



Universidad Fermín
Toro Vicerrectorado
Académico Facultad de
Ingeniería Escuela de
Computación

Manual de Sistema

Autores: Juan Galicia V.30.587.650

Angelo D'Addona V.29.864.529

Indice

1. Introducción

- **Propósito del Manual:** Este manual proporciona una guía completa para usuarios y desarrolladores sobre el sistema de inventario desarrollado en Python con Tkinter, SQLite3 y Pillow.

- **Audiencia:** Este manual está dirigido a usuarios finales del sistema de inventario, así como a desarrolladores y administradores responsables del mantenimiento y la configuración del sistema.

2. Visión General del Sistema

- **Descripción General:** El sistema de inventario es una aplicación desarrollada para gestionar el inventario de productos de una organización. Permite agregar, modificar y eliminar productos, así como realizar seguimiento de existencias y generar informes.

- **Funcionalidades Principales:**

- **Gestión de productos:** Agregar, modificar y eliminar productos.

- **Gestión de existencias:** Actualización automática de existencias según las transacciones.

- **Generación de informes:** Reportes de inventario, ventas, etc.

- **Procesamiento de imágenes:** Carga y visualización de imágenes de productos.

3. Requisitos del Sistema

- **Requisitos de Hardware:**

- Computadora con al menos 2 GB de RAM y procesador dual-core.

- Espacio de almacenamiento suficiente para la base de datos y las imágenes de productos.

- **Requisitos de Software:**

- Python 3.x

- Tkinter

- SQLite3

- Pillow (para procesamiento de imágenes)

- **Instalación del Sistema:**

- Instrucciones detalladas para instalar Python, Tkinter, SQLite3 y Pillow en diferentes sistemas operativos (Windows, macOS, Linux).

4. Manual Técnico del Sistema

4.1 Entorno de Desarrollo

- **Configuración del Entorno:**

- Instalación de Python y configuración del entorno de desarrollo (IDE recomendado: por ejemplo, Visual Studio Code, PyCharm).
- Instalación de bibliotecas adicionales (Tkinter, SQLite3, Pillow).

4.2 Estructura del Proyecto

- **Descripción de la Estructura de Carpetas y Archivos:**

- Explicación detallada de la organización del proyecto (estructura MVC o similar).
- Descripción de archivos principales (main.py, modelos.py, vistas.py, etc.).

4.3 Base de Datos (SQLite3)

- **Diseño de la Base de Datos:**

- Esquema de la base de datos (tablas, relaciones, índices).
- Descripción detallada de cada tabla y sus campos (productos, transacciones, usuarios, etc.).

- **Operaciones de Base de Datos:**

- Ejemplos de consultas SQL para operaciones comunes (selección, inserción, actualización, eliminación).

4.4 Interfaz de Usuario (Tkinter)

- **Descripción de la Interfaz de Usuario:**

- Pantallas principales del sistema (inicio, gestión de productos, informes, configuración).
- Widgets utilizados (botones, entradas de texto, listas, imágenes).

4.5 Procesamiento de Imágenes (Pillow)

- **Funciones de Procesamiento de Imágenes:**

- Ejemplos de código para cargar, mostrar y procesar imágenes de productos.
- Funcionalidades adicionales (redimensionamiento, recorte, rotación).

4.6 Funcionalidades del Sistema

- **Descripción de Funcionalidades:**

- Instrucciones paso a paso para realizar acciones comunes (agregar producto, modificar existencias, generar informes).
- Flujo de trabajo típico del usuario (desde el inicio de sesión hasta la generación de informes).

4.7 Consideraciones de Seguridad

- **Medidas de Seguridad Implementadas:**

- Control de acceso (autenticación y autorización de usuarios).
- Seguridad de datos (encriptación de contraseñas, acceso a la base de datos).

5. Mantenimiento y Soporte

- **Mantenimiento del Sistema:**

- Procedimientos para actualizar el sistema y mantener la base de datos.

- **Soporte Técnico:**

- Contacto de soporte técnico y recursos adicionales (documentación online, comunidad de usuarios).

1. Introducción

Propósito del Manual

Este manual tiene como objetivo proporcionar una guía completa y detallada sobre el sistema de inventario desarrollado en Python utilizando las bibliotecas Tkinter, SQLite3 y Pillow. Su finalidad es facilitar tanto a usuarios finales como a desarrolladores y administradores la comprensión, el uso y el mantenimiento efectivo del sistema.

Audiencia

Este manual está dirigido a dos grupos principales:

1. **Usuarios Finales:** Aquellas personas que utilizarán el sistema de inventario en su día a día para gestionar productos, existencias, realizar transacciones y generar informes. El manual les proporcionará instrucciones claras sobre cómo utilizar cada funcionalidad del sistema de manera eficiente.
2. **Desarrolladores y Administradores:** Aquellos responsables del desarrollo, mantenimiento, actualización y configuración del sistema. El manual les ofrecerá información técnica sobre la arquitectura del sistema, las bibliotecas utilizadas y las mejores prácticas para garantizar su correcto funcionamiento.

