

UNIVERSIDAD AMERICANA

Facultad de Ingeniería y Arquitectura



Ingeniería en Sistemas de Información

Introducción a la programación

Control de caja chica (efectivo), para industrias químicas D & D. S.A (INQUIDSA).

Estudiantes:

- Raul Natanael Valverde Ruiz
- Enrique Javier Arana Quintana
- Allison Katherine Fitoria Herrera
 - Leandro Diaz

Docente:

Sylvia Ticay

Managua, Julio de 2025

Introducción.....	3
Definición y Alcance del Estudio.....	4
Actividades de la Práctica de Familiarización.....	6
Descripción del problema	6
Análisis del problema.....	7
Lista de requerimientos.....	8
Diagrama de programa.....	9
Diseño de algoritmo.....	10
PSeInt: para estructurar el pensamiento algorítmico.....	10
Python: para implementar soluciones reales.....	10
¿Por qué usarlos juntos?.....	10
Codificación, ejecución, verificación y depuración.....	13
DOCUMENTACIÓN DE PROYECTO.....	16
Sistema de Facturación.....	16
Introducción.....	16
Inicio de Sesión.....	16
Menú Principal.....	17
Funcionalidades.....	17
1 Ingresar productos.....	17
2 Editar orden.....	17
3 Eliminar orden.....	17
4 Imprimir factura.....	18
5 Cierre de emergencia.....	18
6 Salir.....	18
Recomendaciones.....	18
CONCLUSIÓN.....	19
Mejoras sugeridas para el futuro.....	20
Anexos.....	21
Referencias - Bibliografía.....	23

Introducción

En el contexto de la asignatura Introducción a la Programación, durante el primer semestre del primer año, se propuso el desarrollo de un proyecto práctico que permitiera a los estudiantes aplicar los conceptos fundamentales de la programación en un caso real. El lenguaje seleccionado para este propósito fue Python, por su sintaxis clara y su amplia utilización en el ámbito educativo y profesional.

El objetivo principal de este trabajo fue diseñar y programar un sistema de control de caja, capaz de gestionar la atención de clientes para Industrias Químicas D&D.S.A (INQUIDSA) mediante una aplicación de consola, ya que el control que se lleva en la actualidad es obsoleto y lleva al uso de cuaderno y eso implica más tiempo ya sea al registrar (escribir) o buscar alguna venta. El sistema permite registrar productos, editar y eliminar órdenes, imprimir facturas y realizar cierres de caja, facilitando así el manejo de información financiera básica en un entorno laboral.

A través de este proyecto, se buscó que los estudiantes implementaran estructuras esenciales como variables, condicionales, bucles, funciones y manejo de listas. Además, se fomentó el desarrollo de la lógica computacional y el pensamiento algorítmico, demostrando cómo la programación puede ser una herramienta eficaz para resolver problemas, en este caso es llevar el control de una caja de ventas en el local, esto dando una ventaja de llevar el control del efectivo que entra y las ventas al igual q generar facturas y asi poder imprimirlas sin necesidad de escribirlas.

Definición y Alcance del Estudio

El estudio se centra en la creación de un sistema informático sencillo que reemplace el uso tradicional de cuadernos para el control de caja en pequeños comercios o entornos simulados. El alcance del proyecto abarca el desarrollo de un programa capaz de gestionar usuarios, registrar ingresos de dinero, llevar un seguimiento del balance y facilitar el manejo de información financiera básica. El sistema está diseñado para ser utilizado por personas con conocimientos básicos de computación, priorizando la facilidad de uso y la eficiencia en la gestión de la caja.

El alcance del caso de estudio determina los límites y el nivel de análisis que se realizará, incluyendo:

- Funcionalidades principales:
 - Carga de usuarios y autenticación.
 - Registro de productos en una orden en proceso.
 - Edición y eliminación de productos en la orden.
 - Cálculo del total de la compra.
 - Emisión e impresión de facturas.
 - Registro de dinero facturado.
 - Funciones de emergencia como cierre de caja urgente.
 - Control de flujo mediante un menú interactivo para facilitar la operación.

