



UEA

UNIVERSIDAD

ESTATAL AMAZÓNICA

UNIVERSIDAD ESTATAL AMAZÓNICA



INGENIERÍA EN TECNOLOGÍAS DE INFORMACIÓN

PROGRAMACIÓN ORIENTADA A OBJETOS

TEMA:

**CREACIÓN DE UNA APLICACIÓN DE AGENDA
PERSONAL**

ESTUDIANTE:

NORIEGA BALDEON EDWIN FABIÁN

DOCENTE:

ING. NOGALES GUERRERO SANTIAGO ISRAEL

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

github.com/EDFANOBA19/PROGRAMACION_ORIENTADA_A_OBJETOS

PROGRAMACION_ORIENTADA_A_OBJETOS Public

main 1 Branch 0 Tags

Go to file Add file Code

About

No description, website, 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

EDFANOBA feat: Implementar aplicación de agenda personal con Tkinter 39 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 Gesti...	feat: Implementación completa de sistema de inventario en ...	3 weeks ago
SEMANA_12_POO_Sistema de Gestión de Bibli...	Implementación completa del Sistema de Gestión de Bibliot...	2 weeks ago
SEMANA_13_POO_Conceptos fundamentales ...	git add ...	last week
SEMANA_14_POO_Creación de una Aplicación ...	feat: Implementar aplicación de agenda personal con Tkinter	1 hour 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_EJEMPLUMUNDREAL	Merge remote-tracking branch 'origin/main'	2 months ago
SEMANA_5_POO_Tipos de datos, Identificadores	Add files via upload	3 months ago
SEMANA_6_POO_Clasas, objetos, herencia, enc	Add files via upload	2 months ago

EDFANOBA19 / PROGRAMACION_ORIENTADA_A_OBJETOS

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main

Go to file

PROGRAMACION_ORIENTADA_A_OBJETOS / SEMANA_14_POO_Creación de una Aplicación de Agenda Personal / Creación de una Aplicación de Agenda Personal.py

EDFANOBA feat: Implementar aplicación de agenda personal con Tkinter

Code Blame 102 lines (82 loc) · 4.56 KB

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from tkcalendar import DateEntry
4
5 class AgendaApp:
6     def __init__(self, root):
7         # Inicializa la ventana principal
8         self.root = root
9         self.root.title("Agenda Personal de Edwin Noriega")
10        self.root.geometry("600x400")
11
12        # Frame para contener la lista de eventos
13        frame_lista = tk.Frame(root)
14        frame_lista.pack(fill=tk.BOTH, padx=10, pady=5, expand=True)
15
16        # Crear TreeView con columnas fecha, hora y descripción
17        columnas = ("fecha", "hora", "descripcion")
18        self.tree = ttk.Treeview(frame_lista, columns=columnas, show='headings')
19        self.tree.heading("fecha", text="Fecha")
20        self.tree.heading("hora", text="Hora")
21        self.tree.heading("descripcion", text="Descripción")
```



3. Código de Aplicación de Agenda Personal