2. Visión General del sistema

Descripción General

El sistema de inventario es una herramienta esencial para cualquier organización que maneje productos físicos. Esta aplicación integral ha sido diseñada para simplificar y optimizar la gestión de inventario, proporcionando funcionalidades clave para el control de existencias, seguimiento de productos y generación de informes.

Funcionalidades Principales

1. Gestión de Productos:

- **Agregar Productos:** Permite introducir nuevos productos al sistema, incluyendo detalles como nombre, descripción, código de barras, precio de compra, precio de venta, categoría, proveedor y cantidad en stock.
- **Modificar Productos:** Facilita la actualización de la información de productos existentes, como cambios en precios, descripciones o proveedores.
- **Eliminar Productos:** Permite eliminar productos del sistema cuando ya no son relevantes o están descontinuados.

2. Gestión de Existencias:

- **Actualización Automática:** El sistema ajusta automáticamente las cantidades en stock a medida que se realizan transacciones como ventas, compras o devoluciones.
- **Alertas de Stock Bajo:** Notifica al usuario cuando las existencias de un producto alcanzan un nivel crítico predefinido.
- **Historial de Movimientos:** Mantiene un registro detallado de todas las transacciones que afectan las existencias de cada producto.

3. Generación de Informes:

- **Informe de Inventario:** Proporciona una visión general del estado actual del inventario, incluyendo cantidades en stock, valor total y productos con stock bajo.
- **Informe de Ventas:** Muestra un resumen de las ventas realizadas en un período determinado, incluyendo productos más vendidos, ingresos totales y ganancias.
- **Informes Personalizados:** Permite generar informes específicos según los criterios del usuario.

4. Procesamiento de Imágenes:

- **Carga de Imágenes:** Permite asociar imágenes a cada producto para facilitar su identificación visual.
- **Visualización de Imágenes:** Muestra las imágenes de los productos en la interfaz del sistema.

Beneficios Clave

- **Control Total del Inventario:** Mantén un registro preciso y actualizado de tus productos y existencias.
- **Optimización de Decisiones:** Toma decisiones informadas sobre compras, ventas y promociones basadas en datos reales.
- **Ahorro de Tiempo y Recursos:** Automatiza tareas repetitivas y reduce errores humanos.
- **Mejora de la Experiencia del Cliente:** Asegúrate de tener los productos adecuados en stock para satisfacer la demanda.

3. Requisitos del Sistema

Requisitos de Hardware

Requisitos Mínimos (Inventario Pequeño)

- **Procesador:** Intel Core i3 o equivalente (AMD Ryzen 3)
- **Memoria RAM:** 4 GB
- **Almacenamiento:** Disco duro de 500 GB (preferiblemente SSD para mayor velocidad)
- **Monitor:** Resolución mínima de 1366x768
- **Periféricos:** Teclado, ratón, escáner de código de barras (opcional)

Requisitos Recomendados (Inventario Mediano)

- **Procesador:** Intel Core i5 o equivalente (AMD Ryzen 5)
- **Memoria RAM:** 8 GB
- **Almacenamiento:** Disco duro SSD de 256 GB o más
- **Monitor:** Resolución de 1920x1080 (Full HD) o superior
- **Periféricos:** Teclado, ratón, escáner de código de barras, impresora de etiquetas (opcional)

Requisitos Avanzados (Inventario Grande o Complejo)

- **Procesador:** Intel Core i7 o equivalente (AMD Ryzen 7)
- **Memoria RAM:** 16 GB o más
- **Almacenamiento:** Disco duro SSD de 512 GB o más, o sistema de almacenamiento en red (NAS)
- **Monitor:** Dos monitores Full HD o un monitor de alta resolución (2560x1440 o superior)
- **Periféricos:** Teclado, ratón, escáner de código de barras de alta velocidad, impresora de etiquetas, lector RFID (opcional)

Requisitos de Servidor (Opcional)

Si necesitas un sistema de inventario más robusto y escalable, puedes considerar alojarlo en un servidor dedicado. Los requisitos del servidor dependerán del número de usuarios concurrentes y del volumen de transacciones.

Recomendaciones Adicionales

- **Sistema Operativo:** Utiliza un sistema operativo actualizado y compatible con el software de inventario que elijas.
- **Conexión a Internet:** Asegúrate de tener una conexión a internet estable y rápida, especialmente si utilizas un sistema de inventario basado en la nube.
- **Respaldo de Datos:** Realiza copias de seguridad periódicas de tus datos para evitar pérdidas en caso de fallos de hardware o errores humanos.

Importante:

Estos son requisitos generales. La configuración óptima dependerá del software de inventario específico que elijas y de tus necesidades particulares. Consulta con el proveedor del software para obtener recomendaciones más precisas.

