



FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

Cátedra: ALGORITMO II.

Docente: Lic. Pedro Antonio Trejo Noble

Investigación: programa de un inventario creado en python

Autores: David Antonio Leiva de León

Gerardo Alejandro Rodríguez

Rubén Ernesto Mejía Cortez

San salvador, 10 de junio de 2018

INTRODUCCION

En el presente proyecto planteamos un software para empresas que se dedican a las ventas tomando en cuenta todas las necesidades de cada una de ellas, sabiendo también que los usuarios buscan interfaz gráfica fácil de usar.

Nos basamos en las principales funciones como: guardar, modifica, borrar.

Dado que se manejan una gran cantidad de productos buscamos almacenar de manera más eficaz sin hacer que los procesos hagan ineficiente el software y el procesamiento del mismo una pequeña base de datos hace fácil de cumplir este propósito

Esperamos satisfacer a los diferentes usuarios que obtengan este software.

OBJETIVOS

Objetivo General:

- ✓ Crear un programa el cual consiste en almacenar, borrar y mostrar los productos de un Inventario de una tienda.

Objetivo Específicos:

- ✓ Detallar el procedimiento de dicho programa
- ✓ Mostrar una aplicación funcional para empresas que comercializan diferentes productos

ANTECEDENTES

El concepto de inventario hoy en día es un término cotidianamente utilizado, dentro de las empresas sin importar su tamaño o giro, pero desde cuándo se puede hablar de la aparición de este concepto y es que desde tiempos inmemorables pueblos de la antigüedad como los Egipcios solían almacenar grandes cantidades de alimentos para ser utilizados en épocas de sequía y escasez, de esta forma surge el dilema de llevar un registro, distribución y cuidado de los recursos, para hacer frente a la travesía de sobrevivir.

En América tendría sus antecedentes hacia el 2.500 A.C con el pueblo Inca ya que crearon una herramienta llamada Quipu se consideran vestigios de inventarios por su uso en registros, censos, contabilidad de cosechas, reservas de producción agrícola, cantidad de productos de la minería etc. .

Con el paso del tiempo y hacia la Segunda Guerra Mundial, con el surgimiento de grandes y complejas computadoras nacen los primeros sistemas para la planificación de inventario y material.

Para el final de los años 50, los sistemas de inventarios como medio de la planificación del requerimiento de material bélico y como gran parte de los adelantos y modernidades de hoy en día, se transformaron su uso en las guerras, para hallar cabida en los sectores productivos, en especial en los Estados Unidos de Norte América.

Con el adelanto tecnológico y diversificado surgieron sistemas para manejo de inventarios, incrementando el número de empresas que hacían uso de estas herramientas.

En nuestro país el surgimiento de empresa o ventas de artículos con fines lucrativos se remontan a la época prehispánica en donde se instalaban en espacios abiertos donde se comercializaban metales preciosos, especies, alimentos, ropa y mercancía para rituales religiosos.

En cuanto al sistema de almacenamiento del inventario, se colocaba la mercancía en montones o se guardaba en bodegas localizadas en la parte trasera de los comercios, de tal forma que los empleados y/o vendedores, tenían que desplazarse para entregar los pedidos solicitados por los clientes.

JUSTIFICACIÓN

El presente proyecto se enfocara en crear un programa en el lenguaje de programación python el cual consistirá en llevar un inventario de x almacén o tienda ya que en los almacenes se maneja mercancía día tras día y es muy cambiante la variedad de los productos.

Debido a esta gran necesidad de cambio, se desarrollara la automatización de las actividades que se realizan en los diferentes procesos del departamento de Servicio General como es el registro de entradas y salidas del inventario de almacén, requisiciones de compras y/o servicios, solicitud y entrega de bienes, materiales y suministros.

Al respecto hay que tener en cuenta que una de las razones que determina la importancia y justificación de desarrollar un sistema automatizado para este departamento, es la de registrar y controlar los procesos, para dar así una respuesta inmediata a los diferentes departamentos que soliciten información y/o recursos, en el momento en que lo requieran evitando pérdida de tiempo.