- Procesos analizados:
 - La interacción con el usuario a través del menú.
 - La gestión de datos en listas y funciones específicas.
 - El flujo de órdenes desde su ingreso hasta la emisión del comprobante.
 - La lógica de cierre y reinicio del proceso para nuevos clientes.

- Límites del análisis:
 - No incluirá detalles internos de las funciones importadas (como `cargar_usuario`, `imprimir_factura`, etc.), sino su integración y uso en el programa principal.
 - No evaluará aspectos de interfaz gráfica o interacción avanzada, dado que el programa funciona en modo consola.
 - No considerará mejoras en la seguridad o persistencia avanzada de datos más allá de la lógica básica implementada.

En resumen, el análisis se enfocará en comprender cómo el código cumple con los requisitos de un sistema básico de gestión de ventas en un entorno comercial, y qué aspectos podrían mejorarse para incrementar su eficiencia, robustez y facilidad de uso.

Actividades de la Práctica de Familiarización

1. Descripción del problema, necesidad o caso de estudio

Tradicionalmente, muchas personas llevan el control de la caja de sus negocios utilizando un cuaderno, lo que puede generar errores, pérdida de información y dificultades para realizar cálculos o consultas rápidas. La necesidad identificada es modernizar este proceso, permitiendo que el registro y control de la caja se realice de manera digital, segura y eficiente, eliminando la dependencia del papel y reduciendo la posibilidad de errores humanos.

En este proceso llevamos a cabo una observación del ambiente que se lleva en INQUIDSA al momento de realizar una venta en ventanilla, que el orden es llevado en el cuaderno como se menciona anteriormente, a lo cual decidimos el ahorro del papel y el uso de la tecnología para una eficiencia y mejora en el orden y automatización de el control de estas ventas.

2. Análisis del problema

- Entradas requeridas:

- - Usuario y contraseña para el acceso al sistema.
- - Códigos y cantidades de productos a ingresar en la orden.
- - Opciones seleccionadas en el menú principal (ingresar productos, editar/eliminar orden, imprimir factura, cierre de emergencia, salir).

- Salidas deseadas:

- - Factura generada con el detalle de la compra y el total a pagar.
- - Registro actualizado de facturas y del dinero acumulado en la caja.
- - Mensajes informativos y de confirmación para el usuario.

- Método que produce las salidas:

El sistema utiliza funciones para procesar las entradas del usuario, gestionar la información de productos y órdenes, calcular totales y generar archivos de registro y facturación. El flujo principal se controla mediante un menú interactivo que guía al usuario en cada paso.