```
Creación de una Aplicación de Agenda Personal.py ×
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from tkcalendar import DateEntry
4
5 class AgendaApp:
6     def __init__(self, root):
7         # Inicializa la ventana principal
8         self.root = root
9         self.root.title("Agenda Personal de Edwin Noriega")
10        self.root.geometry("600x400")
11
12        # Frame para contener la lista de eventos
13        frame_lista = tk.Frame(root)
14        frame_lista.pack(fill=tk.BOTH, padx=10, pady=5, expand=True)
15
16        # Crear TreeView con columnas fecha, hora y descripción
17        columnas = ("fecha", "hora", "descripcion")
18        self.tree = ttk.Treeview(frame_lista, columns=columnas, show='headings')
19        self.tree.heading("fecha", text="Fecha")
20        self.tree.heading("hora", text="Hora")
21        self.tree.heading("descripcion", text="Descripción")
22        self.tree.pack(fill=tk.BOTH, expand=True)
23
24        # Frame para contener campos de entrada
25        frame_entrada = tk.Frame(root)
26        frame_entrada.pack(fill=tk.X, padx=10, pady=5)
27
28        # Etiqueta y widget DateEntry para seleccionar fecha
29        tk.Label(frame_entrada, text="Fecha:").grid(row=0, column=0, padx=5, pady=2)
30        self.entry_fecha = DateEntry(frame_entrada, date_pattern='dd/mm/yyyy')
31        self.entry_fecha.grid(row=0, column=1, padx=5, pady=2)
32
33        # Etiqueta y campo Entry para ingresar hora
34        tk.Label(frame_entrada, text="Hora (HH:MM):").grid(row=0, column=2, padx=5, pady=2)
35        self.entry_hora = tk.Entry(frame_entrada)
36        self.entry_hora.grid(row=0, column=3, padx=5, pady=2)
37
38        # Etiqueta y campo Entry para ingresar descripción
39        tk.Label(frame_entrada, text="Descripción:").grid(row=1, column=0, padx=5, pady=2)
40        self.entry_descripcion = tk.Entry(frame_entrada, width=40)
41        self.entry_descripcion.grid(row=1, column=1, columnspan=3, padx=5, pady=2)
42
43        # Frame para botones
44        frame_botones = tk.Frame(root)
45        frame_botones.pack(fill=tk.X, padx=10, pady=10)
46
47        # Botón para agregar evento, asociado al método agregar_evento
48        btn_agregar = tk.Button(frame_botones, text="Agregar Evento", command=self.agregar_evento)
49        btn_agregar.pack(side=tk.LEFT, padx=5)
50
51        # Botón para eliminar evento seleccionado, asociado al método eliminar_evento
52        btn_eliminar = tk.Button(frame_botones, text="Eliminar Evento Seleccionado", command=self.eliminar_evento)
53        btn_eliminar.pack(side=tk.LEFT, padx=5)
54
55        # Botón para salir de la aplicación
56        btn_salir = tk.Button(frame_botones, text="Salir", command=root.quit)
57        btn_salir.pack(side=tk.RIGHT, padx=5)
```



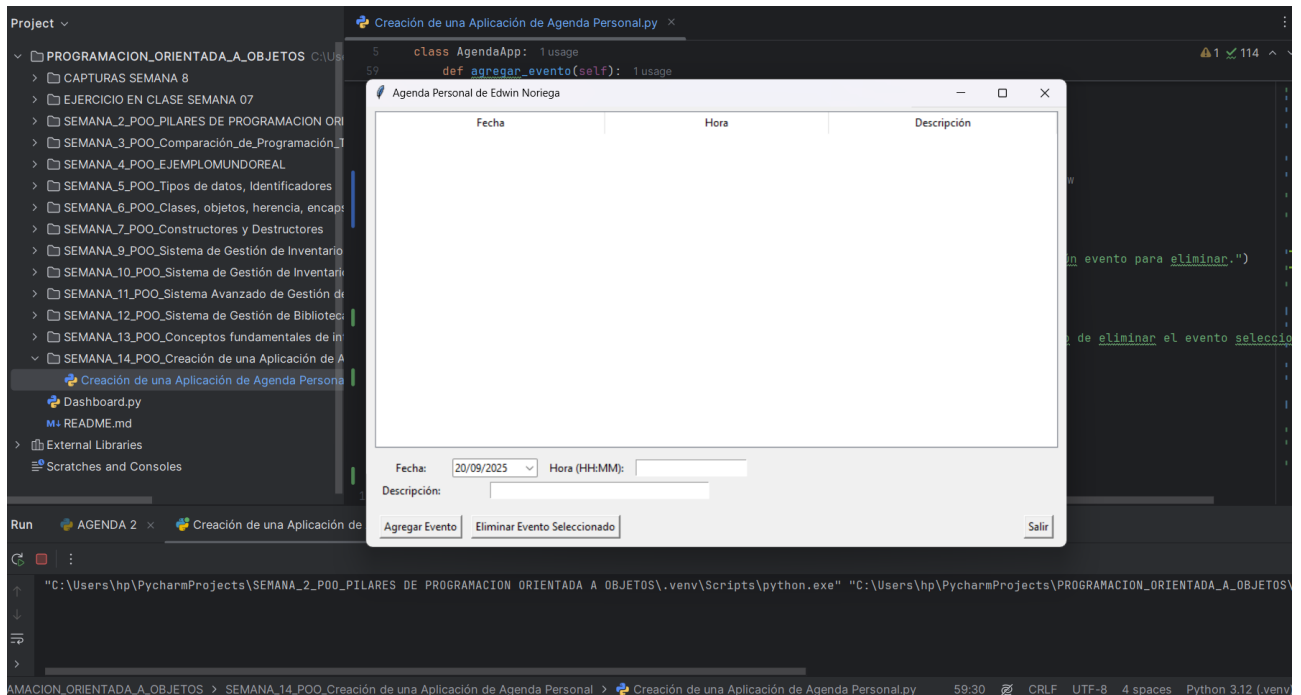
UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

