

PLAN DE TRABAJO DEL ESTUDIANTE

1. INFORMACIÓN GENERAL

Apellidos y Nombres: Sarai Rut Romero Villanueva ID: 1472173
 Dirección Zonal/CFP: Huánuco-Huánuco
 Carrera: Ingeniería de software con inteligencia artificial Semestre: II
 Curso/ Mód. Formativo: Algoritmia de programación del software
 Tema del Trabajo: Creación de Aplicaciones básicas en Python

2. PLANIFICACIÓN DEL TRABAJO

Nº	ACTIVIDADES/ ENTREGABLES	CRONOGRAMA/ FECHA DE ENTREGA									
1.	La interfaz Gráfica con Tkinter para cumplir con las exigencias del usuario	1	1	/	0	3	/	2	0	2	3
2.	Diagrama de flujo y código fuente en Python que resuelva el requerimiento.	1	2	/	0	3	/	2	0	2	3
3.	Un informe del procedimiento desarrollado para resolver el problema planteado.	1	3	/	0	3	/	2	0	2	3

3. PREGUNTAS GUIA

Durante la investigación de estudio, debes obtener las respuestas a las siguientes interrogantes:

Nº	PREGUNTAS
1	¿Cómo se maneja las Cadenas, Operaciones y se crea colecciones de datos en Python?
2	¿Cómo generar Clases, Objetos y Funciones en Python?
3	¿Cómo crear una aplicación con el manejo de errores y gestión de expresiones regulares?
4	¿Cómo estructurar una Base de Datos en Python?
5	¿Cómo generar una interfaz Gráfica con Tkinter?
6	¿Cómo trabajar con módulos, ficheros de texto y binarios en Python?
7	¿Cómo generar documentación Automáticamente y trabajar con Pruebas Automáticas?

HOJA DE RESPUESTAS A LAS PREGUNTAS GUÍA
1. ¿Cómo se maneja las cadenas, operaciones y se crea colecciones de datos en python?

El concepto de cadena es que es un conjunto de caracteres delimitados por comillas y que sus operaciones se manejan únicamente mediante **“Concatenar Cadenas (+)”**, **“Multiplicar una cadena (*)”** y **“Añadir un carácter o cadena (+=)”**. La creación de un colección de datos se hace mediante corchetes “[]” y existen 3 tipos de colecciones (**Listas, Tuplas y Diccionarios**).


2. ¿Cómo generar clases, objetos y funciones en python?

Una clase se puede generar mediante un módulo o función ya hecho por Python o también con la palabra clave “class” y el “nombreClase” y mediante el nombre o tipo de clase podemos crear nuestros objetos y así darles valores en los métodos de la clase. La función es un bloque de código con un nombre asociado, que recibe 0 o más argumentos de entrada y está la podemos generar mediante la palabra clave “def”.

3. ¿Cómo crear una aplicación con el manejo de errores y gestión de expresiones regulares?

Para crear una aplicación con el manejo de errores y gestión de expresiones regulares se utilizan las sentencias try, except y finally.

4. ¿Cómo estructurar una base de datos en python?

Para estructurar una base de datos en Python podemos instalar manualmente “MySQL” o sino en una base de datos podemos encontrar la información de manera lógica, la cual, posee un orden de información concisa.

5. ¿Cómo generar una interfaz gráfica con tkinter?

Tkinter es fácil de usar es una multiplataforma donde viene incluido Python en su versión para Windows, para Mac y para la mayoría de las distribuciones GNU/Linux. Así que para generar una interfaz gráfica necesitamos importar el módulo “Tk/Tcl”.

6. ¿Cómo trabajar con módulos, ficheros de texto y binarios en python?

Se trabajan de la siguiente manera:

Módulo: Permite a organizar lógicamente su código Python. Agrupando código relacionado dentro de un módulo hace el código más sencillo.

Fichero de Texto: Un objeto de tipo file también ofrece un método llamado read() . Este método leerá el fichero en su totalidad y lo devolverá como una cadena de texto.

Binarios: Para procesar el archivo de a bytes en lugar de líneas, se utiliza la función contenido = archivo.read(n) para leer n bytes y archivo.write (contenido), para escribir contenido en la posición actual del archivo.