Lista de Requerimientos

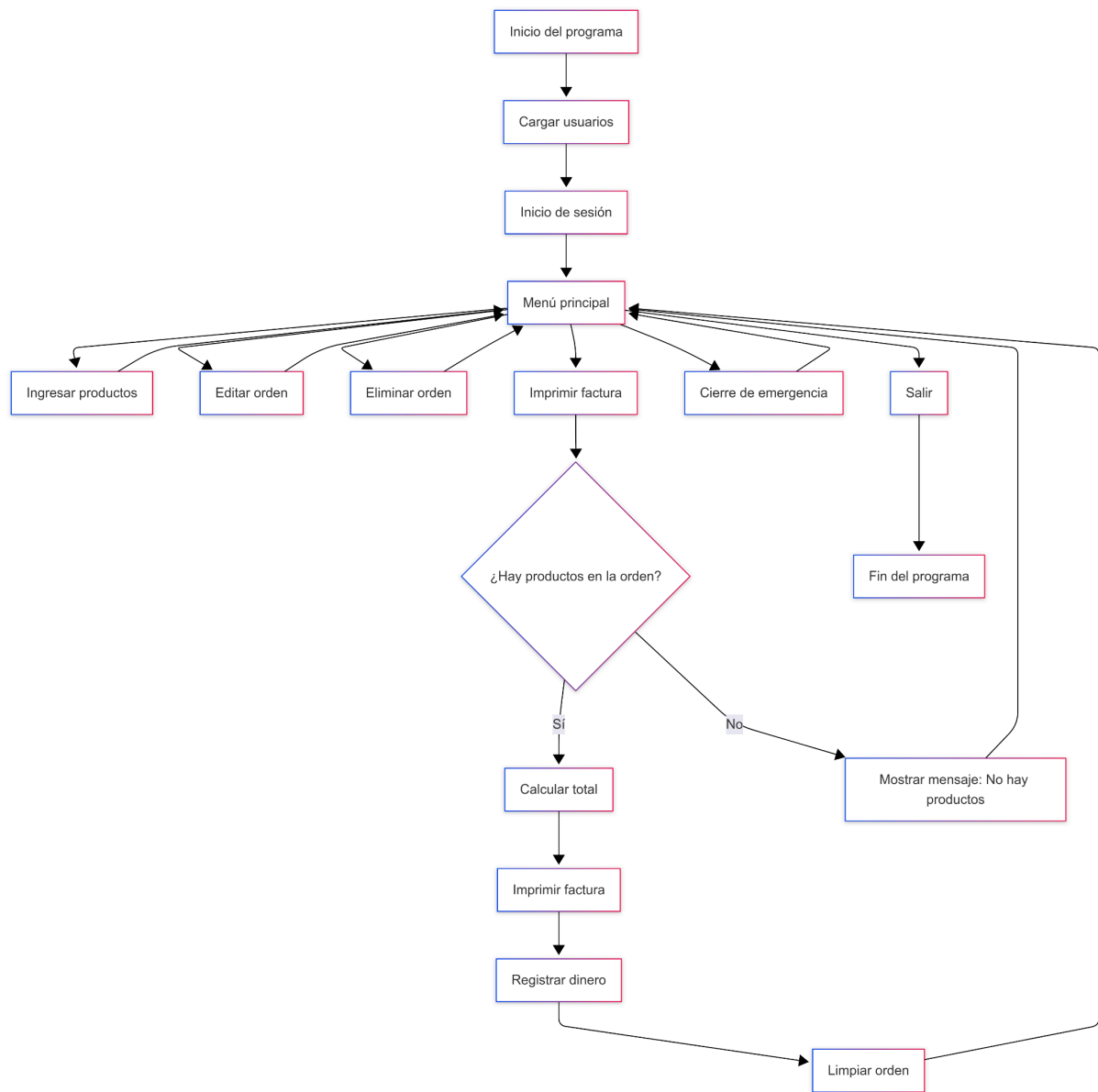
Requerimientos Funcionales

- El sistema debe permitir el inicio de sesión de usuarios mediante usuario y contraseña.
- El sistema debe mostrar un menú principal con opciones para ingresar productos, editar orden, eliminar orden, imprimir factura, cierre de emergencia y salir.
- El sistema debe permitir ingresar productos a la orden mediante códigos y cantidades.
- El sistema debe permitir eliminar toda la orden actual.
- El sistema debe calcular y mostrar el total de la orden.
- El sistema debe generar e imprimir una factura con el detalle de la compra y el total.
- El sistema debe registrar el total de cada factura en un archivo de registro.
- El sistema debe acumular y registrar el dinero total en caja.
- El sistema debe mostrar mensajes claros al usuario en cada operación.

Requerimientos No Funcionales

- El sistema debe ser fácil de usar y comprender para personas con conocimientos básicos de informática.
- El sistema debe proteger la información de acceso mediante la ocultación de la contraseña al escribirla.
- El sistema debe ejecutarse en el entorno de consola de Windows y ser compatible con Python 3.
- El sistema debe permitir la modificación y mantenimiento sencillo del código, utilizando funciones y módulos separados.
- El sistema debe responder rápidamente a las acciones del usuario, sin demoras perceptibles.
- El sistema debe garantizar la integridad de los datos en caso de cierre inesperado.

Diagrama de programa



3. Diseño del algoritmo por los sub-problemas

Introducción al problema

A este problema le daremos una solución que llegue a poder ser actualizable y así mismo en este proceso utilizamos herramientas como pseint, para entender la lógica y lo básico de la creación a la solución de nuestro sistema y de python en VSC (Visual Studio Code) para la redacción de el código del programa principal con sus funciones. Es fundamental contar con herramientas que faciliten tanto el aprendizaje de la lógica como la implementación práctica de soluciones. En este sentido, PSeInt y Python destacan como dos recursos valiosos y complementarios.

