREACIÓN DE UNA APLICACIÓN GUI
BÁSICA”

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

2. REPOSITORIO DE GITHUB

EDFANOBA19 / PROGRAMACION_ORIENTADA_A_OBJETOS

Issues Pull requests Actions Projects Wiki Security Insights Settings

PROGRAMACION_ORIENTADA_A_OBJETOS Public

main 1 Branch 0 Tags

Go to file Add file <> Code About

File	Commit Message	Author	Date
.idea	feat: Añadir clase Cliente con atributos y método para mostr...	EDFANOBA	3 months ago
CAPTURAS SEMANA 8	Implementación completa del sistema de gestión de inventa...	EDFANOBA	last month
EJERCICIO EN CLASE SEMANA 07	Implementación de ejercicios de POO en Python:	EDFANOBA	2 months ago
SEMANA_10_POO_Sistema de Gestión de Inve...	Implementación completa del Sistema de Gestión de Invent...	EDFANOBA	3 weeks ago
SEMANA_11_POO_Sistema Avanzado de Gest...	feat: Implementación completa de sistema de inventario en ...	EDFANOBA	2 weeks ago
SEMANA_12_POO_Sistema de Gestión de Bibli...	Implementación completa del Sistema de Gestión de Bibliot...	EDFANOBA	last week
SEMANA_13_POO_Conceptos fundamentales de interfaces gráficas de usuario...	commit -m "Implementar GUI básica con Tkinter: agregado de texto, lis...	EDFANOBA	now
SEMANA_2_POO_PILARES DE PROGRAMACIO...	Add files via upload	EDFANOBA	3 months ago
SEMANA_3_POO_Comparación de Programaci...	Add files via upload	EDFANOBA	3 months ago
SEMANA_4_POO_EJEMPLOMUNDOREAL	Merge remote-tracking branch 'origin/main'	EDFANOBA	2 months ago

EDFANOBA19/PROGRAMACION_ORIENTADA_A_OBJETOS/commit/c4b45bf71e3d9200e5be188e167c786d00b5ca81

33 Commits

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 / PROGRAMACION_ORIENTADA_A_OBJETOS

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main + Q

Go to file

SEMANA_13_POO_Conceptos fundamentales de interfaces gráficas de usuario

PROGRAMACION_ORIENTADA_A_OBJETOS / SEMANA_13_POO_Conceptos fundamentales de interfaces gráficas de usuario

Conceptos fundamentales de interfaces gráficas de usuario_Creación de una Aplicación GUI Básica.py

EDFANOBA git add .

Code Blame 67 lines (56 loc) · 3.1 KB

```
1 import tkinter as tk
2 from tkinter import messagebox
3
4 class AplicacionGUI:
5     def __init__(self, root):
6         # Creamos la referencia a la ventana principal
7         self.root = root
8         # Título descriptivo de la ventana
9         self.root.title("Gestor Básico de Datos - Limpiar y Eliminar con un botón")
10        # Tamaño fijo para que la lista sea visible sin scroll
11        self.root.geometry("450x350")
12
13        # Etiqueta con instrucciones para el usuario
14        self.etiqueta = tk.Label(root, text="Ingrese información y presione 'Agregar'")
15        self.etiqueta.pack(pady=10) # Margen vertical
16
17        # Campo de texto para ingresar datos
18        self.entrada_texto = tk.Entry(root, width=50)
19        self.entrada_texto.pack(pady=5) # Margen vertical
20
21        # Frame que contiene los botones para alinearlos horizontalmente
```