7. ¿Cómo generar documentación automáticamente y trabajar con pruebas automáticas?

Para crear o generar un documentación automática va depender de la cantidad de líneas que implementaras, en el caso de implementar una línea y quieres trabajar con pruebas automáticas, solo se implementara el #, pero en otro caso que sea más líneas de código, se puede implemente ''' (3 apostrofes) o """ (3 comillas).

También podemos instalar programas de terceros dedicados a la automatización, como lo es "Zapier."



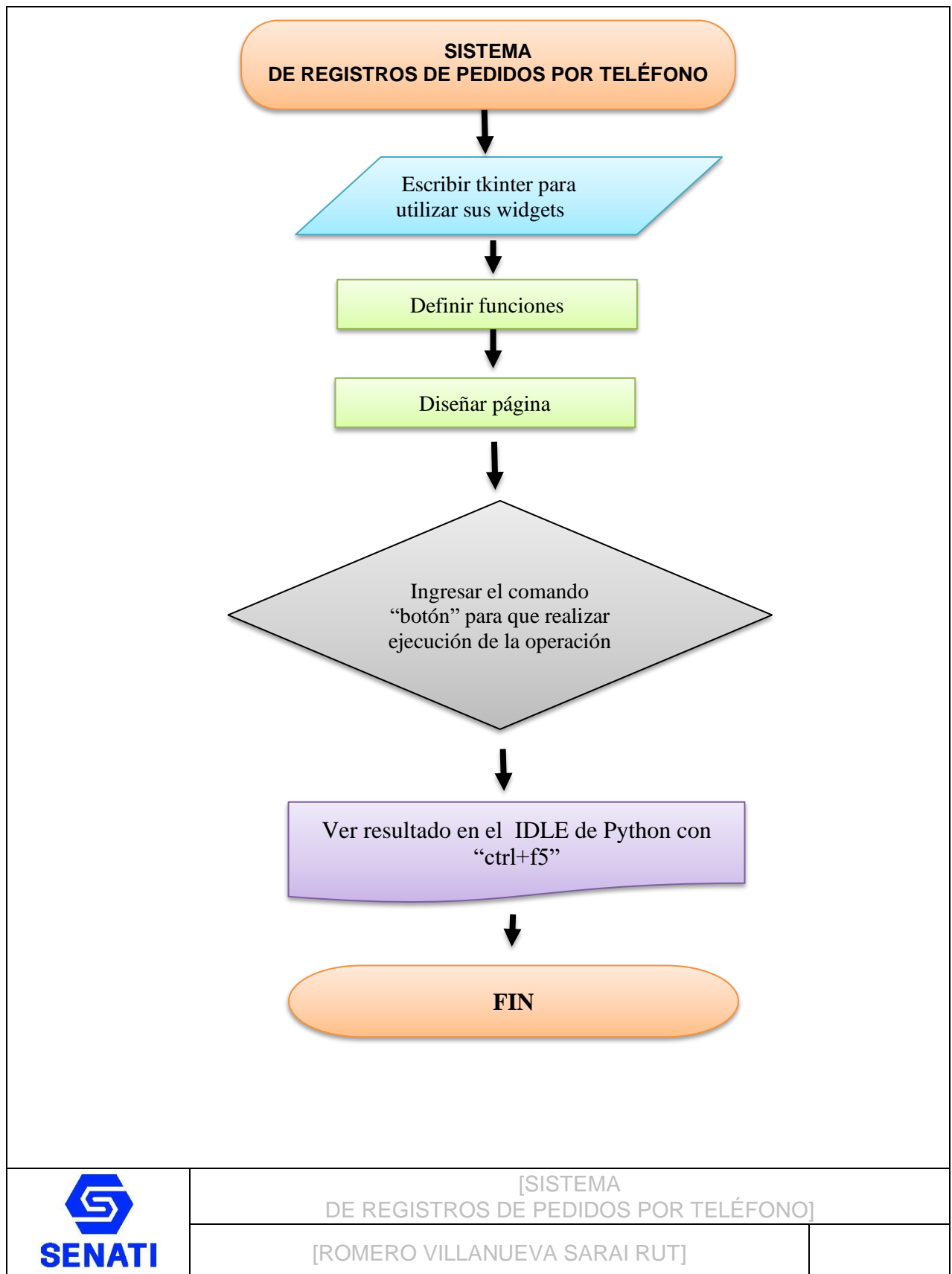
HOJA DE PLANIFICACIÓN

PROCESO DE EJECUCIÓN

[illegible]

INSTRUCCIONES: debes ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. No olvides los aspectos de calidad, medio ambiente y SHI.

