

**“AÑO DE LA UNIDAD, LA PAZ Y EL DESARROLLO”**



**NOMBRE:** Amelia

**APELLIDO:** Martel Nazario

**CARRERA:** Ingeniería de Software con Inteligencia Artificial

**CURSO:** Algoritmia de Programación del Software

**SEMESTRE:** II

**INSTRUCTOR:** Alexander Dominguez Pilco

**TEMA:** Creación de aplicación básica en Python

**TEMA**: **Creación de Aplicación básicas en Python**

OBJETO DEL TRABAJO

Evaluar la capacidad del estudiante para desarrollar aplicación básica en Python y herramientas elementos.

**PLANTEMIENTO DEL TRABAJO**

Debido a la Pandemia la Ferretería El Tornillo Feliz” desea implementar un sistema de registro de pedidos por teléfono, para lo cual necesitar registrar el DNI, nombres y apellidos, la dirección, teléfono y el pedido que se realiza. Al solo contar con una computadora y escasos recursos, decide utilizar el Python.

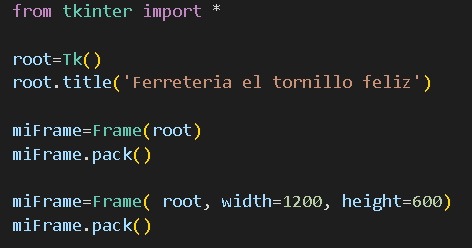
Este sistema debe permitir no solo registrar estos datos, si no también imprimir para que el repartidor los lleve sin problema.



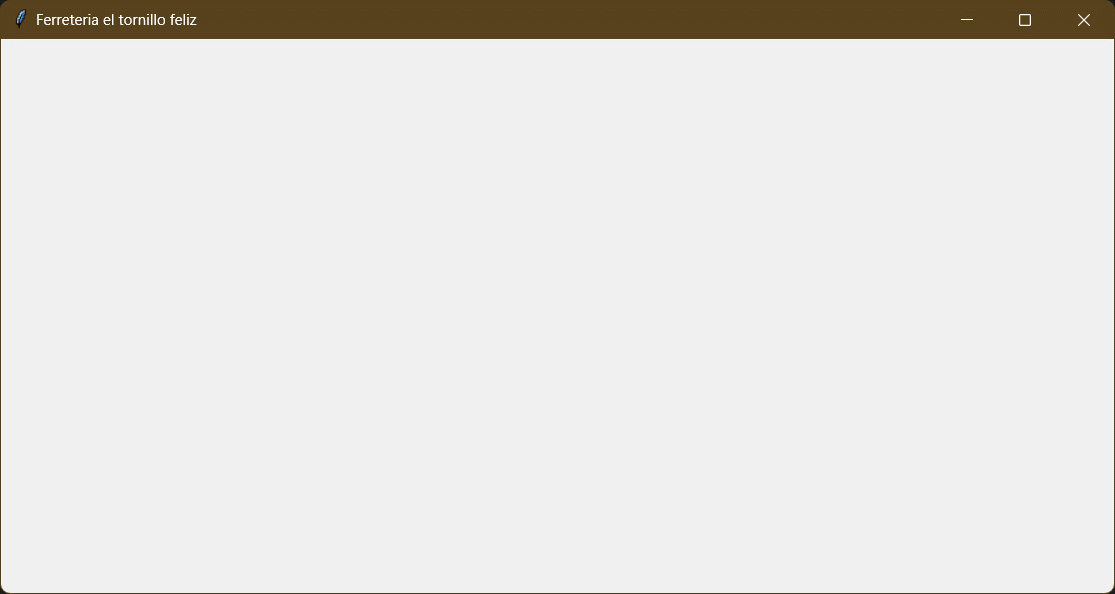
**PROCEDIMIENTO PARA RESOLVER EL PROBLEMA PLANTEADO.**

1. Primer paso tener instalado todas mis herramientas necesitado para asi no tener problema.

Para trabajar con Python y Tkinter use los siguientes valores, en los cuales asigno un ancho de 1200 y una altura de 600.



Y de esta forma es como se visualiza al momento de ejecutar.