¡Espero que este manual te ayude a elegir el hardware adecuado para tu sistema de inventario! Si tienes alguna pregunta, no dudes en consultar.

Requisitos de Software de Inventario

1. Python 3.x

- **¿Qué es?** Python es un lenguaje de programación versátil y potente, ideal para una amplia gama de aplicaciones, desde desarrollo web hasta análisis de datos y scripting.
- **Instalación:**
 - **Windows:** Descarga el instalador de Python 3 desde la página oficial de Python (<https://www.python.org/downloads/>) y sigue las instrucciones. Asegúrate de marcar la casilla "Add Python to PATH" durante la instalación.
 - **macOS/Linux:** Python suele venir preinstalado. Puedes verificar la versión con el comando `python3 --version` en la terminal. Si necesitas actualizar, utiliza el gestor de paquetes de tu sistema (por ejemplo, Homebrew en macOS).
- **Verificación:** Abre una terminal o línea de comandos y ejecuta `python3 --version`. Deberías ver la versión de Python que has instalado.

2. Tkinter

- **¿Qué es?** Tkinter es el kit de herramientas de interfaz gráfica de usuario (GUI) estándar de Python. Te permite crear ventanas, botones, campos de texto y otros elementos visuales para tus aplicaciones.
- **Instalación:** Tkinter viene incluido con la mayoría de las instalaciones de Python. No necesitas instalarlo por separado.

- **Ejemplo de uso:**

Python

```
import tkinter as tk

# Crear una ventana principal
ventana = tk.Tk()
ventana.title("Mi Aplicación con Tkinter")

# Crear un botón
boton = tk.Button(ventana, text="¡Hola!", command=lambda: print("¡Botón"))
boton.pack()

# Iniciar el bucle de eventos de la GUI
ventana.mainloop()
```

3. SQLite3

- **¿Qué es?** SQLite3 es un sistema de gestión de bases de datos ligero y embebido. Es excelente para almacenar datos de aplicaciones de forma local y sencilla.
- **Instalación:** Al igual que Tkinter, SQLite3 viene incluido en Python.
- **Ejemplo de uso:**

```
import sqlite3

# Crear o conectar a una base de datos
conexion = sqlite3.connect('mi_base_de_datos.db')
cursor = conexion.cursor()

# Crear una tabla (si no existe)
cursor.execute('''CREATE TABLE IF NOT EXISTS usuarios
                 (id INTEGER PRIMARY KEY AUTOINCREMENT, nombre TEXT, edad INTEGER)''')

# Insertar datos
cursor.execute("INSERT INTO usuarios (nombre, edad) VALUES (?, ?)", ("Juan", 25))
conexion.commit()

# Consultar datos
cursor.execute("SELECT * FROM usuarios")
resultados = cursor.fetchall()
print(resultados)

# Cerrar la conexión
conexion.close()
```

4. Pillow (PIL Fork)

- **¿Qué es?** Pillow es una biblioteca de Python que facilita el trabajo con imágenes. Te permite abrir, manipular (recortar, rotar, cambiar tamaño) y guardar imágenes en varios formatos.
- **Instalación:** Utiliza `pip`, el gestor de paquetes de Python:

Bash

```
pip install pillow
```

Python

```
from PIL import Image

# Abrir una imagen
imagen = Image.open('mi_imagen.jpg')

# Rotar la imagen 90 grados
imagen_rotada = imagen.rotate(90)

# Guardar la imagen rotada
imagen_rotada.save('mi_imagen_rotada.jpg')
```

Instalación del Sistema: Python

Para instalar Python 3.x en un equipo con el sistema operativo **Windows 7, 8, 8.1 o 10** utilizar una cuenta de usuario con privilegios de administrador, o bien, la propia cuenta del administrador local. Por seguridad, se puede agregar -temporalmente- para este proceso de instalación la cuenta del usuario actual al grupo local **Administradores**. Para ello, iniciar la aplicación de **Administración de equipos**, acceder a la herramienta del sistema **Usuarios y grupos locales** y agregar la cuenta actual a dicho grupo.

En los sistemas con arquitectura **64 bit** se puede instalar tanto la versión de Python para 32 bit como para 64 bit. En general, trabajar con la versión para 64 bit mejora el rendimiento y permite que un programa pueda utilizar más de 4 Gb de RAM, memoria límite en sistemas 32 bit. Normalmente, trabajando por debajo de este límite no se observan diferencias de rendimiento importantes entre ambas versiones pero se recomienda instalar la versión de 64 bit. De todas formas, si se van a utilizar módulos de terceros, conviene comprobar si están disponibles para la versión a instalar