El sistema que se desea desarrollar es de suma importancia porque con él se busca la optimización de los procesos actuales permitiendo conocer en tiempo real la información administrativa del departamento como es, cada uno de los bienes que posee la empresa, en que departamentos están ubicados, cuales están en uso o han sido desincorporados, a su vez permitirá controlar el inventario del almacén, lo cual facilitara saber con qué recursos cuentan la empresa, que materiales se han entregado a los diferentes departamentos, con que material se dispone y que departamento tiene pedidos realizados y faltan por despachar, toda esta información es de suma importancia y la misma con la optimización de los procesos podrá ser suministrada de manera oportuna, confiable y veraz.

PLANTIAMIENTO DEL PROBLEMA

Con respecto a la necesidad vista en las tiendas pequeñas donde no se cuenta con grandes inventarios se tomó la decisión de crear el siguiente programa creado en python con las siguientes características.

- ✓ Utilizaremos una pequeña base de datos para la ejecución del programa
- ✓ Que podamos guardar los productos de nuestro inventario
- ✓ Que los productos almacenados sean visibles para los usuarios
- ✓ Que se pueda colocar su precio
- ✓ Por último que podamos borrar los productos si así lo deseamos

Para poder llenar las necesidades de los clientes como usuarios necesitamos un software que pueda brindar las comodidades y accesibilidad para no volverlo complicado de manipular las diferentes opciones que se brindaran.

Un entorno grafico facilitara el trabajo para los usuarios y disponer de opciones sin tener que andarlas buscando.

Conocer en tiempo real la información administrativa del departamento como es, cada uno de los bienes que posee la empresa, en que departamentos están ubicados, cuales están en uso o han sido desincorporados, a su vez permitirá controlar el inventario del almacén, lo cual facilitara saber con qué recursos cuentan la empresa, que materiales se han entregado a los diferentes departamentos, con que material se dispone y que departamento tiene pedidos realizados y faltan por despachar, toda esta información es de suma importancia y la misma con la optimización de los procesos podrá ser suministrada de manera oportuna, confiable y veraz.

MARCO REFERENCIAL

La aplicación se desarrolla en Python es el lenguaje de programación muy versátil.

Algunas de las características que lo hacen interesante para nosotros son:

- ✓ Es fácil de utilizar.
- ✓ Tiene gran variedad de estructuras de datos incorporadas al propio lenguaje.
- ✓ Tiene una gran cantidad de bibliotecas (librerías).
- ✓ Permite la programación modular, orientada a objetos y su uso como un lenguaje imperativo tradicional.
- ✓ Es interpretado. Esto facilita el desarrollo (aunque ralentice la ejecución).
- ✓ Se puede utilizar desde un entorno interactivo.
- ✓ Se puede extender fácilmente.
- ✓ Es muy expresivo: un programa Python ocupa mucho menos que su equivalente en C.