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

3. Código de Aplicación GUI Básica

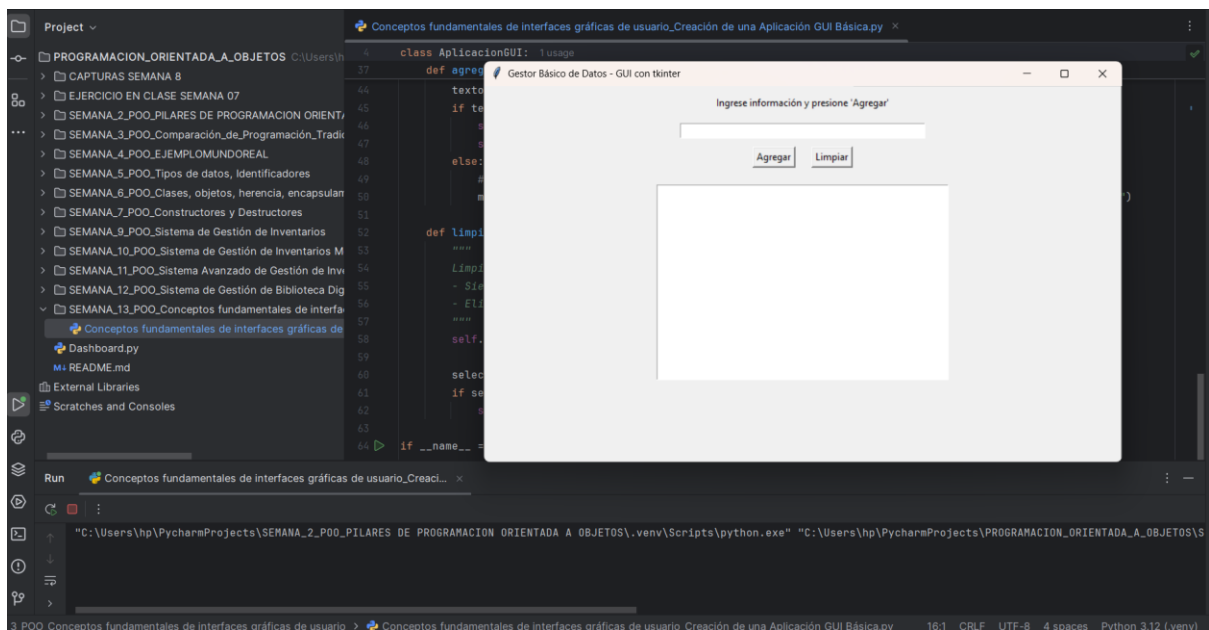
```
Conceptos fundamentales de interfaces gráficas de usuario_Creación de una Aplicación GUI Básica.py ×

1  import tkinter as tk
2  from tkinter import messagebox
3
4  class AplicacionGUI: 1 usage
5      def __init__(self, root):
6          # Guardamos la referencia a la ventana principal
7          self.root = root
8          # Título descriptivo de la ventana
9          self.root.title("Gestor Básico de Datos - GUI con tkinter")
10         # Tamaño fijo para que la lista sea visible sin scroll
11         self.root.geometry("450x350")
12
13         # Etiqueta con instrucciones para el usuario
14         self.etiqueta = tk.Label(root, text="Ingrese información y presione 'Agregar'")
15         self.etiqueta.pack(pady=10) # Margen vertical
16
17         # Campo de texto para ingresar datos
18         self.entrada_texto = tk.Entry(root, width=50)
19         self.entrada_texto.pack(pady=5) # Margen vertical
20
21         # Frame que contiene los botones para alinearlos horizontalmente
22         frame_botones = tk.Frame(root)
23         frame_botones.pack(pady=5) # Margen vertical
24
25         # Botón para agregar el texto de entrada a la lista
26         self.boton_agregar = tk.Button(frame_botones, text="Agregar", command=self.agregar_elemento)
27         self.boton_agregar.pack(side=tk.LEFT, padx=10) # Separación horizontal
28
29         # Botón "Limpiar" que hace doble función: limpiar campo y eliminar elemento seleccionado
30         self.boton_limpiar = tk.Button(frame_botones, text="Limpiar", command=self.limpiar_y_eliminar)
31         self.boton_limpiar.pack(side=tk.LEFT, padx=10) # Separación horizontal
32
33         # Listbox para mostrar los datos que ingresa el usuario
34         self.lista_datos = tk.Listbox(root, width=60, height=15)
35         self.lista_datos.pack(pady=15) # Margen vertical para que se vea la lista claramente
36
37     def agregar_elemento(self): 1 usage
38         """
39         Obtiene el texto ingresado, si no está vacío:
40         - Lo añade al final de la lista.
41         - Limpia el campo de texto para nueva entrada.
42         Si está vacío muestra advertencia.
43         """
44         texto = self.entrada_texto.get().strip() # Eliminar espacios extras
45         if texto: # Si no está vacío
46             self.lista_datos.insert(tk.END, *elements: texto) # Añadir texto al final de la lista
47             self.entrada_texto.delete(first=0, tk.END) # Limpiar campo entrada
48         else:
49             # Mensaje para avisar que es necesario ingresar texto
50             messagebox.showwarning(title="Campo vacío", message="Por favor ingrese algún texto antes de agregar.")
51
52     def limpiar_y_eliminar(self): 1 usage
53         """
54         Limpia el campo de texto y elimina el elemento seleccionado de la lista:
55         - Siempre limpia la caja de texto.
56         - Elimina el ítem seleccionado si hay alguno.
57         """
58         self.entrada_texto.delete(first=0, tk.END) # Limpiar campo
```



```
59
60     seleccion = self.lista_datos.curselection() # Obtener índice(s) seleccionado(s)
61     if seleccion:
62         self.lista_datos.delete(seleccion[0]) # Eliminar el primer elemento seleccionado
63
64     if __name__ == "__main__":
65         ventana = tk.Tk() # Crear ventana principal Tkinter
66         app = AplicacionGUI(ventana) # Instanciar la aplicación con esa ventana
67         ventana.mainloop() # Empezar la escucha de eventos (mostrar ventana)
68
```

3.1. Ejecucion DeCodigo:



4. DESARROLLO DE CÓDIGO CON SUS DIFERENTES PEDIDOS Y REQUISITOS

4.1. objetivo de la tarea

Desarrollar una aplicación de interfaz gráfica de usuario (GUI) que permita a los usuarios interactuar con datos de manera visual, utilizando los conceptos aprendidos sobre GUI.