DIBUJO / ESQUEMA/ DIAGRAMA

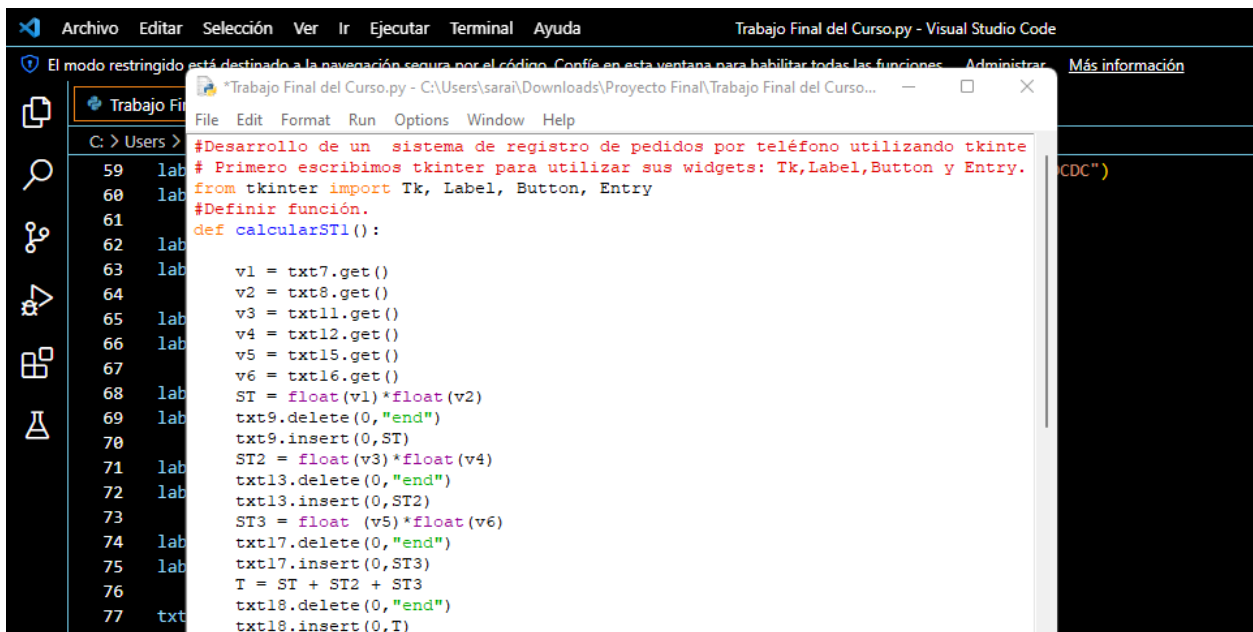


LISTA DE RECURSOS

INSTRUCCIONES: completa la lista de recursos necesarios para la ejecución del trabajo.

1. MÁQUINAS Y EQUIPOS☒ Laptop☒ Celular**3. HERRAMIENTAS E INSTRUMENTOS**☒ Visual Estudy Code☒ IDLE Python☒ Símbolo de Sistema☒ Block de Notas**5. MATERIALES E INSUMOS**☒ Internet☒ Bolígrafo☒ Cuaderno de apuntes