Proceso de Instalación

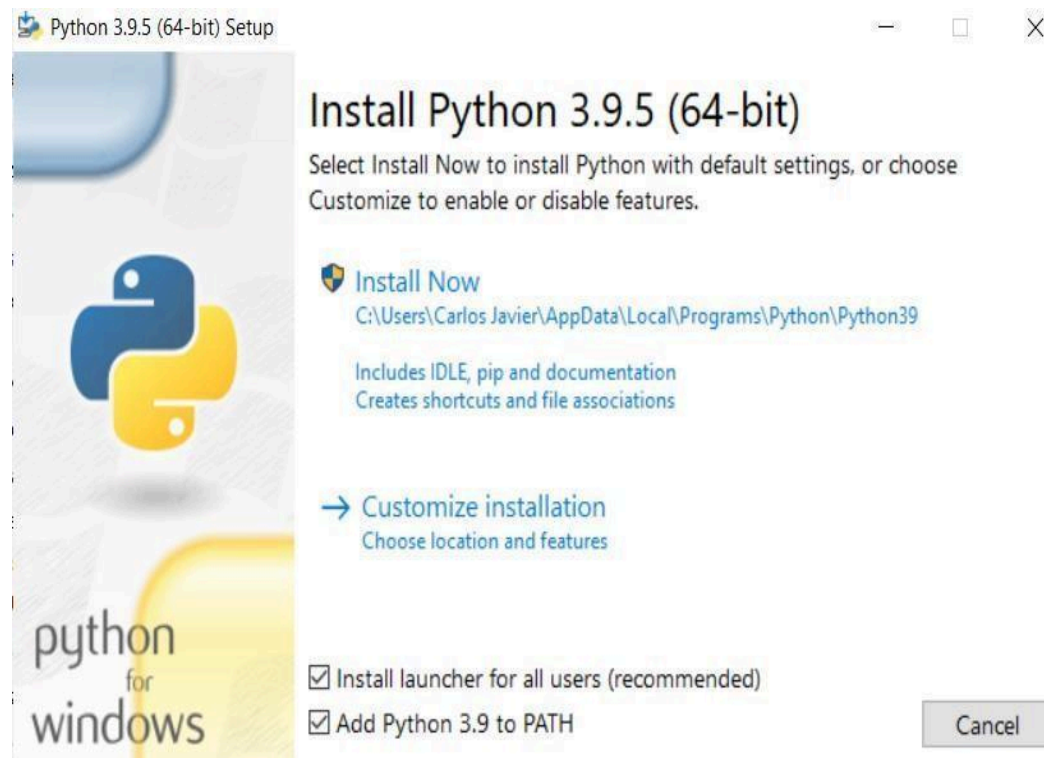
1. Nos dirigimos al siguiente link <https://www.python.org/downloads/> y vamos al área de descargas para Windows y descargar el instalador ejecutable 3.x.

Downloads	Documentation	Community	Success Stories	News
All releases	<h3>Download for Windows</h3> <div>Python 3.9.5</div> <p>Note that Python 3.9+ <i>cannot</i> be used on Windows 7 or earlier.</p> <p>Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads.</p>			
Source code				
Windows				
Mac OS X				
Other Platforms				
License				
Alternative Implementations				

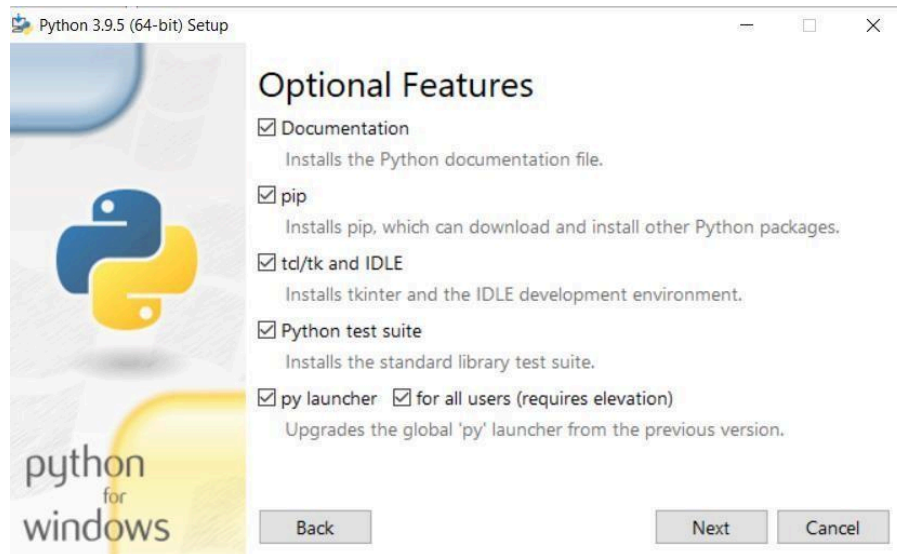
2. Iniciar la instalación ejecutando el archivo descargado **Python-3.x.exe** o **Python-3.x-amd64.exe** con doble clic. Si es necesario, confirmar la ejecución en la ventana de advertencia de seguridad de **Abrir archivo**.