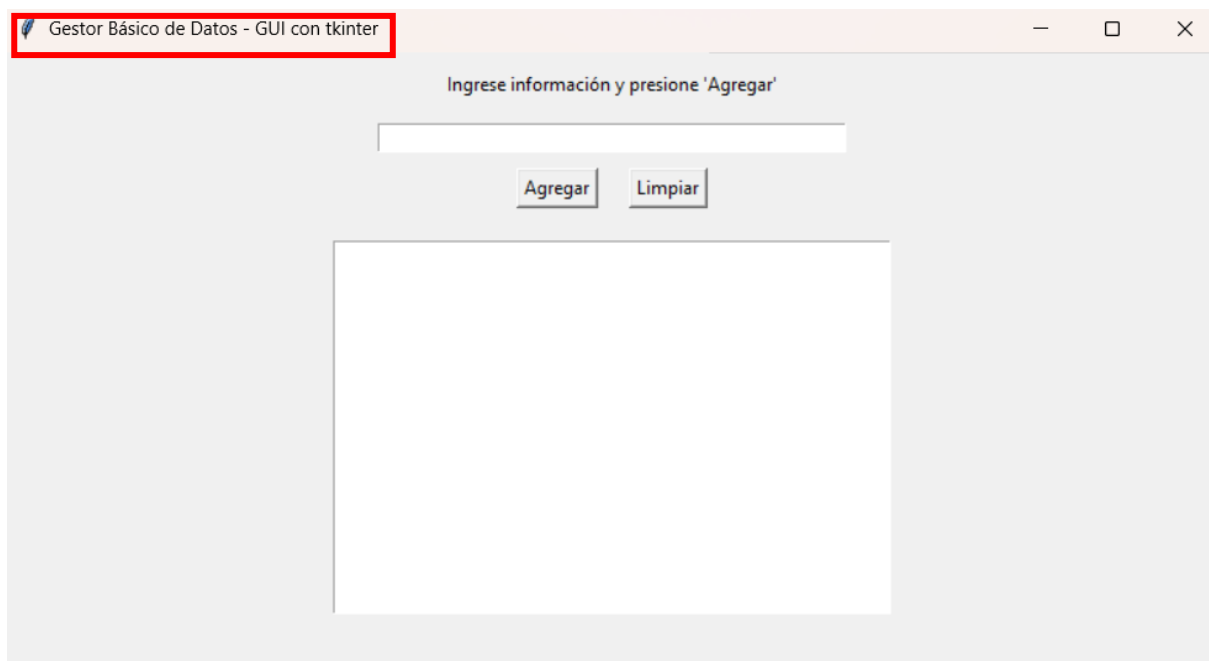
4.2. Requisitos:

4.2.1. Diseño de la Interfaz:

- La aplicación debe tener una ventana principal con un título descriptivo



```
Conceptos fundamentales de interfaces gráficas de usuario_Creación de una Aplicación GUI Básica.py x
1 import tkinter as tk
2 from tkinter import messagebox
3
4 class AplicacionGUI: 1 usage
5     def __init__(self, root):
6         # Guardamos la referencia a la ventana principal
7         self.root = root
8         # Título descriptivo de la ventana
9         self.root.title("Gestor Básico de Datos - GUI con tkinter")
10        # Tamaño fijo para que la lista sea visible sin scroll
11        self.root.geometry("450x350")
12
```