PSeInt: para estructurar el pensamiento algorítmico

PSeInt es un software diseñado para ayudar a aprender los fundamentos de la programación mediante la creación de algoritmos en pseudocódigo. Algunas de sus ventajas son:

- **Permite escribir algoritmos usando un lenguaje cercano al español, lo que facilita comprender la lógica sin preocuparse por la sintaxis estricta de un lenguaje real.**
- **Ayuda a estructurar el pensamiento algorítmico, dividiendo los problemas en pasos secuenciales claros.**
- **Tiene herramientas para depurar el flujo del algoritmo, como la ejecución paso a paso, lo que permite identificar errores lógicos de forma sencilla.**

Por eso, PSeInt es ideal para planificar y analizar la lógica antes de escribir un programa definitivo.

Python: para implementar soluciones reales

Por otro lado, Python es un lenguaje de programación de alto nivel, ampliamente usado en la industria y en la academia, debido a su:

- **Sintaxis simple y legible, muy cercana al lenguaje natural.**
- **Gran cantidad de bibliotecas para matemáticas, ciencia de datos, desarrollo web, inteligencia artificial, entre otros.**
- **Comunidad extensa, que genera recursos, documentación y soluciones a problemas comunes.**

Con Python, es posible tomar los algoritmos diseñados en PSeInt y transformarlos en programas funcionales, listos para ejecutarse y resolver problemas reales.

¿Por qué usarlos juntos?

- **Usar PSeInt permite primero concentrarse en la lógica, sin errores de sintaxis, asegurando que el algoritmo funcione correctamente a nivel conceptual.**
- **Posteriormente, Python se emplea para escribir el programa final, añadiendo detalles propios del lenguaje como tipos de datos, manejo de archivos, interfaces gráficas o conexión a bases de datos.**

De este modo, PSeInt y Python se complementan perfectamente: uno ayuda a pensar y diseñar, y el otro a materializar y ejecutar.

y ya volviendo al problema del caso y haber introducido brevemente que son y las ventajas de nuestras herramientas.

El problema general se descompone en subproblemas, cada uno abordado mediante un subprograma o función:

- - Autenticación de usuarios.
- - Ingreso de productos a la orden.
- - Edición y eliminación de productos en la orden.
- - Cálculo del total de la orden.
- - Generación e impresión de la factura.
- - Registro del dinero acumulado.
- - Cierre de caja por emergencia.

Cada sub-problema fue modelado como una función o módulo independiente, facilitando la organización y el mantenimiento del código.

```
// Inicio del bucle principal del menú
```

```
Mientras salir = Falso Hacer //se agregara antes un inicio de sesion con usuarios restringidos para el personal autorizado  
    Escribir ""  
    Escribir "MENÚ PRINCIPAL"  
    Escribir "1. Ingresar productos"  
    Escribir "2. Eliminar orden" //Se agregaran más opciones en el python por razones de escritura de codigo  
    Escribir "3. Imprimir factura"  
    Escribir "4. Salir"  
    Escribir "Seleccione una opción:"  
    Leer opcion
```

```

// Evaluación de la opción elegida
Segun opcion Hacer
1: // Ingresar productos
  Escribir "¿Cuántos productos desea ingresar?"
  Leer n
  Para j ← 1 Hasta n Hacer
    Escribir "Nombre del producto ", cantidadProductos + 1, ":
    Leer producto
    Escribir "Precio del producto:"
    Leer precio
    // Guardar el producto y su precio en los arreglos
    productos[cantidadProductos] ← producto
    precios[cantidadProductos] ← precio
    // Sumar al total
    total ← total + precio
    cantidadProductos ← cantidadProductos + 1
  FinPara

2: // Eliminar orden
  Escribir "Eliminando orden..."
  cantidadProductos ← 0
  total ← 0
  Escribir "Orden eliminada."

3: // Imprimir factura
  Escribir "-----"
  Escribir "FACTURA"
  Si cantidadProductos = 0 Entonces
    Escribir "No hay productos en la orden."