```
58
59 def agregar_evento(self): 1 usage
60     """Agrega un evento nuevo a la lista desde los campos de entrada."""
61     fecha = self.entry_fecha.get() # Obtiene la fecha seleccionada en el selector de fecha
62     hora = self.entry_hora.get().strip() # Obtiene la hora ingresada y elimina espacios extra
63     descripcion = self.entry_descripcion.get().strip() # Obtiene la descripción y elimina espacios extra
64
65     # Validar que los campos hora y descripción no estén vacíos
66     if not hora or not descripcion:
67         messagebox.showwarning( title: "Campos incompletos", message: "Por favor, complete todos los campos.")
68         return
69
70     # Validar formato simple de hora HH:MM
71     if len(hora) != 5 or hora[2] != ':' or not (hora[2].isdigit() and hora[3].isdigit()):
72         messagebox.showwarning( title: "Hora inválida", message: "Formato de hora debe ser HH:MM (ej. 14:30).")
73         return
74
75     # Insertar el evento en la tabla TreeView
76     self.tree.insert( parent: '', index: 'end', values=(fecha, hora, descripcion))
77
78     # Limpiar los campos de entrada hora y descripción
79     self.entry_hora.delete( first: 0, tk.END)
80     self.entry_descripcion.delete( first: 0, tk.END)
81
82 def eliminar_evento(self): 1 usage
83     """Elimina el evento seleccionado tras confirmación."""
84     seleccion = self.tree.selection() # Obtener el elemento seleccionado en el TreeView
85
86     # Si no hay selección, mostrar mensaje informativo
87     if not seleccion:
88         messagebox.showinfo( title: "Seleccione evento", message: "No ha seleccionado ningún evento para eliminar.")