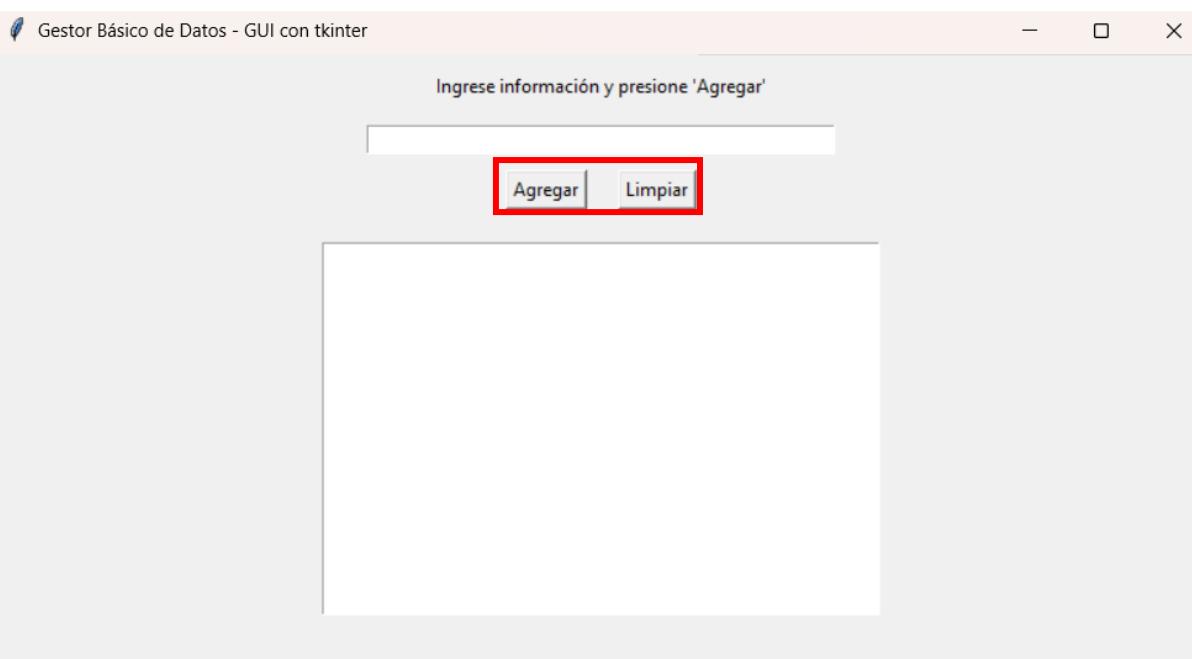
- Incluir al menos los siguientes componentes GUI:
 - Etiquetas (labels)

```
12
13 # Etiqueta con instrucciones para el usuario
14 self.etiqueta = tk.Label(root, text="Ingrese información y presione 'Agregar'")
15 self.etiqueta.pack(pady=10) # Margen vertical
16
```



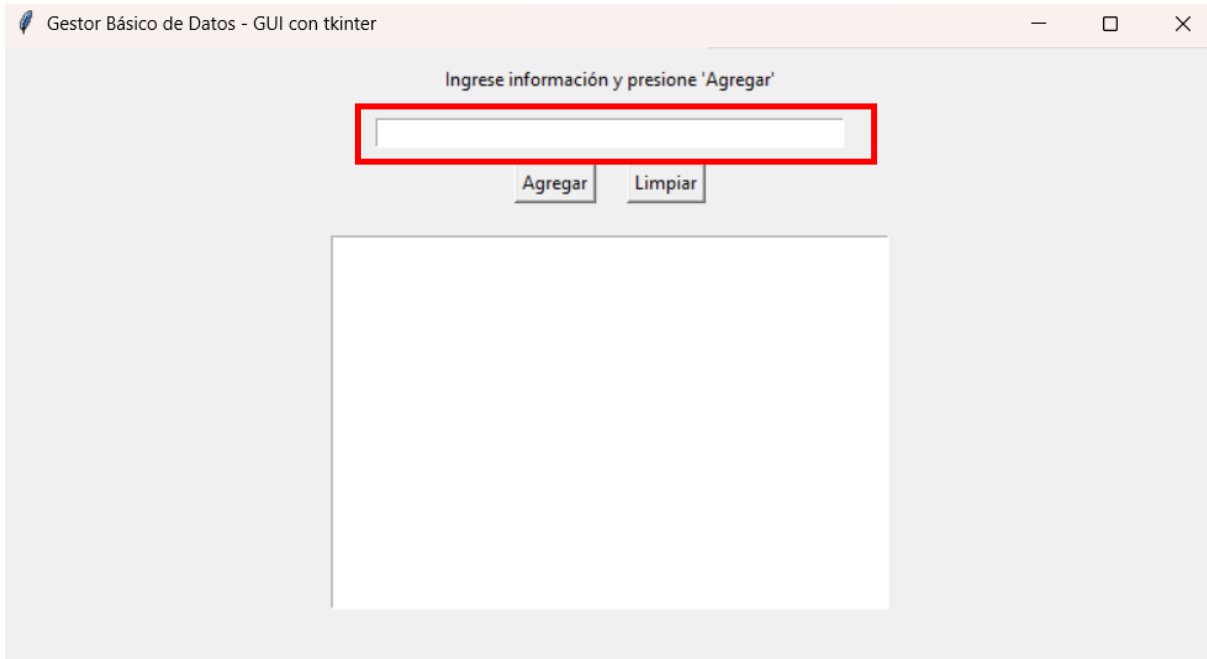
○ Botones

```
21 # Frame que contiene los botones para alinearlos horizontalmente
22 frame_botones = tk.Frame(root)
23 frame_botones.pack(pady=5) # Margen vertical
24
25 # Botón para agregar el texto de entrada a la lista
26 self.boton_agregar = tk.Button(frame_botones, text="Agregar", command=self.agregar_elemento)
27 self.boton_agregar.pack(side=tk.LEFT, padx=10) # Separación horizontal
28
29 # Botón "Limpiar" que hace doble función: limpiar campo y eliminar elemento seleccionado
30 self.boton_limpiar = tk.Button(frame_botones, text="Limpiar", command=self.limpiar_y_eliminar)
31 self.boton_limpiar.pack(side=tk.LEFT, padx=10) # Separación horizontal
32
```