```

4. Codificación, ejecución, verificación y depuración

El proyecto de programación se desarrolló en el IDE Visual Studio Code, utilizando el lenguaje de programación Python. Se creó un módulo Python para cada subprograma diseñado, agrupando las funciones en archivos como 'funciones.py' y 'main.py'. Durante el desarrollo, se realizaron pruebas de ejecución, verificación de resultados y depuración de errores para asegurar el correcto funcionamiento del sistema y el cumplimiento de los requisitos planteados.

```
1  import pwinput
2  import os
3  import textwrap
4  from datetime import datetime
5
6  # Cargar usuarios del archivo usuarios.txt
7  def cargar_usuario():
8      usuarios = {}
9      if os.path.exists("usuarios.txt"):
10         with open("usuarios.txt", "r") as archivo:
11             for linea in archivo:
12                 usuario, contraseña = linea.strip().split(",")
13                 usuarios[usuario] = contraseña
14         return usuarios
15
16 # Verifica usuario y contraseña con 3 intentos
17 def inicio(usuarios):
18     print(textwrap.dedent("""
19     SISTEMA DE CAJAS INQUIDSA
20     """))
21
22     intentos = 0
23     while intentos < 3:
24         usuario = input("Usuario: ")
25         contraseña = pwinput.pwinput("Contraseña: ")
26
27         if usuario in usuarios and usuarios[usuario] == contraseña:
28             print(f"\nIniciando sesión...\nBienvenid@ {usuario}")
29             return usuario
30         else:
31             print("\nUsuario o contraseña incorrectos.")
32             intentos += 1
33
34     print("\nLímite de intentos sobrepasado.")
35     exit()
36
```

para iniciar la digitación del programa empezamos codificando las funciones que se usarán en módulos, el archivo donde se guardaran lleva por nombre, "[funciones.py](#)" en el que la primera imagen muestra la función de cargar los usuarios desde "usuarios.txt" y luego muestra en el programa la siguiente función de inicio de sesión.

```

≡ usuarios.txt
1   raul,1334
2   francisco,1234
3   doralinda,2709

```

(imagen de “usuarios.txt”)

Desde otra función la cual tiene el trabajo de ingresar los productos seleccionados desde el archivo “categorías.txt” es para la escritura y el ingreso de los productos a su factura.

```

funciones.py > eliminar_orden
36
37 # Permite ingresar productos usando códigos desde el archivo categorias.txt
38 def ingreso_productos():
39     productos = {}
40     categoria_actual = ""
41
42     with open("categorias.txt", "r", encoding = "utf-8") as archivo:
43         for linea in archivo:
44             linea = linea.strip()
45             if linea.startswith("[") and linea.endswith("]"):
46                 categoria_actual = linea[1:-1]
47             elif linea:
48                 codigo, nombre, precio = linea.split(",")
49                 productos[codigo] = {
50                     "nombre": nombre,
51                     "precio": float(precio),
52                     "categoria": categoria_actual
53                 }
54
55     print("\nProductos disponibles por código:")
56     for cod, datos in productos.items():
57         print(f"{cod} - {datos['nombre']} (${datos['precio']:.2f}) [{datos['categoria']}]")
58
59     orden = []
60
61     while True:
62         codigo = input("\nIngresa el código del producto (o 'fin' para terminar): ").strip()
63         if codigo.lower() == "fin":
64             break
65         if codigo in productos:
66             try:
67                 cantidad = int(input("Cantidad: "))
68                 nombre = productos[codigo]["nombre"]
69                 precio = productos[codigo]["precio"]
70                 orden.append((nombre, precio, cantidad))
71                 print(f"Añadido: {nombre} x{cantidad}")
72             except ValueError:
73                 print("Cantidad inválida.")
74         else:
75             print("Código no encontrado.")
76     return orden
77

```

El archivo de las categorías esta dividido en 5 categorías básicas divididas según el inventario de la empresa.