89         return
90
91     # Confirmar la eliminación con el usuario
92     respuesta = messagebox.askyesno( title: "Confirmar eliminación", message: "¿Está seguro de eliminar el evento seleccionado?")
93     if respuesta:
94         # Eliminar evento(s) seleccionado(s)
95         for item in seleccion:
96             self.tree.delete(item)
97
98 if __name__ == "__main__":
99     # Crear ventana principal y ejecutar la aplicación
100     root = tk.Tk()
101     app = AgendaApp(root)
102     root.mainloop()
103
```



3.1. Ejecucion De Codigo:



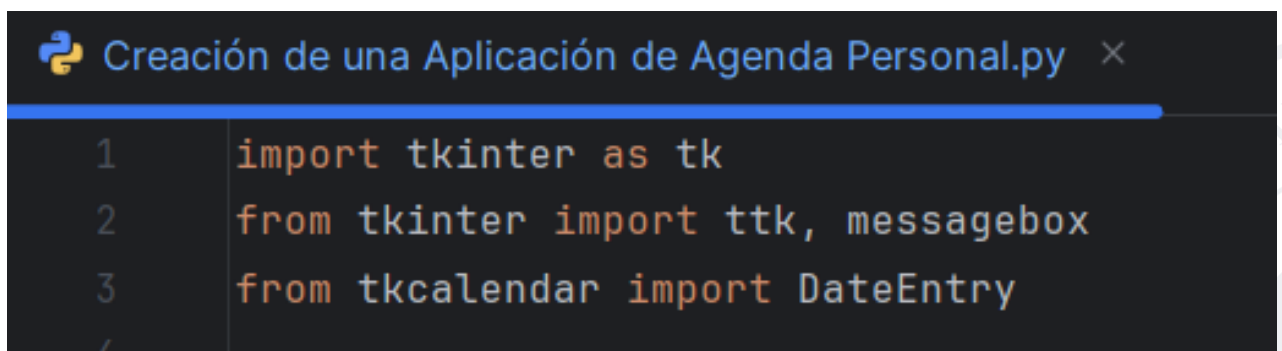
4. DESARROLLO DE CÓDIGO CON SUS DIFERENTES PEDIDOS Y

REQUISITOS:

Objetivo: Desarrollar una aplicación GUI utilizando Tkinter en Python que funcione como una agenda personal. La aplicación permitirá al usuario agregar, ver, y eliminar eventos o tareas programadas.

Requisitos:

- **Interfaz Gráfica:**
 - Utilizar Tkinter para diseñar la interfaz.

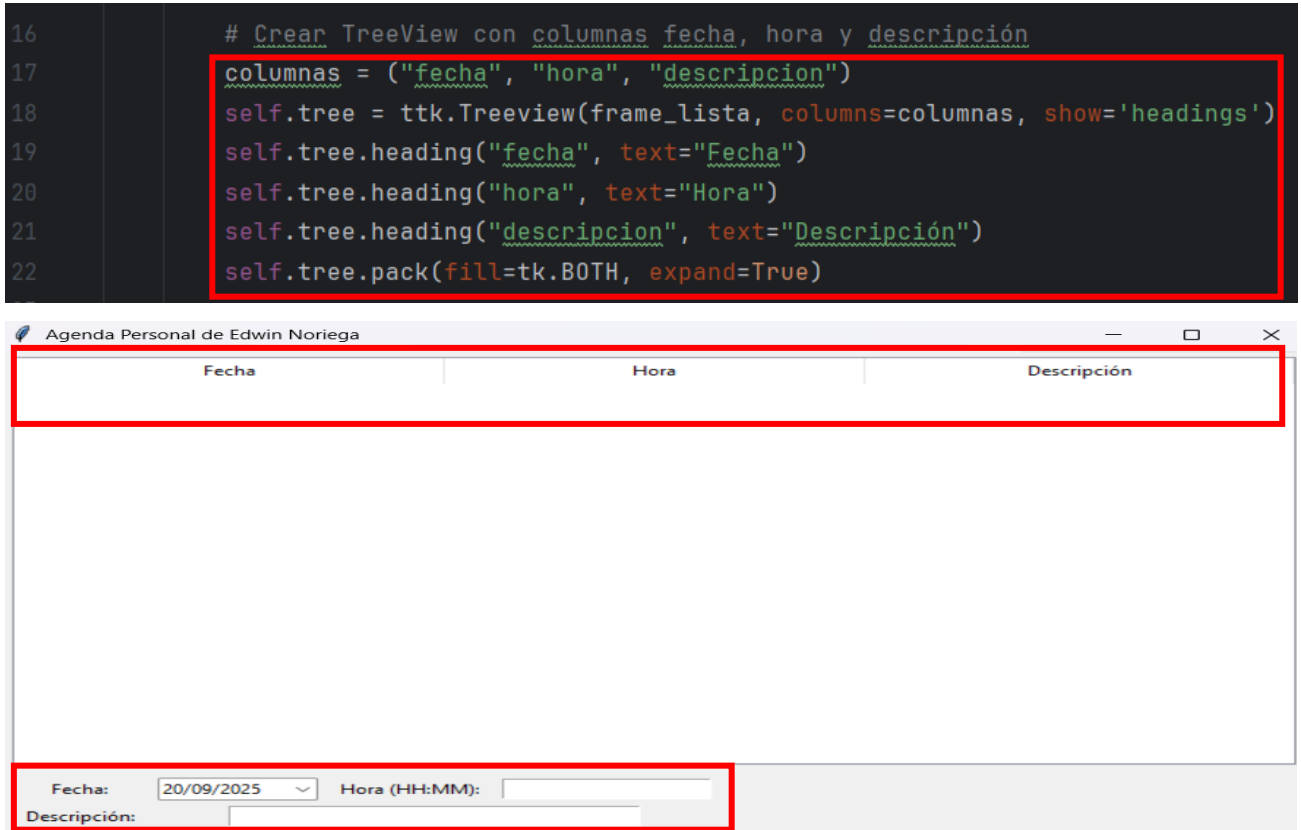




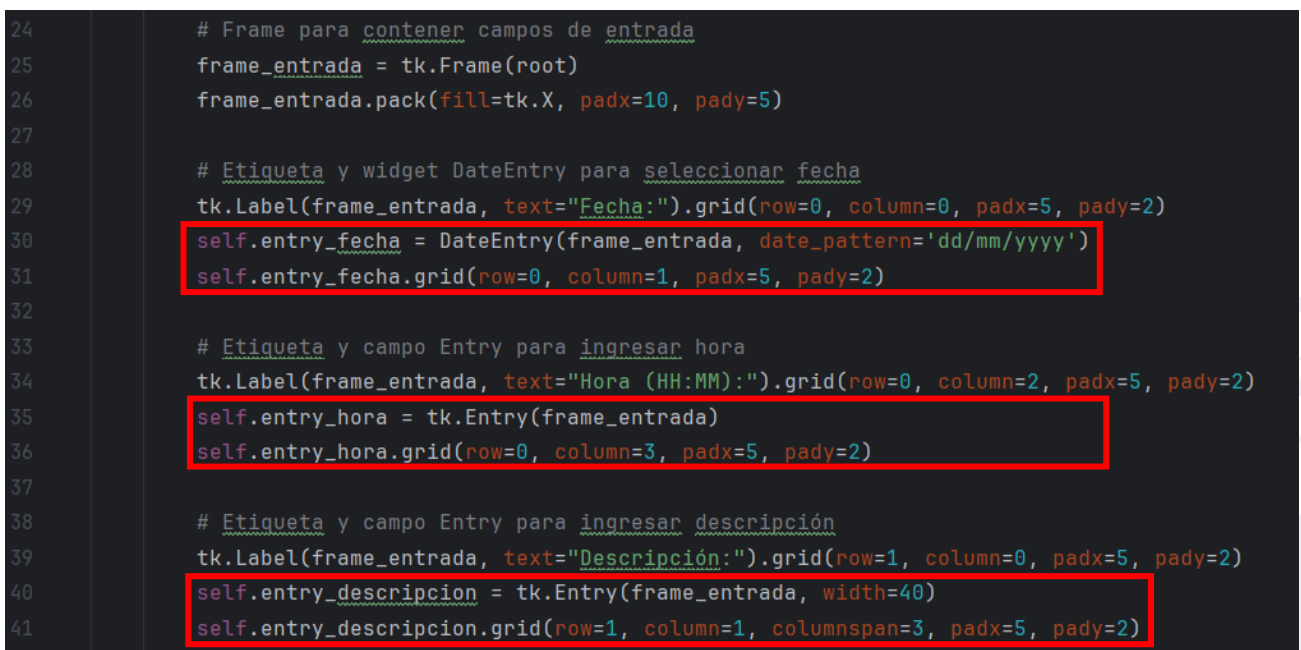
UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

- La aplicación debe tener una ventana principal que muestre una lista (TreeView) de eventos o tareas programadas, incluyendo detalles como la fecha, la hora y una breve descripción.



- Incluir campos de entrada (Entry) para que el usuario pueda introducir la fecha, la hora y la descripción del nuevo evento o tarea.





Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
-------	------	-------------

Fecha: Hora (HH:MM):
Descripción:

- Añadir botones para las acciones de "Agregar Evento", "Eliminar Evento Seleccionado" y "Salir".