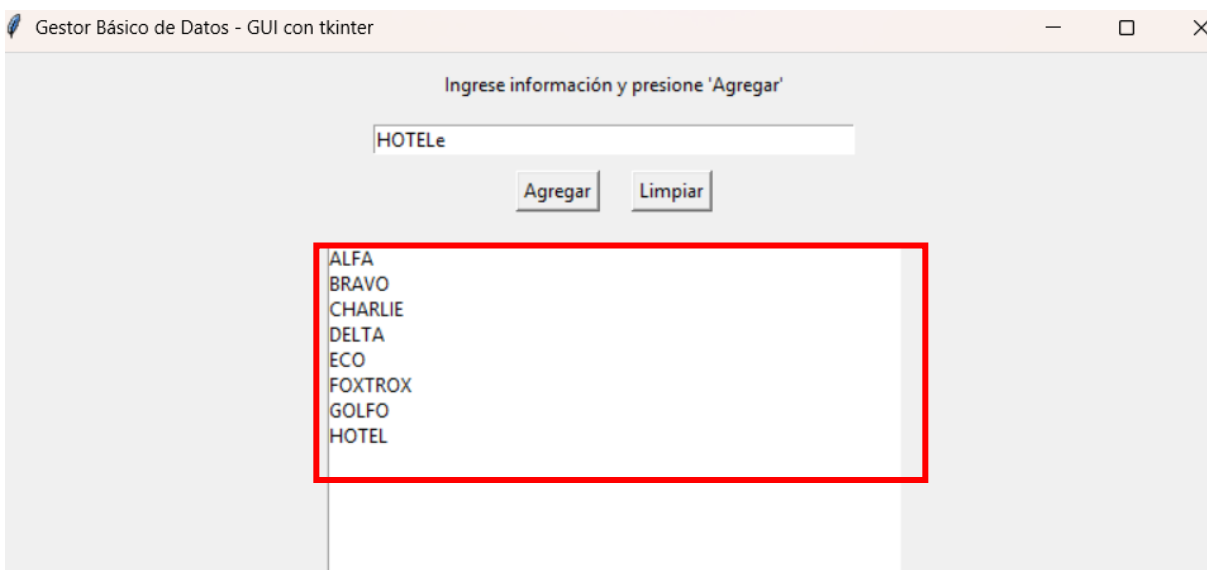


```
17 # Campo de texto para ingresar datos
18 self.entrada_texto = tk.Entry(root, width=50)
19 self.entrada_texto.pack(pady=5) # Margen vertical
20
```



- Una lista o tabla para mostrar datos.

```
33 # Listbox para mostrar los datos que ingresa el usuario
34 self.lista_datos = tk.Listbox(root, width=60, height=15)
35 self.lista_datos.pack(pady=15) # Margen vertical para que se vea la lista claramente
36
```





UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

4.2.2. Funcionalidades:

- Implementar una funcionalidad básica que permita al usuario agregar información a través de un campo de texto y un botón "Agregar".

```
37 def agregar_elemento(self): 1usage
38 """
39 Obtiene el texto ingresado, si no está vacío:
40 - Lo añade al final de la lista.
41 - Limpia el campo de texto para nueva entrada.
42 Si está vacío muestra advertencia.
43 """
44 texto = self.entrada_texto.get().strip() # Eliminar espacios extras
45 if texto: # Si no está vacío
46     self.lista_datos.insert(tk.END, *elements: texto) # Añadir texto al final de la lista
47     self.entrada_texto.delete(first=0, tk.END) # Limpiar campo entrada
48 else:
49     # Mensaje para avisar que es necesario ingresar texto
50     messagebox.showwarning( title: "Campo vacío", message: "Por favor ingrese algún texto antes de agregar.")
51
```

Gestor Básico de Datos - GUI con tkinter

Ingrese información y presione 'Agregar'