≡ categorias.txt

```
1  [Cafeteria]
2  101,Leche Entera,1.75
3  102,Leche Deslactosada,1.90
4  103,Agua purificada,5.50
5  104,Cafe presto 400g,3.10
6  105,Cafe Selecto Superior 200g ,3.50
7
8  [Ambientadores Galones]
9  201,Fragancia Limon,2
10 202,Fragancia Bebe,2
11 203,Fragancia Jardín,2
12 204,Fragancia Canela,3
13 205,Fragancia Cereza,2.50
14
15 [Aereosol]
16 301,Lyson 500ml ,3
17 302,Glade 450ml ,2.50
18 303,Pledge limpia muebles 378ml,5.10
19 304,Limpia Vidrio MR. Musculo 500ml,3.40
20 305,Glade P/Autos clipo 7ml,1.60
21
22 [Pastilla Sanitaria]
23 401,Pastilla Sanitaria Terror 50gr ,0.80
24 402,Pastilla Sanitaria Pato Azul,3.35
25 403,Gel Pato Disco Activo 36ml,5.00
26 404,Pastilla Sanitaria Wesie 50gr, 0.90
27 405,Tapete Anti-Salpicadura UND,5.50.
28
29 [Aseo Personal]
30 501,Jabón Corporal,1.85
31 502,Jabon p/M GAL,3.20
32 503,Jabon p/M l,1.50
33 504,Papel Higiénico 4x1,3.50
34 505,cepillo dental Pk/3,2.40
35
36 [Productos de Limpieza]
37 601,Detergente Líquido,3.50
38 602,Suavizante de Ropa,2.90
39 603,Detergente en polvo,2.30
40 604,Pinesol Gal,6.50$
41 605,Jabon industrial Gal,7.50
```

(“categorias.txt”)

DOCUMENTACIÓN DE PROYECTO

Manual de usuario

¡Perfecto! Aquí tienes un **manual de usuario** bien organizado, basado en tu contenido, con un formato más claro y amigable para entregar o incluir en un documento oficial.

Manual de Usuario

Sistema de Facturación

Introducción

Este sistema está diseñado para facilitar la gestión de facturación en un punto de venta o caja. Permite:

- Gestionar la facturación de productos.
- Editar o eliminar órdenes.
- Imprimir facturas.
- Realizar cierres de caja.

Inicio de Sesión

- Al abrir el programa, se cargan automáticamente los usuarios registrados.
- Debes ingresar tus credenciales (usuario y contraseña) para acceder al sistema.

Menú Principal

Una vez dentro, el sistema mostrará el siguiente menú:

1. **Ingresar productos:** Añade productos a la orden actual.
2. **Editar orden:** Modifica los productos ya ingresados.
3. **Eliminar orden:** Elimina productos específicos de la orden actual.
4. **Imprimir factura:** Calcula el total e imprime la factura.
5. **Cierre de emergencia:** Permite cerrar la caja de forma rápida en caso urgente.
6. **Salir:** Cierra el sistema.

Funcionalidades

1 Ingresar productos

- Selecciona esta opción para añadir productos a la orden.
- Ingresar la información solicitada: nombre del producto, cantidad, precio, etc.

2 Editar orden

- Modifica los datos de productos ya agregados (cantidad, precio o descripción).

3 Eliminar orden

- Elimina uno o varios productos específicos de la orden actual.

4 Imprimir factura

- Calcula automáticamente el total de la orden.
- Imprime la factura con el detalle de los productos y datos del cliente.
- Registra el monto recibido y genera el cambio si aplica.
- Limpia la orden actual para atender al siguiente cliente.

5 Cierre de emergencia

- Realiza un cierre rápido del sistema, útil en casos de corte de luz, fallas técnicas o emergencias.

6 Salir

- Cierra el programa de forma segura y registra el cierre de caja.

Recomendaciones

- Antes de imprimir la factura, **verifica que los productos y cantidades sean correctos.**
- Utiliza el **cierre de emergencia** solo cuando sea estrictamente necesario.
- Mantén actualizados los datos de usuarios y productos para evitar errores en la facturación.

CONCLUSIÓN

A través del desarrollo del presente proyecto, nosotros conseguimos aplicar de manera práctica los conceptos fundamentales de la programación orientados a la solución de un problema real. La creación del sistema de control de caja chica para **Industrias Químicas D & D. S.A (INQUIDSA)** permitió no solo modernizar un proceso tradicionalmente manual, sino también evidenciar cómo las herramientas tecnológicas pueden optimizar el tiempo, reducir errores y mejorar el manejo de la información financiera.