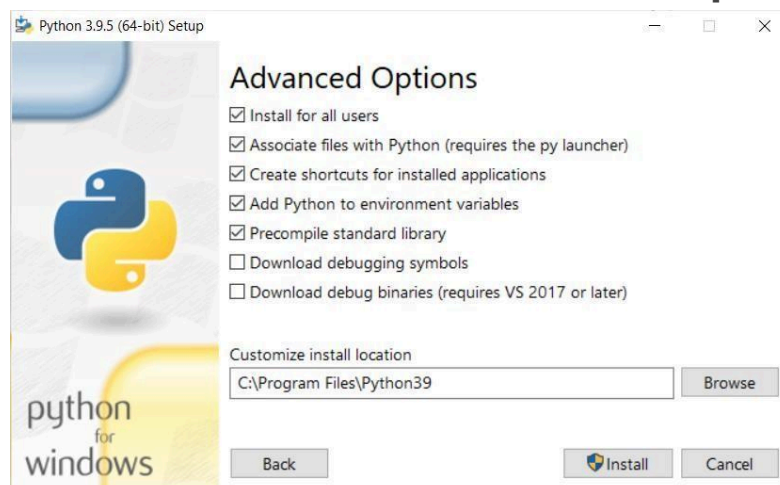
3. Una vez iniciado el instalador, en la ventana **Install Python 3.x (64 bit)** activar las casillas de las opciones: **Install launcher for all users (recommended)** y **Add Python 3.x to PATH**. Después, continuar seleccionando la opción **Customize installation. Choose Location and features**.



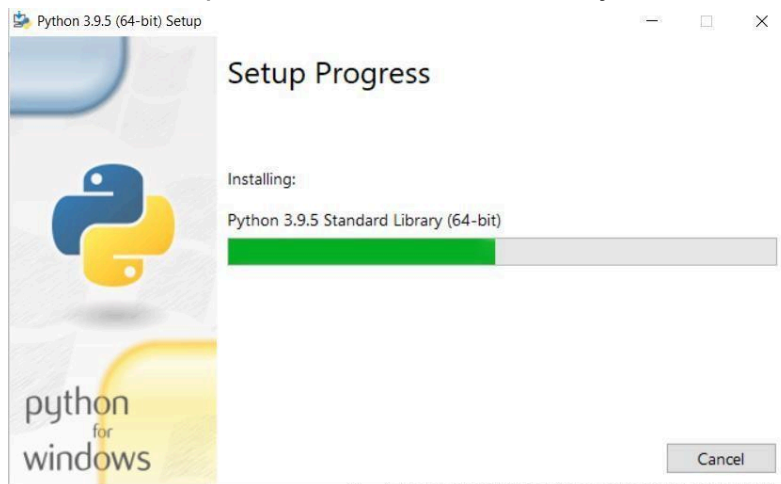
4. En la ventana **Optional features** verificar que están activas todas las opciones y hacer clic en el botón **[Next]**.



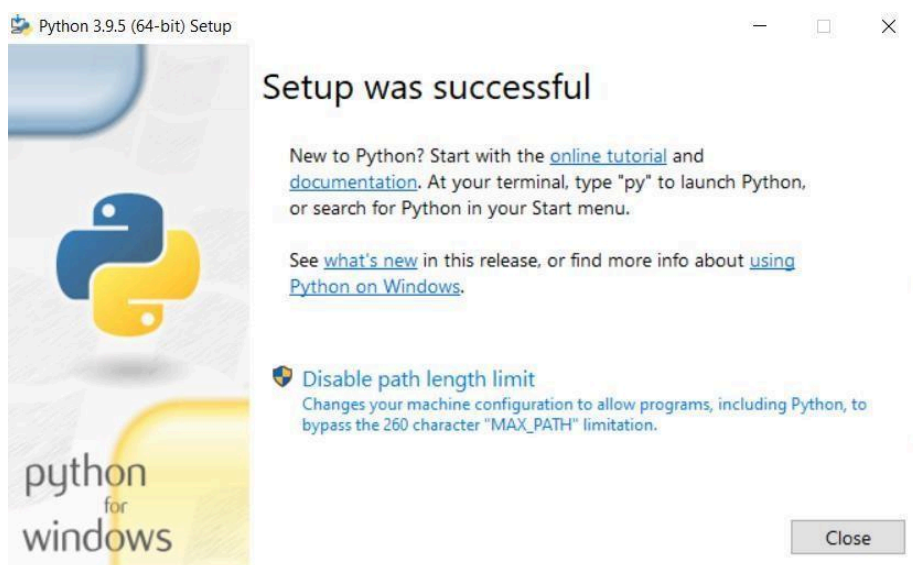
5. En la ventana **Advanced Options** verificar que están activas las opciones de la imagen, escribir la ruta del directorio de instalación "**C:\Python36**" (o aceptar la ruta por defecto) y comenzar la instalación haciendo clic en el botón **[Install]**.