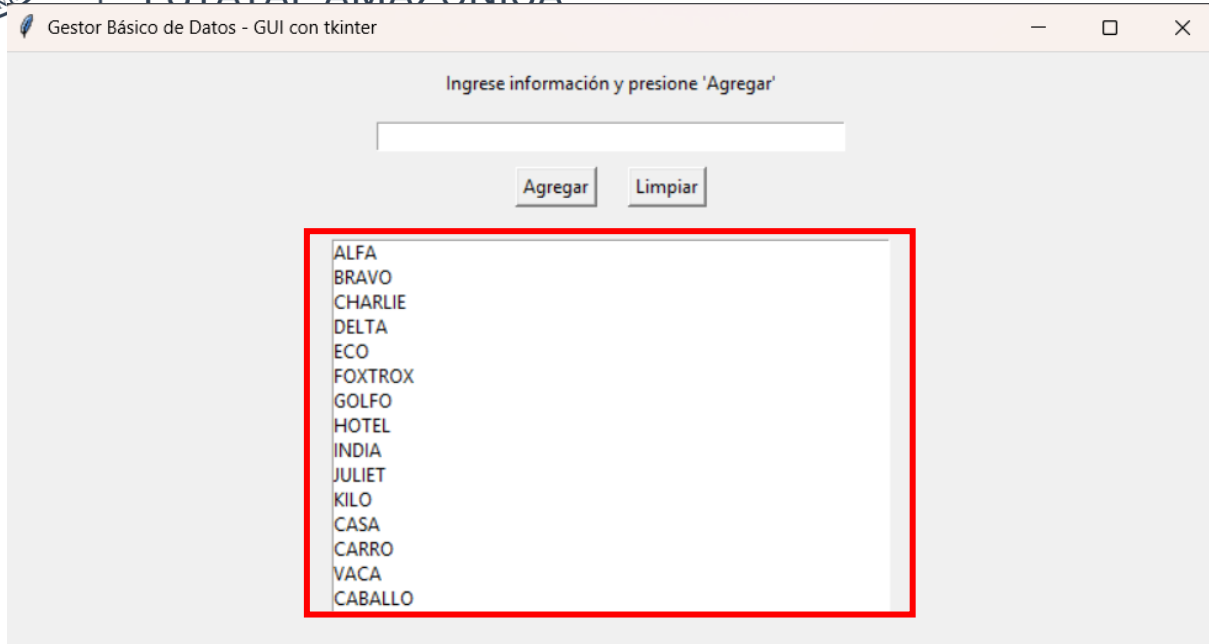
CABALLO

Agregar Limpiar

ALFA
BRAVO
CHARLIE
DELTA
ECO
FOXTROX
GOLFO
HOTEL
INDIA
JULIET
KILO
CASA
CARRO
VACA

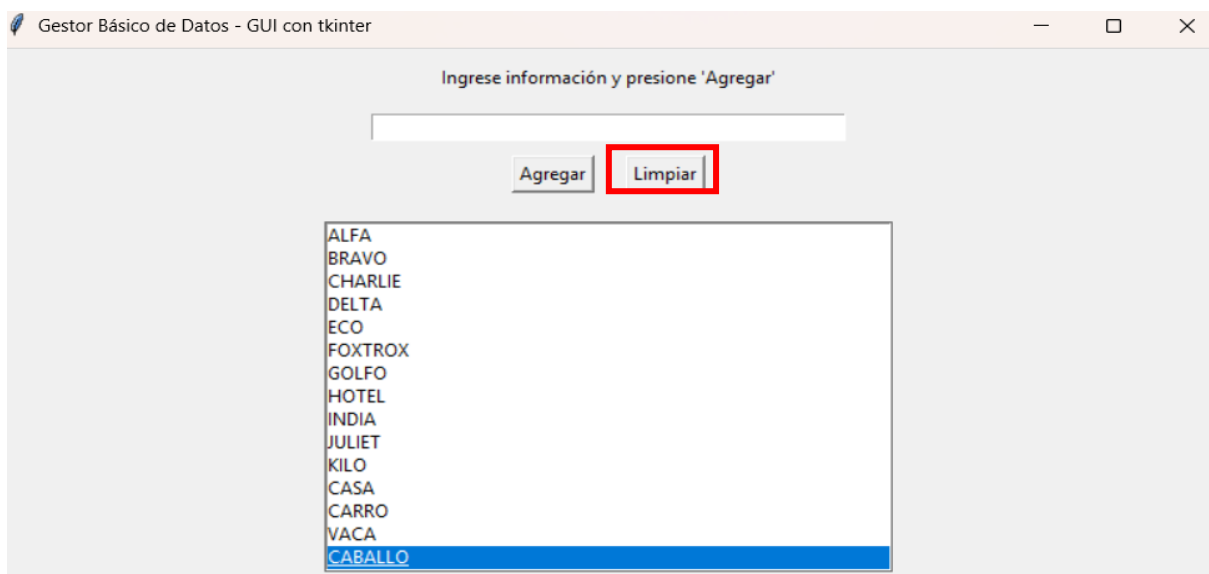
- Mostrar la información agregada en la lista o tabla dentro de la GUI.