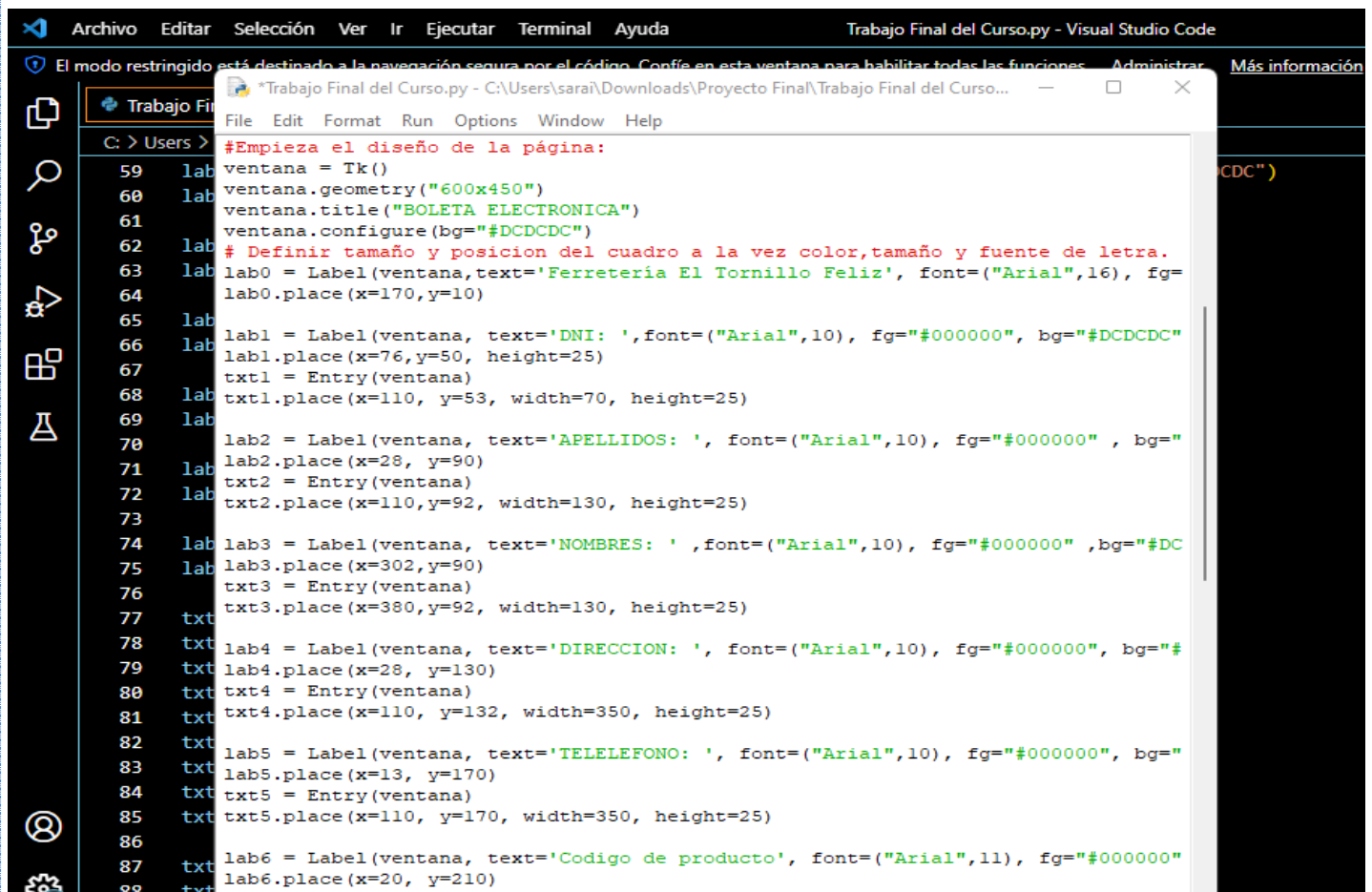
Tkinter-Python:



```

File Edit Format Run Options Window Help
C: > Users > #Desarrollo de un sistema de registro de pedidos por teléfono utilizando tkinte
59 lab # Primero escribimos tkinter para utilizar sus widgets: Tk, Label, Button y Entry.
60 lab from tkinter import Tk, Label, Button, Entry
61 lab #Definir función.
62 lab def calcularST1():
63 lab
64 lab     v1 = txt7.get()
65 lab     v2 = txt8.get()
66 lab     v3 = txt11.get()
67 lab     v4 = txt12.get()
68 lab     v5 = txt15.get()
69 lab     v6 = txt16.get()
70 lab     ST1 = float(v1)*float(v2)
71 lab     txt9.delete(0,"end")
72 lab     txt9.insert(0,ST1)
73 lab     ST2 = float(v3)*float(v4)
74 lab     txt13.delete(0,"end")
75 lab     txt13.insert(0,ST2)
76 lab     ST3 = float(v5)*float(v6)
77 lab     txt17.delete(0,"end")
78 lab     txt17.insert(0,ST3)
79 lab     T = ST1 + ST2 + ST3
80 lab     txt18.delete(0,"end")
81 lab     txt18.insert(0,T)