La versión de Python que se utilizo es 3.6.4 que hasta la fecha es la más actual

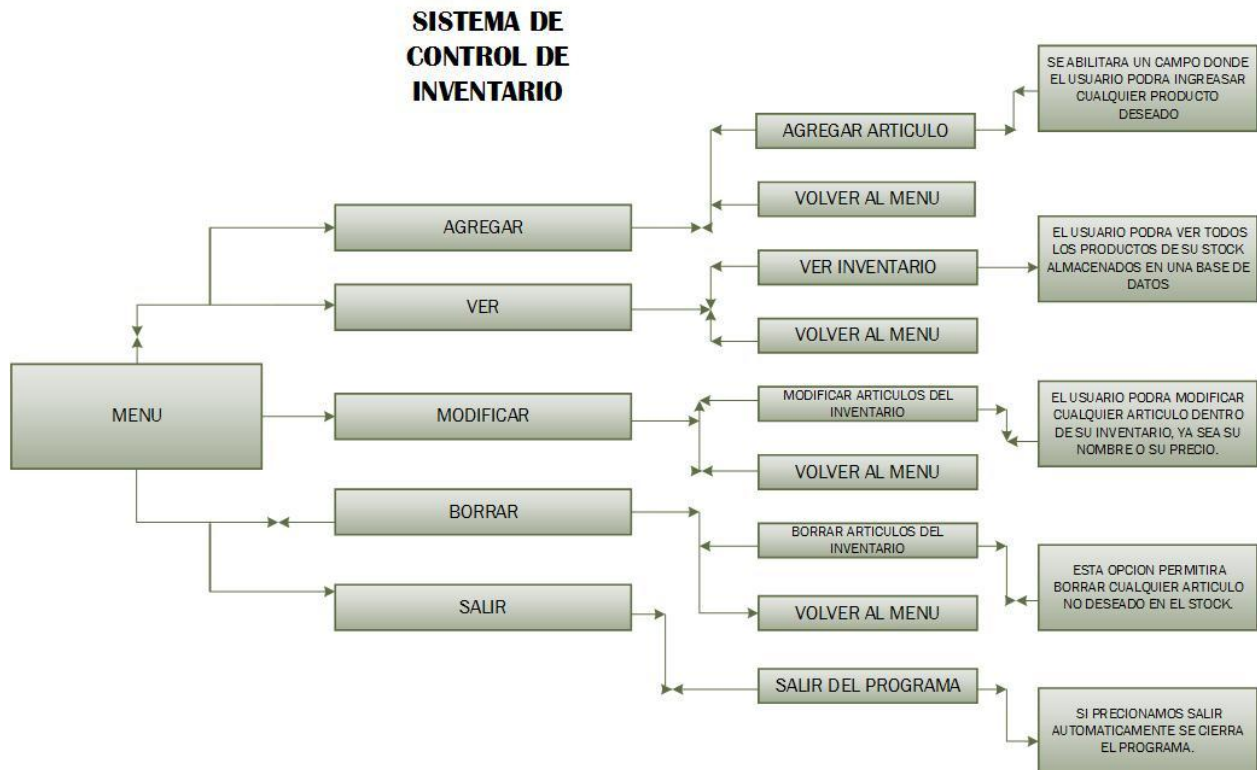
También para guardar los datos que el usuario va a ingresar se hace uso de SQLite 3 que es un gestor de base de datos que aparte de ser fácil de usar nos da la facilidad de crear tablas en entorno grafico.

El motor de SQLite no es un proceso independiente con el que el programa principal se comunica , en lugar de eso la biblioteca de SQLite se enlaza con el programa pasando a ser parte integral del mismo.



MARCO CONCEPTUAL

La creación de un diagrama de actividades para tener una base para saber paso a paso lo que nuestra aplicación hará.



Partimos desde un menú que será la ventana principal en el cual estarán 4 funciones principales lo que la aplicación ofrece

- ✓ Agregar
- ✓ ver
- ✓ Modificar
- ✓ Borrar o Eliminar

Cada una de esas tendrá una ventana secundaria la cual le pedirá al usuario la información necesaria para poder ejecutar la accione elegida

En la creación de la base de datos con SQLite 3 creamos la tabla con los campos necesarios o requeridos para cada modificación que se hará en dicha base de datos

Editar Tabla

Nombre Tabla:

Campos Tabla

Nombre Campo	Tipo Campo	Valor Por Omisión	Constraints del ...
codigo	INTEGER		NOT NULL PRIM...
nombre	varchar(30)		NULL
precio	varchar(30)		NULL

Llave Primaria Agregar Campo Editar Campo Eliminar Campo

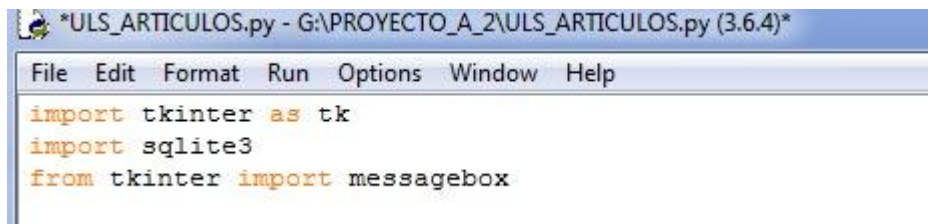
Editar Cancelar

Una tabla con tres campos código, nombre, precio el primer campo es la llave primaria que se le dejo con auto incremento para no estar ingresando un código por cada producto .