1. Se uso los siguientes comandos para añadir los cuadros que van a aparecer en la interfaz.

cuadroDNI=Entry(miFrame)

cuadroDNI.grid(row=0, column=1, padx=10, pady=10)

cuadroDNI.config(fg="blue")

cuadroApellido=Entry(miFrame)

cuadroApellido.grid(row=1, column=1, padx=10, pady=10)

cuadroNombre=Entry(miFrame)

cuadroNombre.grid(row=1, column=3, padx=10, pady=10)

cuadroDirección=Entry(miFrame)

cuadroDirección.grid(row=2, column=1,columnspan=3,sticky="we",  padx=10, pady=10)

cuadroTeléfono=Entry(miFrame)

cuadroTeléfono.grid(row=3, column=1, columnspan=3,sticky="we", padx=10, pady=10)

Nos sale de la siguiente forma la interfaz



Seguidamente agregamos los nombres a la parte superior de la interfaz

DNILabel=Label(miFrame, text="DNI:")

DNILabel.grid(row=0, column=0, sticky="w", padx=10, pady=10)

ApellidoLabel=Label(miFrame, text="Apellido:")

ApellidoLabel.grid(row=1, column=0, sticky="w", padx=10, pady=10)

NombreLabel=Label(miFrame, text="Nombre:")

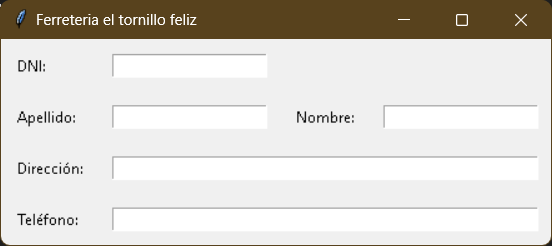
NombreLabel.grid(row=1, column=2, sticky="w", padx=10, pady=10)

DirecciónLabel=Label(miFrame, text="Dirección:")

DirecciónLabel.grid(row=2, column=0, sticky="w", padx=10, pady=10)

TeléfonoLabel=Label(miFrame, text="Teléfono:")

TeléfonoLabel.grid(row=3, column=0, sticky="w", padx=10, pady=10)



Finalmente se agrega la parte inferior de la interfaz

cod\_prodLabel=Label(miFrame,text="cod\_prod")

cod\_prodLabel.grid(row=4, column=0, sticky="e", padx=10, pady=10)

cod\_prod=Entry(miFrame, width=7)

cod\_prod.grid(row=5, column=0, padx=10, pady=10)

cod\_prod=Entry(miFrame, width=7)

cod\_prod.grid(row=6, column=0, padx=10, pady=10)

cod\_prod=Entry(miFrame, width=7)

cod\_prod.grid(row=7, column=0, padx=10, pady=10)

DescripciónLabel=Label(miFrame,text="Descripción")

DescripciónLabel.grid(row=4, column=1, sticky="ew", padx=10, pady=10)

Descripción=Entry(miFrame, width=7, state="readonly")

Descripción.grid(row=5, column=1, padx=10, pady=10)

Descripión=Entry(miFrame, width=7, state="readonly")

Descripión.grid(row=6, column=1, padx=10, pady=10)

Descripción=Entry(miFrame, width=7, state="readonly")

Descripción.grid(row=7, column=1, padx=10, pady=10)

UnidadLabel=Label(miFrame,text="Unidad")

UnidadLabel.grid(row=4, column=2, sticky="ew", padx=10, pady=10)

Unidad=Entry(miFrame, width=7, state="readonly")

Unidad.grid(row=5, column=2, padx=10, pady=10)

Unidad=Entry(miFrame, width=7, state="readonly")

Unidad.grid(row=6, column=2, padx=10, pady=10)

Unidad=Entry(miFrame, width=7, state="readonly")

Unidad.grid(row=7, column=2, padx=10, pady=10)

CantidadLabel=Label(miFrame,text="Cantidad")

CantidadLabel.grid(row=4, column=3, sticky="ew", padx=10, pady=10)

Cantidad=Entry(miFrame, width=7)

Cantidad.grid(row=5, column=3, padx=10, pady=10)

Cantidad=Entry(miFrame, width=7)

Cantidad.grid(row=6, column=3, padx=10, pady=10)

Cantidad=Entry(miFrame, width=7)

Cantidad.grid(row=7, column=3, padx=10, pady=10)

PrecioLabel=Label(miFrame,text="Precio")

PrecioLabel.grid(row=4, column=4, sticky="ew", padx=10, pady=10)

Precio=Entry(miFrame, width=7, state="readonly")

Precio.grid(row=5, column=4, padx=10, pady=10)

Precio=Entry(miFrame, width=7, state="readonly")

Precio.grid(row=6, column=4, padx=10, pady=10)

Precio=Entry(miFrame, width=7, state="readonly")

Precio.grid(row=7, column=4, padx=10, pady=10)

SubtotalLabel=Label(miFrame,text="Subtotal")

SubtotalLabel.grid(row=4, column=5, sticky="ew", padx=10, pady=10)