```



```

File Edit Format Run Options Window Help
C: > Users > #Empieza el diseño de la página:
59 lab ventana = Tk()
60 lab ventana.geometry("600x450")
61 lab ventana.title("BOLETA ELECTRONICA")
62 lab ventana.configure(bg="#DCDCDC")
63 lab # Definir tamaño y posición del cuadro a la vez color,tamaño y fuente de letra.
64 lab lab0 = Label(ventana,text='Ferretería El Tornillo Feliz', font=("Arial",16), fg=
65 lab lab0.place(x=170,y=10)
66 lab lab1 = Label(ventana, text='DNI: ',font=("Arial",10), fg="#000000", bg="#DCDCDC"
67 lab lab1.place(x=76,y=50, height=25)
68 lab txt1 = Entry(ventana)
69 lab txt1.place(x=110, y=53, width=70, height=25)
70 lab lab2 = Label(ventana, text='APELLIDOS: ', font=("Arial",10), fg="#000000", bg="
71 lab lab2.place(x=28, y=90)
72 lab txt2 = Entry(ventana)
73 lab txt2.place(x=110,y=92, width=130, height=25)
74 lab lab3 = Label(ventana, text='NOMBRES: ', font=("Arial",10), fg="#000000", bg="#DC
75 lab lab3.place(x=302,y=90)
76 lab txt3 = Entry(ventana)
77 lab txt3.place(x=380,y=92, width=130, height=25)
78 lab lab4 = Label(ventana, text='DIRECCION: ', font=("Arial",10), fg="#000000", bg="#
79 lab lab4.place(x=28, y=130)
80 lab txt4 = Entry(ventana)
81 lab txt4.place(x=110, y=132, width=350, height=25)
82 lab lab5 = Label(ventana, text='TELEFONO: ', font=("Arial",10), fg="#000000", bg="
83 lab lab5.place(x=13, y=170)
84 lab txt5 = Entry(ventana)
85 lab txt5.place(x=110, y=170, width=350, height=25)
86 lab lab6 = Label(ventana, text='Codigo de producto', font=("Arial",11), fg="#000000"
87 lab lab6.place(x=20, y=210)
88 lab