Los campos Nombre y Precio se dejaron con tipo de dato VARCHAR con un maximo de caracteres de 30 asi nos queda un archivo con **extencion.s3db**



Para la creación de la aplicación en Python 3 necesitamos hacer las importaciones de las librerías necesarias para su adecuada ejecución



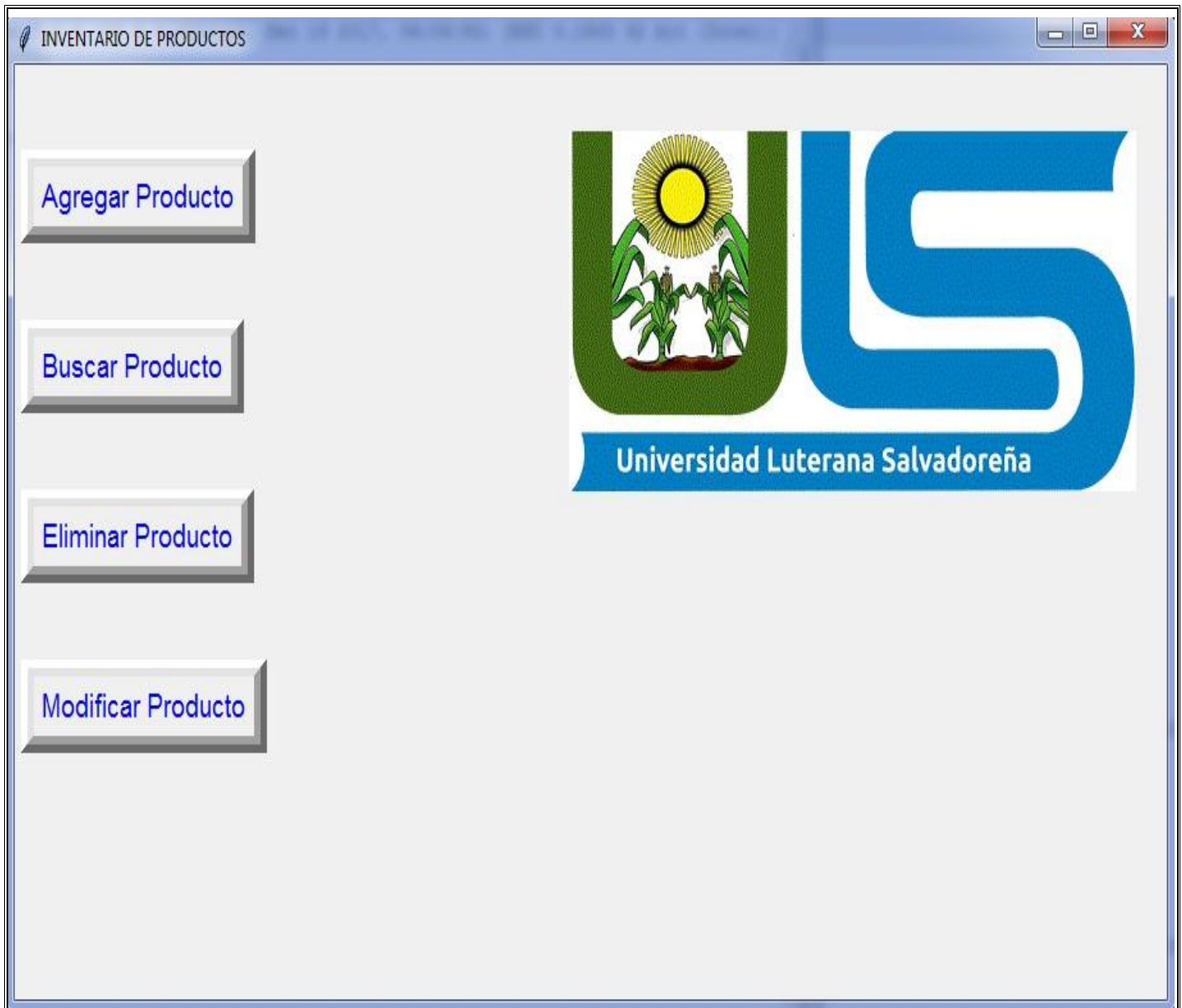
```
*ULS_ARTICULOS.py - G:\PROYECTO_A_2\ULS_ARTICULOS.py (3.6.4)*
File Edit Format Run Options Window Help
import tkinter as tk
import sqlite3
from tkinter import messagebox
```

- ✓ La librería de Tkinter
- ✓ la librería de sqlite3 (Para la base de datos)
- ✓ MessageBox que es un método de la librería de Tkinter para mostrar pequeñas ventanas flotantes de errores y avisos