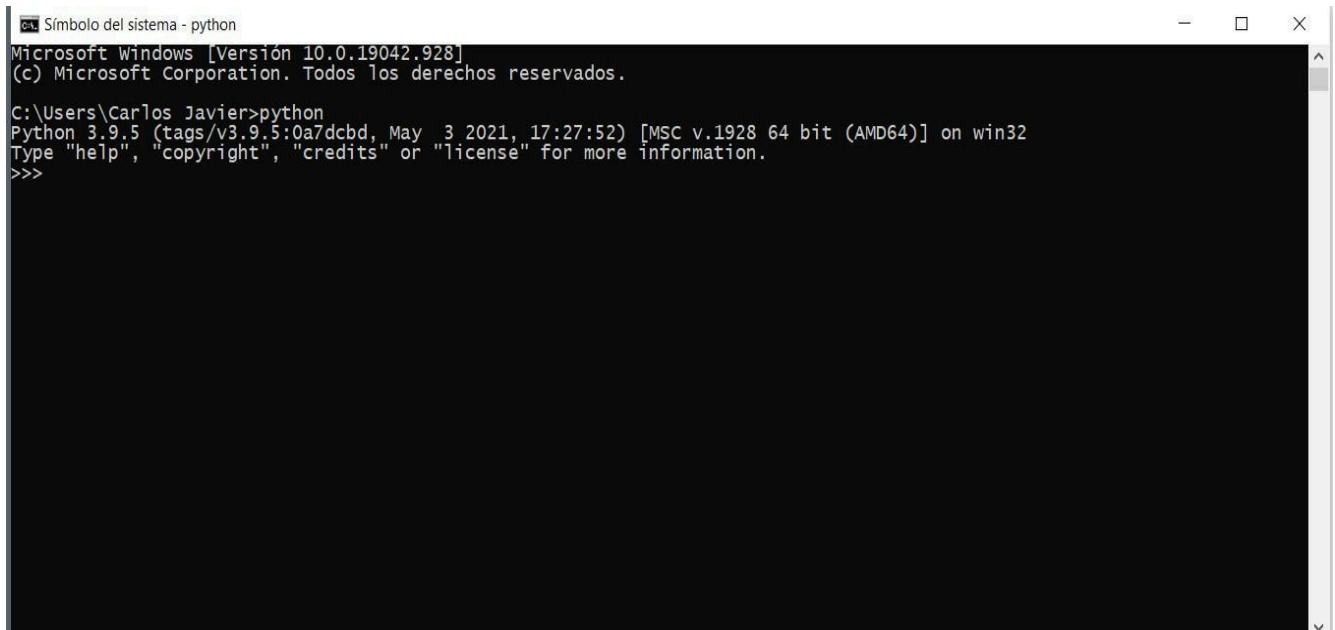
6. A continuación, después de unos segundos de espera, comenzará el proceso de instalación de Python.



7. En la ventana **Setup was successful**, una vez que ha concluido el proceso de instalación hacer clic en el botón **[Close]**. Desde esta ventana es posible acceder a un **tutorial online** de Python, a la **documentación** oficial del lenguaje y a información con las **novedades** de la presente versión.



8. Para probar Python acceder a la línea de comandos ejecutando el programa **cmd.exe** o **Símbolo del sistema**. Después, escribir "**python**" y aceptar para iniciar el **entorno interactivo** de Python donde se puede verificar la versión de Python instalada y es posible introducir comandos Python. Para finalizar, introducir **quit()** o presionar **Ctrl+D**



```
Símbolo del sistema - python
Microsoft Windows [Versión 10.0.19042.928]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Carlos Javier>python
Python 3.9.5 (tags/v3.9.5:0a7dcbbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

4. Entorno de Desarrollo

4.1 Configuración del Entorno

- **Instalación de Python:**
 - Descarga e instala la última versión estable de Python 3.x desde el sitio web oficial: <https://www.python.org/downloads/>
 - Asegúrate de marcar la casilla "Add Python to PATH" durante la instalación para acceder a Python desde la línea de comandos.
- **IDE Recomendado:**
 - Visual Studio Code: Editor de código ligero y extensible con soporte para Python y depuración.
 - PyCharm: IDE completo con herramientas avanzadas para desarrollo en Python.
- **Instalación de Bibliotecas:**
 - Abre una terminal o línea de comandos y ejecuta los siguientes comandos:

Bash