```


Trabajo Final del Curso.py - Visual Studio Code

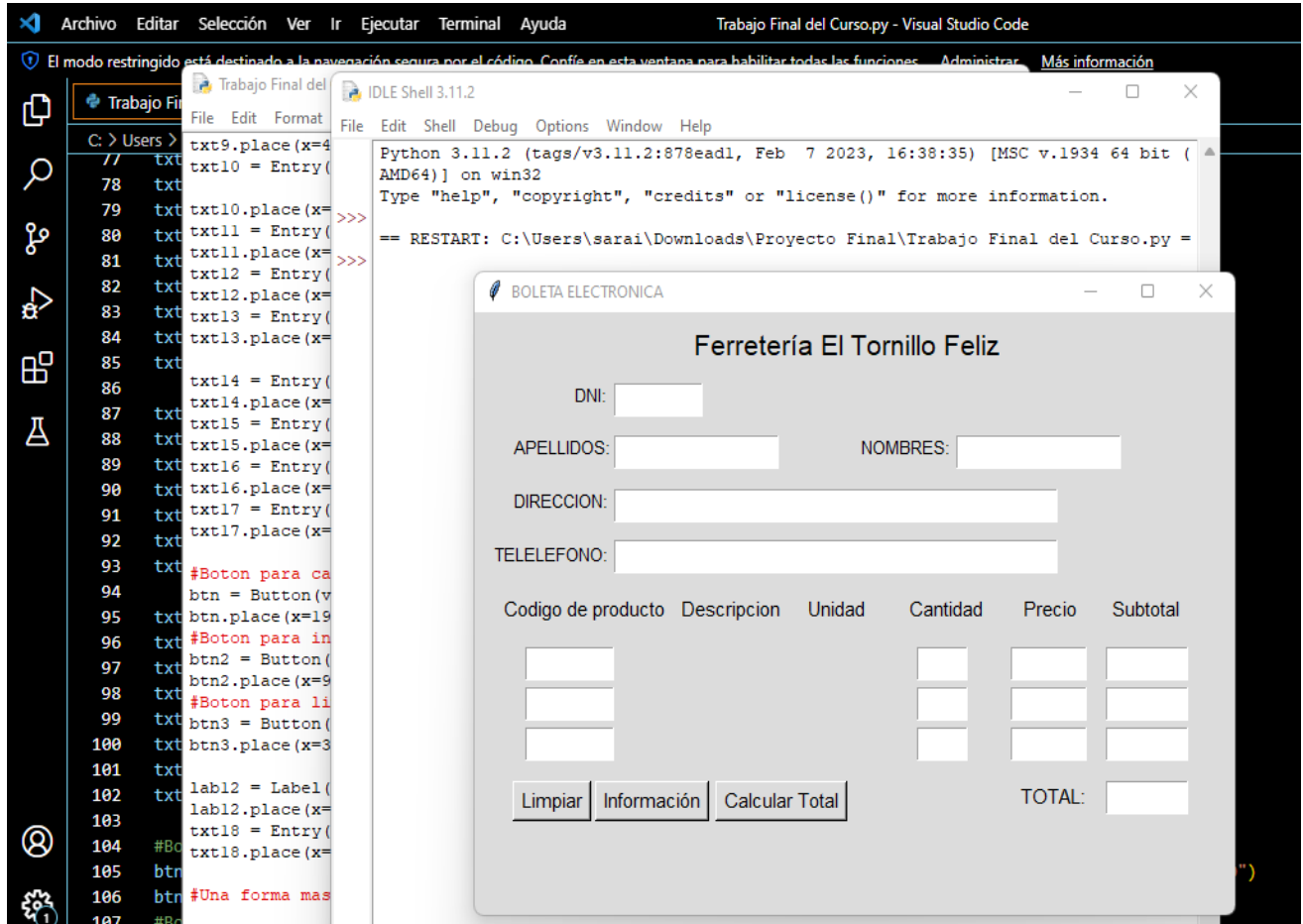
El modo restringido está destinado a la navegación segura por el código. Confíe en esta ventana para habilitar todas las funciones. Administrar Más información

```

C:\Users>
77 txt lab7 = Label(ventana, text='Descripción', font=("Arial",11), fg="#000000", bg="#
78 txt lab7.place(x=160, y=210)
79 txt lab8 = Label(ventana, text='Unidad', font=("Arial",11), fg="#000000", bg="#DCDCDC
80 txt lab8.place(x=260, y=210)
81 txt lab9 = Label(ventana, text='Cantidad', font=("Arial",11), fg="#000000", bg="#DCD
82 txt lab9.place(x=340, y=210)
83 txt
84 txt lab10 = Label(ventana, text='Precio', font=("Arial",11), fg="#000000", bg="#DCDCDC
85 txt lab10.place(x=430, y=210)
86
87 txt lab11 = Label(ventana, text='Subtotal', font=("Arial",11), fg="#000000", bg="#DC
88 txt
89 txt txt6 = Entry(ventana)
90 txt txt6.place(x=40,y=250,width=70, height=25)
91 txt txt7 = Entry(ventana)
92 txt txt7.place(x=349,y=250,width=40, height=25)
93 txt txt8 = Entry(ventana)
94 txt txt8.place(x=423,y=250,width=60, height=25)
95 txt txt9 = Entry(ventana)
96 txt txt9.place(x=498,y=250,width=65, height=25)
97 txt txt10 = Entry(ventana)
98 txt txt10.place(x=40,y=280,width=70, height=25)
99 txt txt11 = Entry(ventana)