La ventana principal que sería el menú con solo 4 botones y el logo ULS

```
windows=tk.Tk()
windows.title("INVENTARIO DE PRODUCTOS")
windows.geometry("900x550")
## Boton para Agregar Producto
b1=tk.Button(windows,text="Agregar Producto", fg="blue", font=("arial", 14), borderwidth=10, command = ventana_agregar)
b1.pack()
b1.place(x=5,y=50)
## Boton para Buscar producto
b2=tk.Button(windows,text="Buscar Producto", fg="blue", font=("arial", 14), borderwidth=10, command = ventana_ver)
b2.pack()
b2.place(x=5,y=150)
## Boton para Eliminar Producto
b3=tk.Button(windows,text="Eliminar Producto", fg="blue", font=("arial", 14), borderwidth=10, command = ventana_eliminar)
b3.pack()
b3.place(x=5,y=250)
## Boton para Modificar Producto
b4=tk.Button(windows,text="Modificar Producto", fg="blue", font=("arial", 14), borderwidth=10, command = modificar_producto)
b4.pack()
b4.place(x=5,y=350)
## Se importa la imagen de logo ULS
image=tk.PhotoImage(file="uls.gif")
image=image.subsample(1,1)
label=tk.Label(image=image)
label.place(x=205, y=-130, relwidth=1, relheight=1)

windows.mainloop()
```



Cada uno de los botones realiza una acción que se activa cuando el usuario hace click sobre alguno la palabra **command** ejecuta la instrucción y abre una función que corresponde a ese botón y tiene una ventana independiente en este caso Guardad producto

```
12),command = guarda)
```

Ventana agregar producto (Código)

```
def ventana_agregar():
    ## windows.withdraw()
    window=tk.Toplevel()
    window.geometry("512x512")
    e1=tk.Label(window, text="AGREGAR PRODUCTOS :",bg="white",fg="black").place(x=50, y=50),

    # variable producto
    entryproducto=tk.StringVar()
    productotx=tk.Entry(window,textvariable=entryproducto).place(x=50, y=150)

    # variable precio
    entryprecio=tk.StringVar()
    preciottx=tk.Entry(window,textvariable=entryprecio).place(x=50, y=230)

    # Etiqueta para "INGRESE NOMBRE DEL PRODUCTO"
    etiquetanombre=tk.Label(window, text="INGRESE NOMBRE DEL PRODUCTO.", padx=10 ).place(x=30, y=115)

    #Etiqueta para "INGRESE PRECIO DEL PRODUCTO"
    etiquetaprecio = tk.Label(window, text="INGRESE PRECIO DEL PRODUCTO", padx=10 ).place(x=30, y=200)

    ## Boton menu regresa a la pantalla principal

    menu=tk.Button(window, text="MENU", fg="red",font=("arial", 12),command = windows.destroy)
    menu.pack()
    menu.place(x=50,y=350)

    ## Funcion para coneccion a la base de datos
    def guarda():

        db = sqlite3.connect("articulos.s3db")
        c = db.cursor()

        nombre = entryproducto.get()
        precio = entryprecio.get()

        c.execute("insert into articulos (nombre,precio) values ('"+nombre+"','"+precio+"')")
        db.commit()
        c.close()
        messagebox.showwarning("MODIFICACION","ARTICULO INGRESADO" )

    ## Boton 'Guardar' que ejecuta la funcion para la coneccion a la base de datos
    btguardar = tk.Button(window, text = "GUARDAR", fg="blue",font=("arial", 12),command = guarda)
    btguardar.pack()
    btguardar.place(x=300,y=350)
```

Ventana Agregar Producto (Interfaz Grafica)

INVENTARIO DE PRODUCTOS

AGREGAR PRODUCTOS :

INGRESE NOMBRE DEL PRODUCTO.

