LFP

PRÁCTICA NO. 1

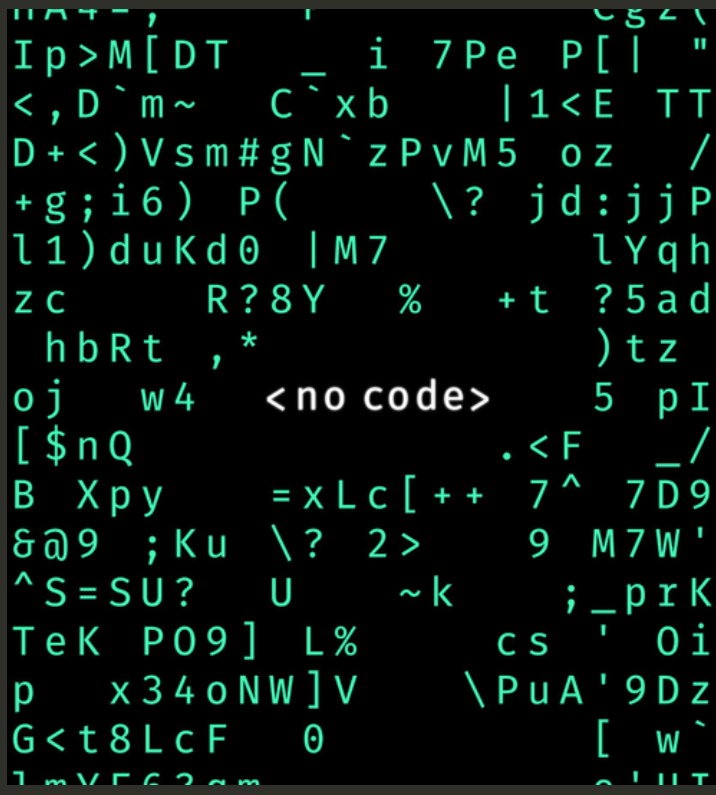
GEOVANNY HERRERA
3556794340101
202110588

ÍNDICE

01 LECTURA

02 MOVIMIENTOS

03 INVENTARIO



LECTURA

PROCESO DE LECTURA DEL ARCHIVO DE ENTRADA

El proceso de lectura del archivo empieza con la primera condicional de la lectura, y es confirmar que el texto "crear_producto" se encuentra en la línea de texto, y así poder seguir con demás verificaciones, luego de proceder con esto procedemos a remover ciertas palabras de nuestra línea para poder ordenar de manera correcta el string, luego utilizamos la función "split" para crear un array de nuestra línea y así luego crear un objeto, sin embargo si el try catch falla, podemos indicar que el producto no contenía las características necesarias para poder crear un nuevo producto.

```
def lectura():

    file = filedialog.askopenfilename(initialdir="C:/Users/sebas/Documents/USAC/Segundo Semestre 2023/Lab LFP/Práctica 1/Archivos de
Entrada",title="Elige un archivo de entrada", filetypes=(("Archivos de Datos","*.inv*"),("Todos los archivos","*.*")))

    with open(file, mode="r") as archivo:
        for linea in archivo:
            if "crear_producto" in linea:
                linea = linea.replace("crear_producto","")
                linea = linea.replace("\n","")
                splitted_line = linea.split(";")
                try:
                    name = splitted_line[0]
                    quantity = int(splitted_line[1])
                    price = float(splitted_line[2])
                    location = splitted_line[3]
                    newProduct = Producto(name,quantity,price,location)
                    db.append(newProduct)
                except:
                    print(Fore.RED+"EL PRODUCTO NO PUEDE SER CREADO, NO CUMPLE CON LOS REQUERIMIENTOS"+Fore.WHITE)
        archivo.close()
        root.destroy()
```

MOVIMIENTOS

LECTURA Y ANÁLISIS DE ARCHIVO DE MOVIMIENTOS

```
def movimientos():

    file = input(Fore.CYAN+"INGRESE LA RUTA DEL ARCHIVO"+"\n"+Fore.WHITE)
    file = file.replace("\\","/")
    file = file.replace('"',"")

    with open(file, mode="r") as archivo:
        for linea in archivo:
            if "agregar_stock" in linea:
                linea = linea.replace("agregar_stock","")
                linea = linea.replace("\n","")
                splitted_line = linea.split(";")
                name = splitted_line[0]
                quantity = int(splitted_line[1])
                location = splitted_line[2]
                for i in range(len(db)):
                    if name == (db[i].nombre):
                        if location == (db[i].ubicacion):
                            if quantity>=0:
                                db[i].cantidad=(db[i].cantidad)+quantity
                            else:
                                print(Fore.RED+"LA CANTIDAD INGRESADA ES NEGATIVA"+Fore.WHITE)
                        else:
                             print(Fore.RED+"NO EXISTE EL PRODUCTO"+Fore.YELLOW,name,Fore.RED+"EN LA",location+Fore.WHITE)
```

LECTURA

La lectura de el archivo de movimientos funciona de una manera bastante similar, por no decir idéntica a la de la lectura del archivo inicial, la única diferencia es que en este caso la lectura únicamente empezara si se encuentra alguna de las dos palabras "agregar_stock" o "vender_producto", de esta manera el proceso de lectura empezará tal cual lo hace en la lectura del archivo inicial (.inv)

ANÁLISIS

Para el análisis se manejan varias condicionales para asegurarnos de que todos los datos están siendo seleccionados, ordenados y contados de manera correcta, de esta manera podemos ser capaces de reunir la lista de errores causados durante la lectura del archivo y de la misma manera poder ordenar en una lista de objetos "Producto" y de esta manera actualizar cuando sea conveniente y necesario según el estado de la acción a realizar.

INVENTARIO

VISUALIZACIÓN DE INVENTARIO

Para la visualización de inventario es bastante sencillo el proceso de implementar lo siguiente, incluye un único condicional que trabaja de manera bastante sencilla, el inventario esta organizado por cada característica de los productos que tenemos creados en nuestro array objetos de tipo Producto, y es tan fácil el imprimir en consola estos datos como simplemente recorrer un array de objetos, sin embargo en este caso por intentar obtener un poco de más estética en el uso de la consola, se ha intentado alinear lo mejor posible el texto, de esta manera el único condicional que incluye la función entra en acción, determinando el número de caracteres del nombre del producto, basado en esto añade o remueve espacios para intentar alinear lo más posible las columnas en la consola