```
pip install tkinter  
pip install sqlite3  
pip install pillow
```

4.2 Estructura del Proyecto

El proyecto sigue una estructura Model-View-Controller (MVC) para separar la lógica de la aplicación, la interfaz de usuario y la gestión de datos:

- **/main.py:** Archivo principal que inicia la aplicación y gestiona el flujo general.
- **/modelos.py:** Contiene las clases que representan los objetos del sistema (Producto, Transacción, Usuario).
- **/vistas.py:** Define las clases que crean y manejan las ventanas y widgets de la interfaz gráfica.
- **/controladores.py:** Implementa la lógica de negocio y la interacción entre modelos y vistas.
- **/base_de_datos.py:** Contiene funciones para interactuar con la base de datos SQLite3.

4.3 Base de Datos (SQLite3)

- **Diseño de la Base de Datos:**

Tabla	Campos
productos	id (INTEGER PRIMARY KEY AUTOINCREMENT), nombre (TEXT), descripcion (TEXT), codigo_barras (TEXT), precio_compra (REAL), precio_venta (REAL), stock (INTEGER), imagen (BLOB)
transacciones	id (INTEGER PRIMARY KEY AUTOINCREMENT), tipo (TEXT), producto_id (INTEGER), cantidad (INTEGER), fecha (DATETIME)
usuarios	id (INTEGER PRIMARY KEY AUTOINCREMENT), nombre_usuario (TEXT), contrasena (TEXT)

- **Operaciones de Base de Datos (Ejemplos):**

SQL

SQL

```
-- Seleccionar todos los productos
SELECT * FROM productos;

-- Insertar un nuevo producto
INSERT INTO productos (nombre, descripcion, precio_compra, precio_venta)

-- Actualizar el stock de un producto
UPDATE productos SET stock = stock - 10 WHERE id = 1;

-- Eliminar un producto
DELETE FROM productos WHERE id = 1;
```

4.4 Interfaz de Usuario (Tkinter)

- **Pantallas Principales:**

- Inicio de Sesión
- Gestión de Productos (agregar, modificar, eliminar, buscar)
- Gestión de Inventario (registrar transacciones, ver historial)
- Generación de Informes
- Configuración del Sistema

- **Widgets Utilizados:**

- Ventanas (`tk.Tk`, `tk.Toplevel`)
- Botones (`tk.Button`)
- Etiquetas (`tk.Label`)
- Entradas de Texto (`tk.Entry`)
- Listas (`tk.Listbox`)

- Imágenes (`tk.Label` con imágenes cargadas con Pillow)
- ...

4.5 Procesamiento de Imágenes (Pillow)

Python