INGRESE PRECIO DEL PRODUCTO

MENU

GUARDAR

Ventana Ver Producto (Código)

```
def ventana_ver():

    window=tk.Toplevel()
    window.geometry("512x512")
    e1=tk.Label(window, text="BUSCAR PRODUCTOS :",bg="white",fg="black").place(x=50, y=50),
## Etiquetas codigo, nombre, precio
    e_codigo=tk.Label(window, text="CODIGO",bg="white",fg="black").place(x=50, y=70)
    e_nombre=tk.Label(window, text="NOMBRE",bg="white",fg="black").place(x=150, y=70)
    e_precio=tk.Label(window, text="PRECIO",bg="white",fg="black").place(x=250, y=70)
## Coneccion a la base de datos para mostrar
    def mostrar():
##         Creacion de la listabox para codigo
        lista=tk.Listbox(window, width = 30, font=("arial", 12), height =15 )
        lista.pack()
        db = sqlite3.connect("articulos.s3db")
        c = db.cursor()
        c.execute("select * from articulos ORDER BY (codigo)DESC")
##         Interacion para insertar la consulta
        for row in c:
            lista.insert(0,row[1]+"-----"+ row[2])
        lista.place(x=150,y=90)
##         nombre
        lista_1=tk.Listbox(window, width = 10, font=("arial", 12), height =15 )
        lista_1.pack()
        db = sqlite3.connect("articulos.s3db")
        c = db.cursor()
        c.execute("select codigo from articulos ORDER BY (codigo)DESC")
        for row in c:
            lista_1.insert(0,row[0])
        lista_1.place(x=50,y=90)
##         Boton 'menu' para regresar a la pantalla principal
        menu=tk.Button(window, text="MENU", fg="red",font=("arial", 12),command = window.destroy)
        menu.pack()
        menu.place(x=50,y=400)
##         Boton para ejecutar la funcion del la consulta
        bt_mostrar = tk.Button(window, text = "MOSTRAR PRODUCTOS", fg="blue",font=("arial", 12),command = mostrar)
        bt_mostrar.pack()
        bt_mostrar.place(x=280,y=400)
```

Ventana Ver Producto (Interfaz Grafica)

INVENTARIO DE PRODUCTOS

BUSCAR PRODUCTOS :

CODIGO	NOMBRE	PRECIO
4	libro-----	5
5	llave-----	10
7	jabon -----	.035
10	pollo-----	0.50
16	papas-----	0.50
17	zanahorias-----	0.35
18	ayote-----	0.50
19	ejotes-----	2.00

MENU

MOSTRAR PRODUCTOS

Ventana Eliminar Producto (Código)

```
def ventana_eliminar():
    window=tk.Toplevel()
    window.geometry("512x512")
    e1=tk.Label(window, text=" ELIMINAR PRODUCTOS :",bg="white",fg="black").place(x=50, y=50)

#### VARIABLE PARA ID

    entry_id=tk.StringVar()
    productotx=tk.Entry(window,textvariable=entry_id).place(x=50, y=150)

#### ETIQUETA PARA ID

    etiquetanombre=tk.Label(window, text="INGRESE CODIGO DEL PRODUCTO", padx=10 ).place(x=30, y=115)
#### Funcion para la coneccion a la base de datos
    def eliminar():
        db = sqlite3.connect("articulos.s3db")
        c = db.cursor()

        id_producto = entry_id.get()

        c.execute("DELETE  from articulos where codigo = ('"+id_producto+"')")
        db.commit()
        c.close()
        messagebox.showwarning("MODIFICACION","ARTICULO ELIMINADO" )

#### BOTON PARA MENU

    menu=tk.Button(window, text="MENU", fg="red",font=("arial", 12),command = window.destroy)
    menu.pack()
    menu.place(x=50,y=350)
#### BOTON PARA ELIMINAR
    bt_eliminar = tk.Button(window, text = "ELIMINAR PRODUCTOS", fg="blue",font=("arial", 12),command = eliminar)
    bt_eliminar.pack()
    bt_eliminar.place(x=280,y=350)
```


Ventana Eliminar Producto (Interfaz Grafica)

INVENTARIO DE PRODUCTOS

ELIMINAR PRODUCTOS :