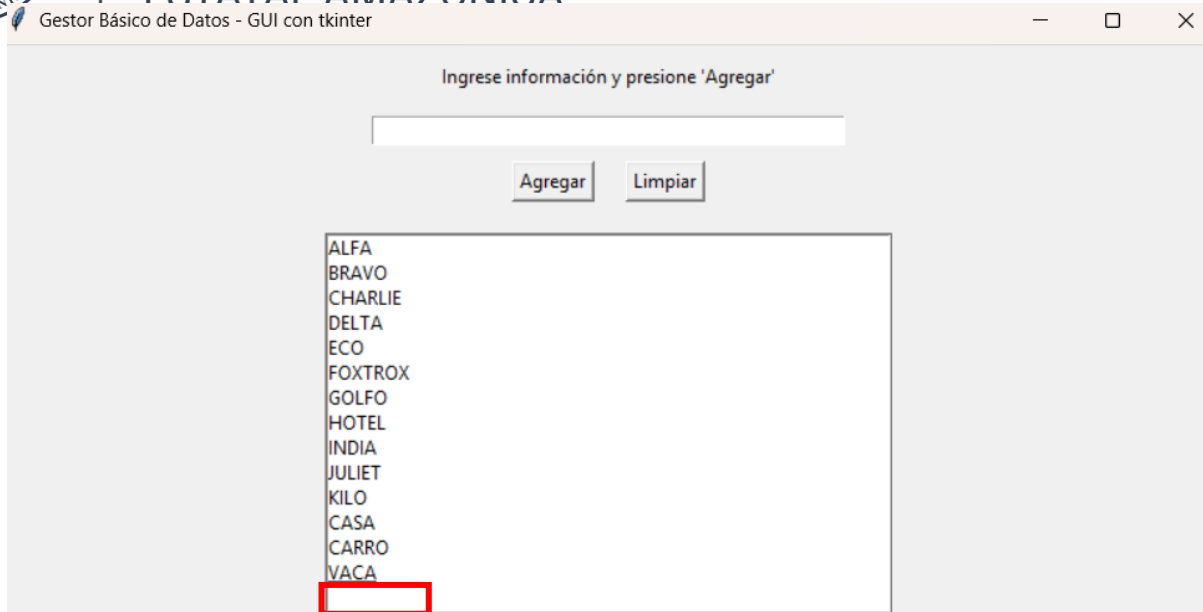
```
33 # Listbox para mostrar los datos que ingresa el usuario
34 self.lista_datos = tk.Listbox(root, width=60, height=15)
35 self.lista_datos.pack(pady=15) # Margen vertical para que se vea la lista claramente
36
```

- Incluir un botón "Limpiar" que borre la información ingresada o seleccionada por el usuario.

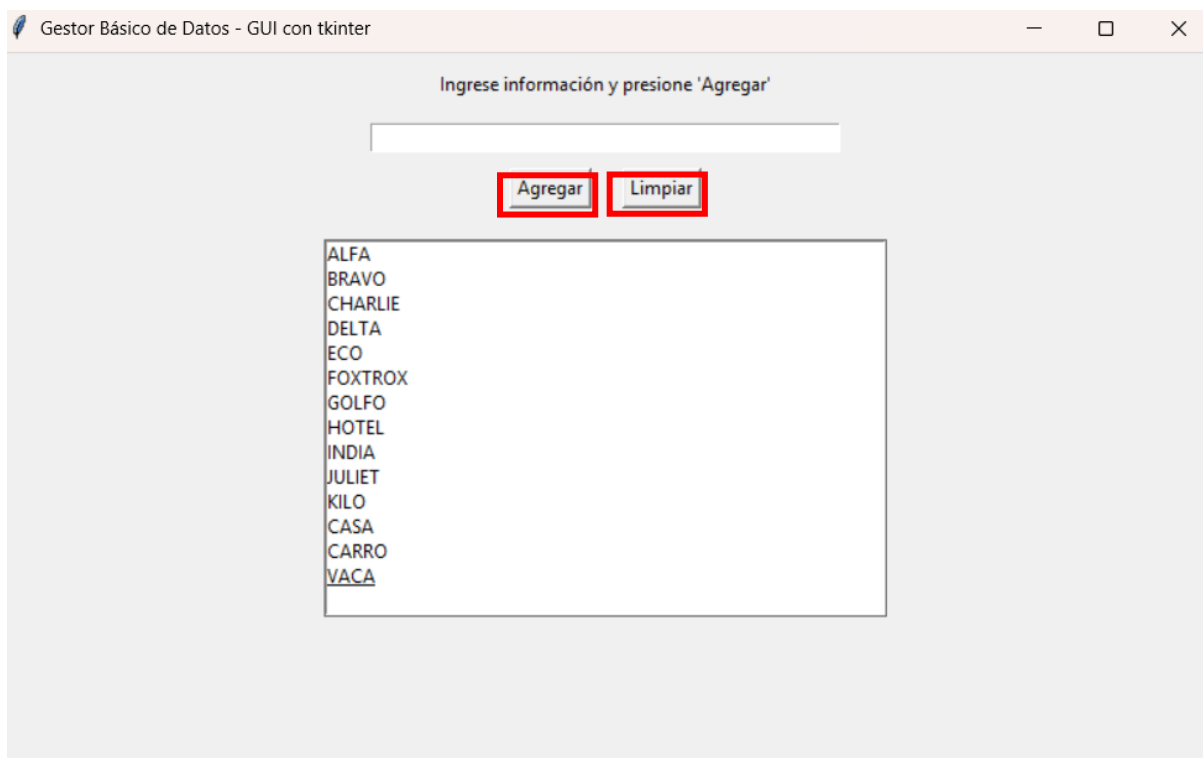
```
52 def limpiar_y_eliminar(self): 1 usage
53     """
54     Limpia el campo de texto y elimina el elemento seleccionado de la lista:
55     - Siempre limpia la caja de texto.
56     - Elimina el ítem seleccionado si hay alguno.
57     """
58     self.entrada_texto.delete(first=0, tk.END) # Limpiar campo
59
60     seleccion = self.lista_datos.curselection() # Obtener índice(s) seleccionado(s)
61     if seleccion:
62         self.lista_datos.delete(seleccion[0]) # Eliminar el primer elemento seleccionado
63
```