100 txt txt11.place(x=349,y=280,width=40, height=25)
101 txt txt12 = Entry(ventana)
102 txt txt12.place(x=423,y=280,width=60, height=25)
103 txt txt13 = Entry(ventana)
104 txt txt13.place(x=498,y=280,width=65, height=25)
105
106 txt txt14 = Entry(ventana)
107 txt txt14.place(x=40,y=310,width=70, height=25)
108 txt txt15 = Entry(ventana)
109 txt txt15.place(x=349,y=310,width=40, height=25)
110 txt txt16 = Entry(ventana)
111 txt txt16.place(x=423,y=310,width=60, height=25)
112 txt txt17 = Entry(ventana)
113 txt txt17.place(x=498,y=310,width=65, height=25)
114
115 txt #Boton para calcular el total:
116 btn = Button(ventana, text='Calcular Total', font=("Arial",11),command = calcula
117 btn.place(x=190, y=350)
118 txt #Boton para información:
119 btn2 = Button(ventana, text='Información', font=("Arial",11),command = calcula
120 btn2.place(x=95, y=350)
121 txt #Boton para limpiar:
122 btn3 = Button(ventana, text='Limpiar', font=("Arial",11),command = calcularST1,
123 btn3.place(x=30, y=350)
124 txt
125 txt lab12 = Label(ventana, text='TOTAL: ', font=("Arial",11), fg="#000000", bg="#DCDC
126 lab12.place(x=428,y=350)
127 txt txt18 = Entry(ventana)
128 #Bot
129 txt txt18.place(x=498, y= 350, width=65, height=25)
130 btn
131 #Una forma mas sencilla para ver el resultado de la creación de la página se cop
132 #Bot