```
from PIL import Image, ImageTk

# Cargar una imagen
imagen = Image.open("ruta/a/la/imagen.jpg")

# Redimensionar la imagen (opcional)
imagen = imagen.resize((100, 100)) # Ejemplo de tamaño

# Convertir a formato Tkinter
imagen_tk = ImageTk.PhotoImage(imagen)

# Mostrar la imagen en un Label
label_imagen = tk.Label(ventana, image=imagen_tk)
label_imagen.pack()
```

4.6 Funcionalidades del Sistema

4.6.1 Gestión de Productos

- **Agregar Producto:**
 1. Acceder a la pantalla de "Gestión de Productos".
 2. Hacer clic en el botón "Agregar Producto".
 3. Completar el formulario con los detalles del producto (nombre, descripción, código de barras, precio de compra, precio de venta, categoría, proveedor, stock inicial).
 4. Cargar una imagen del producto (opcional) utilizando el botón "Seleccionar Imagen".
 5. Hacer clic en el botón "Guardar Producto".
- **Modificar Producto:**
 1. Buscar el producto a modificar en la lista de productos.
 2. Hacer clic en el botón "Modificar" correspondiente al producto.
 3. Actualizar los campos necesarios en el formulario de edición.
 4. Hacer clic en el botón "Guardar Cambios".
- **Eliminar Producto:**
 1. Buscar el producto a eliminar en la lista de productos.
 2. Hacer clic en el botón "Eliminar" correspondiente al producto.
 3. Confirmar la eliminación en el cuadro de diálogo emergente.
- **Buscar Producto:**
 1. Ingresar el nombre o código de barras del producto en el campo de búsqueda.
 2. Presionar la tecla "Enter" o hacer clic en el botón "Buscar".
 3. El sistema mostrará los resultados de la búsqueda en la lista de productos.

4.6.2 Gestión de Inventario

- **Registrar Transacción:**
 1. Acceder a la pantalla de "Gestión de Inventario".

2. Seleccionar el tipo de transacción (compra, venta, ajuste).
 3. Buscar y seleccionar el producto involucrado en la transacción.
 4. Ingresar la cantidad correspondiente.
 5. Hacer clic en el botón "Registrar Transacción".
- **Ver Historial de Transacciones:**
 1. Acceder a la pantalla de "Gestión de Inventario".
 2. Hacer clic en el botón "Ver Historial".
 3. El sistema mostrará una lista de todas las transacciones registradas, con detalles como tipo, producto, cantidad y fecha.

4.6.3 Generación de Informes

- **Generar Informe de Inventario:**
 1. Acceder a la pantalla de "Generación de Informes".
 2. Seleccionar el tipo de informe "Inventario".
 3. Elegir el formato de salida (PDF, CSV, Excel).
 4. Hacer clic en el botón "Generar Informe".
- **Generar Informe de Ventas:**
 1. Acceder a la pantalla de "Generación de Informes".
 2. Seleccionar el tipo de informe "Ventas".
 3. Especificar el rango de fechas para el informe.
 4. Elegir el formato de salida (PDF, CSV, Excel).
 5. Hacer clic en el botón "Generar Informe".

4.7 Consideraciones de Seguridad

4.7.1 Autenticación y Autorización

- Implementar un sistema de inicio de sesión seguro que requiera nombre de usuario y contraseña.
- Almacenar las contraseñas de los usuarios en forma encriptada (utilizando hash y sal) para protegerlas en caso de acceso no autorizado a la base de datos.
- Definir roles de usuario (por ejemplo, administrador, usuario regular) y asignar permisos específicos a cada rol.
- Utilizar un mecanismo de autorización para controlar el acceso a las diferentes funcionalidades del sistema según el rol del usuario.

4.7.2 Seguridad de Datos

- Restringir el acceso directo a la base de datos SQLite3 a través de la interfaz de usuario. Todas las operaciones de la base de datos deben realizarse a través de las funciones definidas en el archivo `base_de_datos.py`.
- Implementar un sistema de copias de seguridad periódicas de la base de datos para proteger contra pérdida de datos en caso de fallos de hardware o errores.
- Considerar el uso de cifrado para proteger datos sensibles almacenados en la base de datos, como información personal de los usuarios.

4.7.3 Seguridad de la Aplicación

- Validar y sanitizar todas las entradas del usuario para prevenir ataques de inyección SQL y otros tipos de ataques.
- Mantener el sistema y las bibliotecas utilizadas actualizadas para corregir vulnerabilidades de seguridad conocidas.
- Realizar pruebas de seguridad periódicas para identificar y corregir posibles debilidades en el sistema.

5. Mantenimiento y Soporte

Esta sección detalla los procedimientos para mantener el sistema actualizado y proporciona información sobre cómo obtener soporte técnico.

5.1 Mantenimiento del Sistema

5.1.1 Actualización del Sistema

- **Control de versiones:** Utilizar un sistema de control de versiones (Git) para gestionar las actualizaciones del sistema. Esto permite realizar un seguimiento de los cambios, revertir a versiones anteriores si es necesario y colaborar de manera eficiente.
- **Pruebas en Entorno de Desarrollo:** Antes de aplicar actualizaciones en el entorno de producción, probarlas exhaustivamente en un entorno de desarrollo idéntico para garantizar que no introduzcan errores o problemas de compatibilidad.
- **Copia de Seguridad:** Realizar una copia de seguridad completa de la base de datos y los archivos del sistema antes de aplicar cualquier actualización. Esto permite restaurar el sistema a un estado anterior en caso de problemas.
- **Actualización de Bibliotecas:** Mantener actualizadas las bibliotecas de Python utilizadas en el proyecto (Tkinter, SQLite3, Pillow). Utilizar el comando `pip` para actualizar las bibliotecas a sus últimas versiones estables.
- **Documentación:** Documentar los cambios realizados en cada actualización, incluyendo la fecha, los cambios específicos y los responsables.

5.1.2 Mantenimiento de la Base de Datos

- **Optimización:** Realizar periódicamente tareas de mantenimiento de la base de datos, como la eliminación de registros innecesarios, la reorganización de índices y la optimización de consultas, para garantizar un rendimiento óptimo.
- **Copias de Seguridad:** Programar copias de seguridad automáticas y regulares de la base de datos. Almacenar las copias de seguridad en una ubicación segura y externa al sistema principal.
- **Verificación de Integridad:** Realizar verificaciones periódicas de la integridad de la base de datos para detectar y corregir posibles errores o inconsistencias.

5.2 Soporte Técnico

- **Contacto de Soporte:** Proporcionar información de contacto clara y accesible para que los usuarios puedan reportar problemas o solicitar asistencia técnica. Esto puede incluir una dirección de correo electrónico, un número de teléfono o un formulario de contacto en línea.
- **Documentación en Línea:** Crear y mantener una documentación en línea completa y actualizada que incluya:
 - **Guía de Usuario:** Instrucciones detalladas sobre cómo utilizar todas las funcionalidades del sistema.

- **Preguntas Frecuentes (FAQ):** Respuestas a las preguntas más comunes de los usuarios.
 - **Guía de Solución de Problemas:** Instrucciones para solucionar problemas comunes y errores.
- **Comunidad de Usuarios:** Fomentar la creación de una comunidad de usuarios en línea (foro, grupo de discusión) donde los usuarios puedan intercambiar conocimientos, compartir consejos y resolver problemas de forma colaborativa.