```
43 # Frame para botones
44 frame_botones = tk.Frame(root)
45 frame_botones.pack(fill=tk.X, padx=10, pady=10)
46
47 # Botón para agregar evento, asociado al método agregar_evento
48 btn_agregar = tk.Button(frame_botones, text="Agregar Evento", command=self.agregar_evento)
49 btn_agregar.pack(side=tk.LEFT, padx=5)
50
51 # Botón para eliminar evento seleccionado, asociado al método eliminar_evento
52 btn_eliminar = tk.Button(frame_botones, text="Eliminar Evento Seleccionado", command=self.eliminar_evento)
53 btn_eliminar.pack(side=tk.LEFT, padx=5)
54
55 # Botón para salir de la aplicación
56 btn_salir = tk.Button(frame_botones, text="Salir", command=root.quit)
57 btn_salir.pack(side=tk.RIGHT, padx=5)
```

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
-------	------	-------------

Fecha: Hora (HH:MM):
Descripción:



- Implementar un DatePicker (utilizando un widget adicional o un método alternativo en Tkinter) para la selección de fechas.

[illegible]

- Usar etiquetas (Labels) para identificar cada campo de entrada y botón.



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

```
28 # Etiqueta y widget DateEntry para seleccionar fecha
29 tk.Label(frame_entrada, text="Fecha:").grid(row=0, column=0, padx=5, pady=2)
30 self.entry_fecha = DateEntry(frame_entrada, date_pattern='dd/mm/yyyy')
31 self.entry_fecha.grid(row=0, column=1, padx=5, pady=2)
32
33 # Etiqueta y campo Entry para ingresar hora
34 tk.Label(frame_entrada, text="Hora (HH:MM):").grid(row=0, column=2, padx=5, pady=2)
35 self.entry_hora = tk.Entry(frame_entrada)
36 self.entry_hora.grid(row=0, column=3, padx=5, pady=2)
37
38 # Etiqueta y campo Entry para ingresar descripción
39 tk.Label(frame_entrada, text="Descripción:").grid(row=1, column=0, padx=5, pady=2)
40 self.entry_descripcion = tk.Entry(frame_entrada, width=40)
41 self.entry_descripcion.grid(row=1, column=1, columnspan=3, padx=5, pady=2)
```

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
-------	------	-------------

Fecha: 20/09/2025 Hora (HH:MM): Descripción:

- (Opcional) Incorporar un diálogo de confirmación cuando se solicite eliminar un evento.