INGRESE CODIGO DEL PRODUCTO

MENU

ELIMINAR PRODUCTOS

Ventana Modificar Producto (Código)

```
def modificar_producto():
    window=tk.Toplevel()
    window.geometry("512x512")
    e1=tk.Label(window, text=" MODIFICAR PRODUCTOS :",bg="white",fg="black").place(x=50, y=50)

#### VARIABLE PARA ID

    entry_id=tk.StringVar()
    productotx=tk.Entry(window,textvariable=entry_id).place(x=50, y=150)

#### ETIQUETA PARA ID

    etiquetanombre=tk.Label(window, text="INGRESE CODIGO DEL PRODUCTO", padx=10 ).place(x=30, y=100)

#### VARIABLE PARA VALOR NUEVO

    entry_valor=tk.StringVar()
    valortx=tk.Entry(window,textvariable=entry_valor).place(x=50, y=250)

#### ETIQUETA PARA NUEVO VALOR

    etiquetanombre=tk.Label(window, text="INGRESE EL NUEVO PRECIO PARA EL PRODUCTO", padx=10 ).place(x=30, y=200)
## FUNCION PARA LA CONECCION A LA BASE DE DATOS
def modificar():
    db = sqlite3.connect("articulos.s3db")
    c = db.cursor()

    id_producto = entry_id.get()
    nuevo_precio = entry_valor.get()

    c.execute("update articulos set precio =('"+nuevo_precio+"' ) where codigo = ('"+id_producto+"')")
    db.commit()
    c.close()
    messagebox.showwarning("MODIFICACION", "ARTICULO MODIFICADO" )

#### BOTON PARA MENU

    menu=tk.Button(window, text="MENU", fg="red",font=("arial", 12),command = window.destroy)
    menu.pack()
    menu.place(x=50,y=350)
#### BOTON PARA EJECUTAR LA FUNCION MODIFICAR
    bt_modificar = tk.Button(window, text = "MODIFICAR PRODUCTO", fg="blue",font=("arial", 12),command = modificar)
    bt_modificar.pack()
    bt_modificar.place(x=280,y=350)
```

Ventana Modificar Producto (Interfaz Grafica)



The image shows a graphical user interface window titled "INVENTARIO DE PRODUCTOS". The window has a purple header bar with standard window control buttons (minimize, maximize, close) on the right. The main content area is light gray and contains the following elements:

- A label "MODIFICAR PRODUCTOS:" in a white box.
- A label "INGRESE CODIGO DEL PRODUCTO" followed by a text input field.
- A label "INGRESE EL NUEVO PRECIO PARA EL PRODUCTO" followed by a text input field.
- Two buttons at the bottom: "MENU" (with red text) and "MODIFICAR PRODUCTO" (with blue text).

CONCLUSION

En el desarrollo de la aplicación podemos decir que tenemos un software muy competitivo en diferentes áreas en la comercialización de productos ya que logramos desarrollar una interfaz grafica muy amigable así el usuario no necesita de tener mucho conocimiento para poder manipular todas la funciones que se le ofrecen para llevar un control de los productos que su empresa vende.

Pudimos enriquecer nuestro conocimiento con algo que en principio era desconocido en su totalidad pero logramos construir una aplicación estable con buena presentación y muy versátil.

También podemos decir que superamos nuestros objetivos de manera eficaz lo cual estamos satisfechos en la elaboración de dicho software.

DIAGRAMA DE ACTIVIDADES

- ✓ Perfil de proyecto Domingo 11 de febrero de 2018

Se presenta la idea principal de la aplicación las características que debe cumplir y el funcionamiento esperado

- ✓ Primer avance de proyecto Domingo 4 de marzo de 2018

Se justifica el proyecto con el método para abordar el problema o el desarrollo de la aplicación con las tecnologías que se necesitan la el mismo.

- ✓ Segundo avance Miércoles 18 de abril de 2018

Se plantean el proyecto con el 25% de funcionamiento con lo que se tiene algunas partes del programa funcionando sin entorno grafico

- ✓ Tercer avance Domingo 20 de mayo de 2018

Presentamos nuestra aplicación con más del 75% de avance lo cual funciona con algunos detalles de transiciones de ventanas , fuentes y colores

- ✓ Entrega final Viernes 8 de junio de 2018

Presentamos nuestro proyecto terminado y documentado todo los procesos que conllevo la finalización de proyecto