El proyecto abordó desde el análisis del problema, pasando por el diseño del algoritmo en pseudocódigo utilizando **PSeInt**, hasta su implementación final en **Python**, mostrando la importancia de una planificación estructurada y del uso de herramientas que faciliten el aprendizaje progresivo de la lógica computacional. Asimismo, el trabajo en equipo y la distribución de tareas fomentaron habilidades de colaboración y responsabilidad compartida.

Finalmente, este proyecto constituye un ejemplo tangible de cómo los conocimientos adquiridos durante el primer año en la carrera de **Ingeniería en Sistemas de Información** pueden aplicarse directamente para resolver necesidades empresariales concretas, sentando las bases para futuros desarrollos de mayor complejidad y fomentando una mentalidad crítica y analítica en torno al uso de la tecnología.

Mejoras sugeridas para el futuro

1. Persistencia avanzada de datos:

Incorporar una base de datos (SQLite, MySQL) en lugar de archivos `.txt`, lo que permitiría un manejo más seguro, rápido y estructurado de la información.

2. Interfaz gráfica:

Desarrollar una interfaz GUI usando bibliotecas como `Tkinter` o `PyQt` para facilitar la interacción, haciendo el sistema más intuitivo.

3. Gestión de inventario:

Ampliar el sistema para incluir control de stock, alertas de productos bajos y reportes automáticos.

4. Historial de facturas:

Guardar las facturas emitidas con un identificador único, permitiendo su posterior consulta e impresión.

5. Mejoras en seguridad:

- Encriptar contraseñas en el archivo de usuarios.
- Implementar logs de acceso para auditar quién usa el sistema.

6. Soporte multisesión y multiterminal:

Preparar el programa para permitir su uso simultáneo desde varias cajas o computadoras conectadas en red local.

Guía de observación

Nombre del lugar observado: Industrias Químicas D & D.S.A (INQUIDSA).

Dirección: Nicaragua, Managua, Del BAC de Ciudad Jardín ½ cuadra al sur, casa M44.

Fecha de Observación: 25 de junio del 2025

Hora de Observación: 10:00 a.m - 3:00 p.m

Observadores:

- Raúl Natanael Valverde Ruíz
- Enrique Javier Arana Quintana

Objetivo de la observación:

Registrar de forma directa los principales problemas al momento de vender productos en el local, con el fin de sustentar el desarrollo de una solución eficaz que nosotros buscamos.

Aspectos Observados

Aspecto Evaluado	Descripción de lo observado
Registro de factura	El personal anotaba manualmente las órdenes en un cuaderno y se observaron que hubieron errores al momento de escribir y en otros casos eran ilegibles.
Tiempo de búsqueda de producto	La persona encargada de realizar las facturas buscaba en un libro todos los productos que tiene buscando código y precio y tardaba entre 1 min y 3 min a veces.
Tiempo de espera del cliente	A veces los clientes realizaban el pedido y el producto ya no estaba disponible..
Impresión de factura	Al momento de que la persona no solicitaba o no requerir factura membretada no se crean facturas y eso hace que no se lleve un control eficaz de estas ventas.

Herramientas utilizadas

- **Toma de nota de puntos claves**
- **Cronómetro**
- **Visualización de cámaras**

Conclusión

Durante la visita se evidenciaron deficiencias en la gestión de facturas, en el cual los errores humanos, las demoras con el cliente y una comunicación ineficiente, hicieron que viéramos viable como propuesta a una solución un sistema que proporcione eficacia y la optimización del tiempo para el local.

Referencias - Bibliografía

colaboradores de Wikipedia (2025a, marzo 3) PSEint. Wikipedia, la Enciclopedia Libre
<https://es.wikipedia.org/wiki/PSeint>

colaboradores de Wikipedia (2025b, junio 15) Python. Wikipedia, la Enciclopedia Libre
<https://es.wikipedia.org/wiki/Pythno>

colaboradores de Wikipedia (2025c, junio 25) Visual Studio Code. Wikipedia, la Enciclopedia Libre
https://es.wikipedia.org/wiki/Visual_Studio_Code

El tutorial de Python. (s.f.) Python Documentation
<https://docs.python.org/es/3.13/tutrotial/index.html>