```
82 def eliminar_evento(self): 1 usage
83 """Elimina el evento seleccionado tras confirmación."""
84 seleccion = self.tree.selection() # Obtener el elemento seleccionado en el TreeView
85
86 # Si no hay selección, mostrar mensaje informativo
87 if not seleccion:
88     messagebox.showinfo(title="Seleccione evento", message="No ha seleccionado ningún evento para eliminar.")
89     return
90
91 # Confirmar la eliminación con el usuario
92 respuesta = messagebox.askyesno(title="Confirmar eliminación", message="¿Está seguro de eliminar el evento seleccionado?")
93 if respuesta:
94     # Eliminar evento(s) seleccionado(s)
95     for item in seleccion:
96         self.tree.delete(item)
```



Fecha	Hora	Descripción
20/09/2025	18:30	Conexion a tutoria de estadística
20/09/2025	22:00	Entrega de TS_14_POO

Confirmar eliminación

¿Está seguro de eliminar el evento seleccionado?

Fecha: 21/09/2025 Hora (HH:MM):

Descripción:

- **Manejo de Eventos:**

- Programar los botones para que respondan adecuadamente a los clics del usuario, ejecutando las acciones correspondientes.

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
-------	------	-------------

Fecha: 21/09/2025 Hora (HH:MM):

Descripción:



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

- La acción de "Agregar Evento" deberá actualizar la lista de eventos mostrada en la ventana principal.

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras
21/09/2025	14:00	Salida al boly
21/09/2025	18:00	Verificar si hay actividades pendientes de la U
21/09/2025	22:30	Organizar todo para el trabajo

Fecha: 22/09/2025 Hora (HH:MM): 06:00

Descripción: Desayuno reforzado

Agregar Evento Eliminar Evento Seleccionado Salir

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras
21/09/2025	14:00	Salida al boly
21/09/2025	18:00	Verificar si hay actividades pendientes de la U
21/09/2025	22:30	Organizar todo para el trabajo
22/09/2025	06:00	Desayuno reforzado

Fecha: 22/09/2025 Hora (HH:MM):

Descripción:

Agregar Evento Eliminar Evento Seleccionado Salir



- **Organización con Contenedores:**

- Organizar los componentes de la interfaz usando Frames para separar lógicamente las áreas de visualización de eventos, entrada de datos y acciones.

