UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES Y DE PROGRAMACIÓN
SEGUNDO SEMESTRE 2023

Práctica 1

Objetivos:

- Que el estudiante implemente una solución de software con base en los distintos paradigmas de programación vistos en clase y laboratorio.
- Que el estudiante adquiera habilidades en el manejo de archivos, lógica de programación y manipulación de estructuras de datos en Python.

Enunciado:

Se solicita al estudiante del curso de lenguajes formales y de programación el desarrollo de un programa en Python que permita gestionar un inventario y registrar los movimientos de productos utilizando archivos de texto.

El sistema de gestión de inventario permitirá a los usuarios realizar las siguientes instrucciones:

Menú Principal:

Debe realizarse un menú con las siguientes opciones:

Ejemplo:

```
-------------------------------------------
Practica 1 - Lenguajes formales y de programacion
-------------------------------------------
# Sistema de inventario:

1. Cargar Inventario inicial
2. Cargar Instrucciones de movimientos
3. Crear Informe de inventario
4. Salir

Ingrese una opcion:
```

Cargar Inventario Inicial:

El archivo **.inv** contendrá las instrucciones para configurar el inventario inicial. Cada línea del archivo será de la forma:

```
crear_producto <nombre>;<cantidad>;<precio_unitario>;<ubicacion>
```

Donde **"crear_producto"** es el nombre de la instrucción, **"nombre"** es el nombre del producto, **"cantidad"** es la cantidad inicial disponible en el inventario, **"precio_unitario"** es el precio por unidad del producto y **"ubicación"** es la bodega donde se almacenará el producto. El programa leerá este archivo al inicio para crear el inventario con los productos y cantidades iniciales especificadas. Por ejemplo:

```
crear_producto Manzanas;100;2.50;BodegaA
crear_producto Peras;50;3.00;BodegaB
crear_producto Platanos;75;1.75;BodegaC
```

Cargar Instrucciones de Movimientos:

El archivo **.mov** contendrá las instrucciones de movimientos de productos. **Cada línea del archivo será de alguna de las siguientes formas**:

Agregar stock:

Indica que se agregará una cantidad de unidades del producto con el nombre "*<nombre>*" al inventario, únicamente en la ubicación con el nombre "*<ubicacion>*".

```
agregar_stock <nombre>;<cantidad>;<ubicacion>
```

Ejemplo:

```
agregar_stock Manzanas;50;BodegaA
```

Reglas:

1. Si el producto existe en esa ubicacion, se debe actualizar la cantidad.
2. Si el producto no existe en esa ubicacion, se debe mostrar un **mensaje de error**.

Vender producto:

Indica que se realizará una venta de una cantidad de unidades del producto con el nombre "*<nombre>*" del inventario de la ubicacion "*<ubicacion>*". La cantidad vendida debe ser menor o igual a la cantidad disponible en el inventario.

```
vender_producto <nombre>;<cantidad>;<ubicacion>
```

Ejemplo:

```
vender_producto Peras;20;BodegaB
```

Reglas:

1. Si el producto no existe en esa ubicacion, se debe mostrar un **mensaje de error**.
2. Si el producto existe en esa ubicacion y la cantidad es menor o igual que la existencia, se debe actualizar la cantidad.
3. Si la cantidad a vender es mayor a la cantidad en esa ubicacion, se debe mostrar un **mensaje de error**

Crear Informe de inventario:

Al finalizar las operaciones de carga de archivos, el programa deberá mostrar en un **archivo .txt** una visión general del estado actual del inventario, mostrando la información detallada de todos los productos existentes en el sistema. Este informe contendrá una tabla con las siguientes columnas:

- **Producto**: El nombre del producto registrado en el inventario.
- **Cantidad**: La cantidad actual de unidades disponibles del producto en el inventario.
- **Precio Unitario**: El precio por unidad del producto.
- **Valor Total**: El valor total del inventario para cada producto, calculado como el producto de la cantidad disponible y el precio unitario.
- **Ubicación**: El nombre de la ubicación de ese producto

Ejemplo:

resultado_123123.txt

```
Informe de Inventario:

Producto     Cantidad     Precio Unitario     Valor Total     Ubicación
------------------------------------------------------------------------
Manzanas     120          $2.50               $300.00         BodegaA
Manzanas     23           $1.30               $29.90          BodegaB
```

Peras	30	$3.00	$90	BodegaB
...	...	...	...	...

Entregables

En UEDI entregar únicamente el link del repositorio de GitHub que debe incluir:

- Código fuente de la aplicación
- Manual técnico (debe explicar la lógica de su programa)

Consideraciones importantes:

- **NO** se debe tomar en cuenta el uso de números negativos, ni vendrán ubicaciones en él .mov que no existan anteriormente en él .inv, o palabras mal escritas en los archivos de entrada.
- La práctica se debe desarrollar de forma individual.
- Se debe de crear un repositorio privado en GitHub con el siguiente nombre: LFP_S2_2023_Practica_<carnet>
- Se debe utilizar el lenguaje Python.
- No se aceptan entregas vía correo electrónico u otro medio.
- No se puede agregar o quitar algún símbolo en el archivo de entrada. El proyecto deberá
- funcionar con los archivos de prueba que se disponga para la calificación, sin modificación.
- El estudiante es responsable del horario que elija para calificarse, en caso de no poder presentarse deberá notificar al auxiliar con suficiente anticipación (1 días antes) para ceder su lugar a otro estudiante, en caso contrario el estudiante solo obtendrá el 80% de su nota obtenida.
- No se dará prórroga para la entrega de la práctica.
- **COPIA PARCIAL O TOTAL DE LA PRÁCTICA TENDRÁ UNA NOTA DE 0 PUNTOS, Y SE NOTIFICARÁ A LA ESCUELA DE SISTEMAS PARA QUE SE APLIQUEN LAS SANCIONES CORRESPONDIENTES.**
- **En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará la práctica; por lo cual, se tendrá una nota de cero puntos.**
- Fecha de entrega: 22 de agosto de 2023, antes de las 23:59, no se recibirán entregas después de la fecha y hora establecida.