4.2.3. Eventos:

- La aplicación debe responder adecuadamente a los eventos generados por el usuario, como clics en los botones.



4.2.4. Librería GUI:

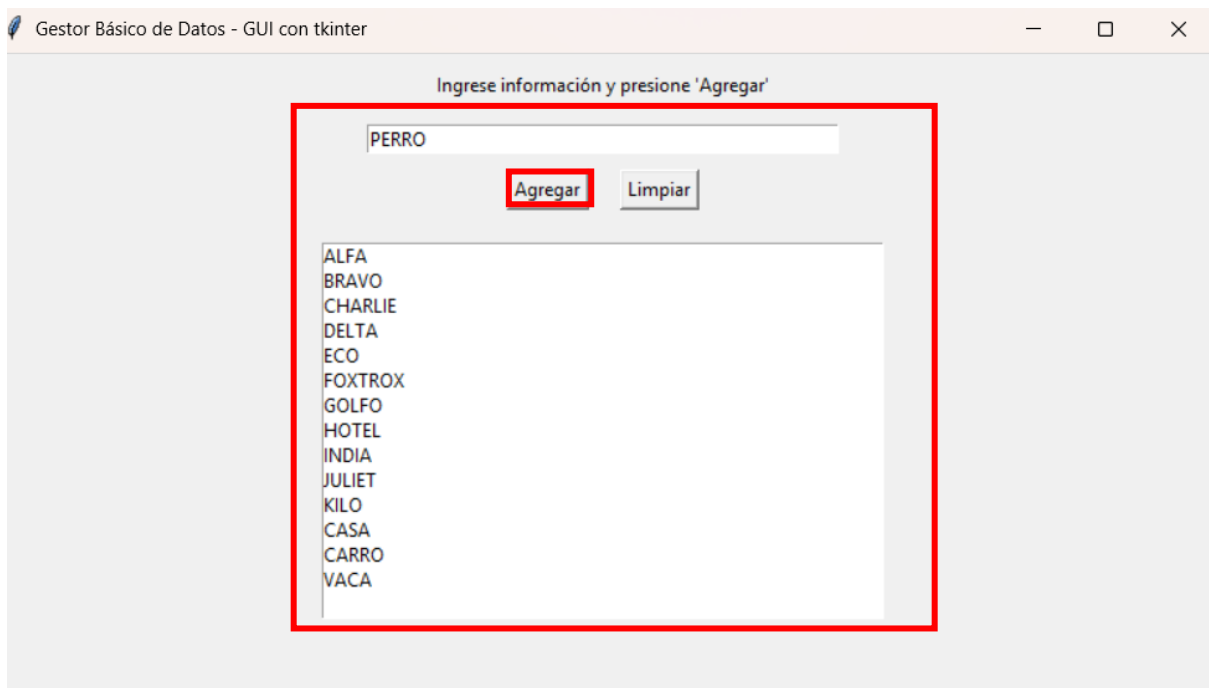


- Utilizar la librería Tkinter en Python o equivalente en el lenguaje de programación de tu elección.

```
Conceptos fundamentales de interfaces gráficas de usuario_Creación de una Aplicación GUI Básica.py x
1 import tkinter as tk
2 from tkinter import messagebox
```

4.3. Instrucciones:

- Diseño y Planificación: Antes de comenzar a codificar, esboza el diseño de tu interfaz, incluyendo la disposición de los componentes. Planifica las funcionalidades específicas que implementarás.
- Desarrollo: Programa la aplicación siguiendo tu diseño. Asegúrate de que la interfaz sea intuitiva y fácil de usar.
- Pruebas: Verifica que todos los componentes funcionen como se espera y que la aplicación maneje los eventos adecuadamente.
 - Agregar información



- Mostrar información



Ingresa información y presione 'Agregar'

- ALFA
- BRAVO
- CHARLIE
- DELTA
- ECO
- FOXTROX
- GOLFO
- HOTEL
- INDIA
- JULIET
- KILO
- CASA
- CARRO
- VACA

○ Eliminar información

Gestor Básico de Datos - GUI con tkinter

Ingresa información y presione 'Agregar'

- ALFA
- BRAVO
- CHARLIE
- DELTA
- ECO
- FOXTROX