```
12 # Frame para contener la lista de eventos
13 frame_lista = tk.Frame(root)
14 frame_lista.pack(fill=tk.BOTH, padx=10, pady=5, expand=True)
15
24 # Frame para contener campos de entrada
25 frame_entrada = tk.Frame(root)
26 frame_entrada.pack(fill=tk.X, padx=10, pady=5)
27
43 # Frame para botones
44 frame_botones = tk.Frame(root)
45 frame_botones.pack(fill=tk.X, padx=10, pady=10)
```

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras
21/09/2025	14:00	Salida al boly
21/09/2025	18:00	Verificar si hay actividades pendientes de la U
21/09/2025	22:30	Organizar todo para el trabajo
22/09/2025	06:00	Desayuno reforzado

Fecha: 22/09/2025 Hora (HH:MM):

Descripción:

Agregar Evento Eliminar Evento Seleccionado Salir

Instrucciones Adicionales:

- Pruebas: Probar la aplicación para asegurar que todos los componentes funcionan como se espera y que no hay errores al agregar o eliminar eventos.



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

Agregar evento:

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras
21/09/2025	14:00	Salida al boly
21/09/2025	18:00	Verificar si hay actividades pendientes de la U
21/09/2025	22:30	Organizar todo para el trabajo
22/09/2025	06:00	Desayuno reforzado

Fecha: 22/09/2025 Hora (HH:MM): 07:30

Descripción:

Agregar Evento

September 2025

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
36	1	2	3	4	5	6	7
37	8	9	10	11	12	13	14
38	15	16	17	18	19	20	21
39	22	23	24	25	26	27	28
40	29	30	1	2	3	4	5
41	6	7	8	9	10	11	12

#UEAesExcelencia

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras
21/09/2025	14:00	Salida al boly
21/09/2025	18:00	Verificar si hay actividades pendientes de la U
21/09/2025	22:30	Organizar todo para el trabajo
22/09/2025	06:00	Desayuno reforzado

Fecha: 22/09/2025 Hora (HH:MM): 07:30

Descripción:

Agregar Evento Eliminar Evento Seleccionado



Evento agregado

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras
21/09/2025	14:00	Salida al boly
21/09/2025	18:00	Verificar si hay actividades pendientes de la U
21/09/2025	22:30	Organizar todo para el trabajo
22/09/2025	06:00	Desayuno reforzado
22/09/2025	07:30	Agregar accion de prueba

Fecha: Hora (HH:MM):

Descripción:

Eliminar eventos

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras
21/09/2025	14:00	Salida al boly
21/09/2025	18:00	Verificar si hay actividades pendientes de la U
21/09/2025	22:30	Organizar todo para el trabajo
22/09/2025	06:00	Desayuno reforzado
22/09/2025	07:30	Agregar accion de prueba

Fecha: Hora (HH:MM):

Descripción:



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras
21/09/2025	14:00	Salida al boly
21/09/2025	18:00	Verificar si hay actividades pendientes de la U
21/09/2025	22:30	Organizar todo para el trabajo
22/09/2025	06:00	Desayuno reforzado
22/09/2025	07:30	Agregar accion de prueba

Confirmar eliminación

¿Está seguro de eliminar el evento seleccionado?

Sí No

Fecha: 22/09/2025 Hora (HH:MM):

Descripción:

Agregar Evento Eliminar Evento Seleccionado Salir

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM
21/09/2025	10:30	Ir al mercado por verduras

Confirmar eliminación

¿Está seguro de eliminar el evento seleccionado?

Sí No

Fecha: 22/09/2025 Hora (HH:MM):

Descripción:

Agregar Evento Eliminar Evento Seleccionado Salir



UEA

UNIVERSIDAD
ESTATAL AMAZÓNICA

Agenda Personal de Edwin Noriega

Fecha	Hora	Descripción
20/09/2025	22:00	Entrega de la TS_14_POO
21/09/2025	08:00	GYM

Fecha: Hora (HH:MM):

Descripción:



www.uea.edu.ec



Km. 2. 1/2 vía Puyo a Tena (Paso Lateral)



032892-118 / 032892-188 032892-098 / 032896-188 032896-476

#UEAesExcelencia