Subtotal=Entry(miFrame, width=7, state="readonly")

Subtotal.grid(row=5, column=5, padx=10, pady=10)

Subtotal=Entry(miFrame, width=7, state="readonly")

Subtotal.grid(row=6, column=5, padx=10, pady=10)

Subtotal=Entry(miFrame, width=7, state="readonly")

Subtotal.grid(row=7, column=5, padx=10, pady=10)

TotalLabel=Label(miFrame,text="Total")

TotalLabel.grid(row=7, column=6, sticky="ew", padx=10, pady=10)

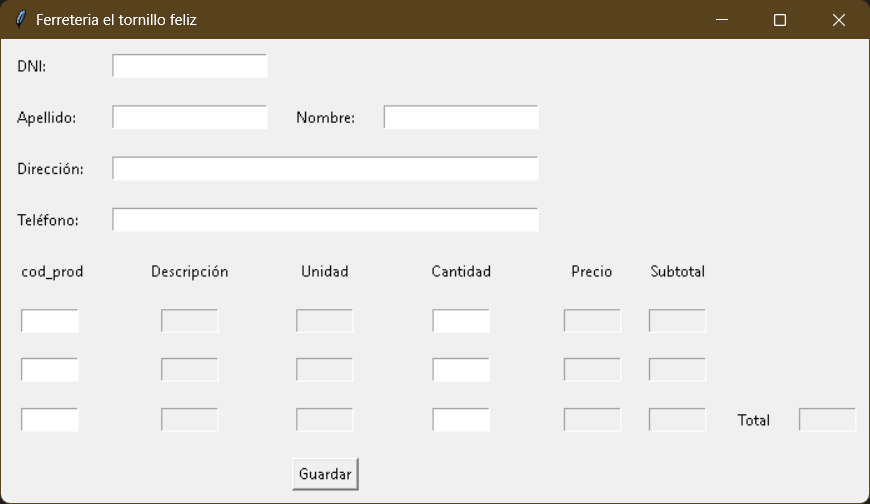
Total=Entry(miFrame, width=7, state="readonly")

Total.grid(row=7, column=7, padx=10, pady=10)

Guardar=Button(miFrame, text="Guardar")

Guardar.grid(row=8, column=2, padx=10, pady=10)

Con lo que finalmente la interfaz se ve de la siguiente manera.



**PREGUNTAS**

**1. ¿Cómo se maneja las Cadenas, Operaciones y se crea colecciones GUÍA de datos e Python?**

**Las cadenas** se manejan en Python para un tipo inmutable de datos que permite almacenar información. Para crear una cadena se debe escribir el texto entre comillas dobles "\*\*”.

**Las operaciones** sirven para efectuar las operaciones aritméticas conocidas tales como suma ( + ), resta (-), multiplicación (x), división (/); entre otros.

**Las colecciones** son las que se usan para almacenar objetos en una misma variable con un mismo nombre, existen 3 tipos: **listas, diccionarios y tuplas**.

**2. ¿Cómo generar Clases, Objetos y Funciones en Python?**

**Las clases** se crean con la palabra reservada “class”, con el cual se asignan un nombre a la clase.

**Los objetos** se crean a partir de una determinada clase, se usa el nombre de la clase y luego se añade paréntesis.

**Las funciones** se crean con la palabra reservada def, seguida de un nombre que se le quiere dar a la función.

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

El uso de las expresiones regulares en Python viene dado por el paquete re, que hay que importar a nuestro código. Algunos de los métodos proporcionados en este paquete son “re.search”, “re.findall” y “re.split”.

**4. ¿Cómo estructurar una Base de Datos en Python?**

Python es un lenguaje que puede sincronizar un ordenador con una base de datos, la base de dato se crea en diferentes programas como por ejemplo MySql.

**5. ¿Cómo generar una interfaz Gráfica con Tkinter?**

El paquete Tkinter debe ser importado para así poder hacer uso de sus códigos y funciones que este ofrece, se utiliza los códigos “from” e “import”.

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

Los módulos son archivos de Python cuyos objetos (funciones, clases, excepciones, etc.) pueden ser accedidos desde otro archivo, se utiliza el código “import” para así acceder al archivo.

Para abrir un fichero, la más común es utilizar la función open del módulo io.

**7. ¿Cómo generar documentación Automáticamente y trabajar con Pruebas Automáticas?**

El módulo “pydoc" genera automáticamente documentación de módulos de Python. La documentación se puede presentar como páginas de texto en la consola, servidos en un buscador web, o guardados en archivos HTML y para realizar pruebas se puede utilizar el módulo “doctest”.