```

Modo restringido



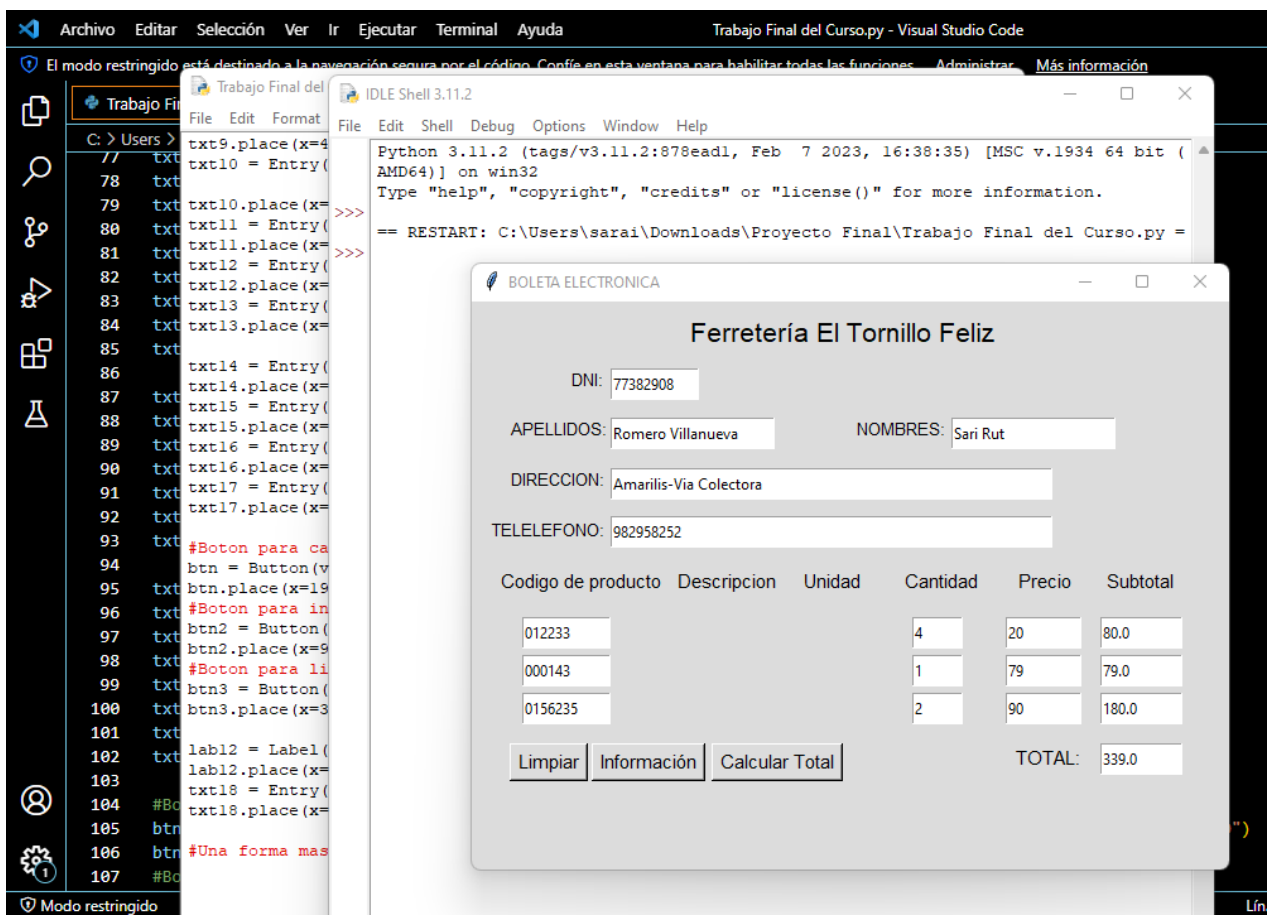
The screenshot shows the Visual Studio Code editor with a Python file named 'Trabajo Final del Curso.py'. The code defines a GUI for a hardware store named 'Ferreteria El Tornillo Feliz'. The GUI includes input fields for DNI, APELLIDOS, NOMBRES, DIRECCION, and TELELEFONO. It also features a table for product details with columns: Codigo de producto, Descripcion, Unidad, Cantidad, Precio, and Subtotal. Below the table are buttons for 'Limpiar', 'Información', and 'Calcular Total', along with a 'TOTAL' label and input field.

The Python code in the background includes the following snippets:

```

txt9.place(x=40, y=15)
txt10 = Entry(
    root, width=30)
txt10.place(x=40, y=20)
txt11 = Entry(
    root, width=30)
txt11.place(x=40, y=25)
txt12 = Entry(
    root, width=30)
txt12.place(x=40, y=30)
txt13 = Entry(
    root, width=30)
txt13.place(x=40, y=35)
txt14 = Entry(
    root, width=30)
txt14.place(x=40, y=40)
txt15 = Entry(
    root, width=30)
txt15.place(x=40, y=45)
txt16 = Entry(
    root, width=30)
txt16.place(x=40, y=50)
txt17 = Entry(
    root, width=30)
txt17.place(x=40, y=55)
txt18 = Entry(
    root, width=30)
txt18.place(x=40, y=60)
#Boton para calcular
btn = Button(
    root, text="Calcular Total")
btn.place(x=190, y=65)
#Boton para limpiar
btn2 = Button(
    root, text="Limpiar")
btn2.place(x=90, y=65)
#Boton para informacion
btn3 = Button(
    root, text="Información")
btn3.place(x=300, y=65)
lab12 = Label(
    root, text="TOTAL:")
lab12.place(x=750, y=65)

```



This screenshot shows the same Visual Studio Code environment, but with sample data entered into the GUI. The input fields are populated with the following values:

- DNI: 77382908
- APELLIDOS: Romero Villanueva
- NOMBRES: Sari Rut
- DIRECCION: Amarilis-Via Colectora
- TELELEFONO: 982958252

The product table is also populated with three items:

Codigo de producto	Descripcion	Unidad	Cantidad	Precio	Subtotal
012233			4	20	80.0
000143			1	79	79.0
0156235			2	90	180.0

The 'TOTAL' field now displays 339.0. The Python code in the background remains the same as in the previous screenshot.