BIBLIOGRAFÍA

<https://docs.python.org/3/library/tk.html>

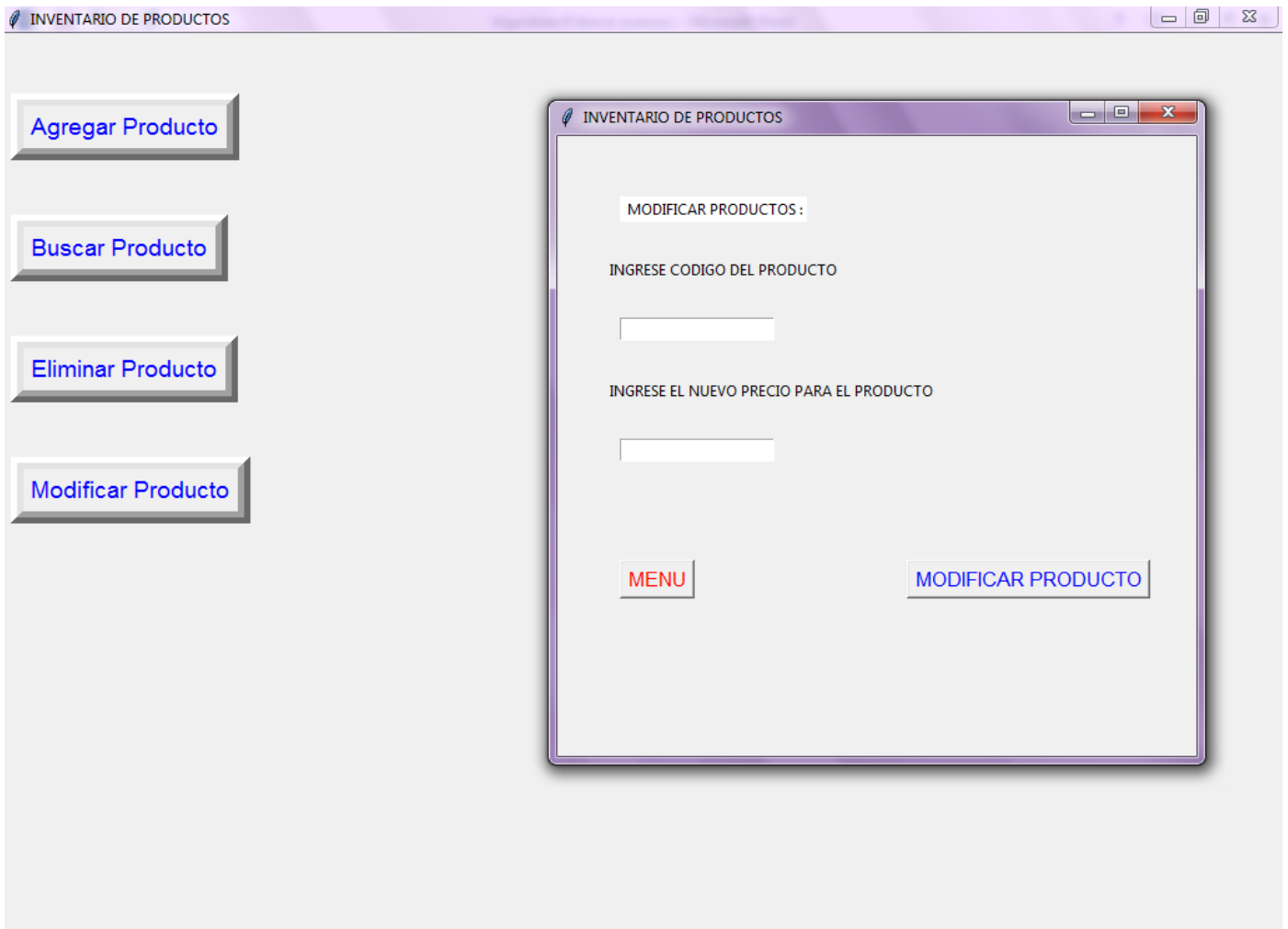
<https://docs.python.org/2/library/os.html>

<https://www.lawebdelprogramador.com/cursos/Python/7494-Manual-de-Tkinter-para-Python.html>

<http://comunicacionmaquina.blogspot.com/2015/07/como-crear-una-base-de-datos-sqlite3.html>

https://python-para-impacientes.blogspot.com/p/blog-page_27.html

ANEXOS



```
def modificar():  
    db = sqlite3.connect("articulos.s3db")  
    c = db.cursor()  
  
    id_producto = entry_id.get()  
    nuevo_precio = entry_valor.get()  
  
    c.execute("update articulos set precio = ('"+nuevo_precio+"' ) where codigo = ('"+id_producto+"")"  
    db.commit()  
    c.close()  
    messagebox.showwarning("MODIFICACION", "ARTICULO MODIFICADO" )
```