

A.

Práctica Evaluativa – Recuperatorio Parcial I

Asignatura: Programación I

Tema: Estructuras Repetitivas con Listas

Desafío: Una empresa necesita gestionar su stock de productos de forma sencilla. Para ello, se desea desarrollar un programa que permita:

- 1. Ingresar una lista con los nombres de los productos disponibles.
- 2. Ingresar una lista con las cantidades correspondientes a cada producto.
- 3. Mostrar un listado con el stock disponible.
- 4. Permitir al usuario consultar el stock de un producto específico.
- 5. Informar cuáles productos tienen stock agotado (cantidad igual a cero).

Requisitos Técnicos

- Utilizar exclusivamente estructuras repetitivas (for, while), estructuras condicionales (if, elif, else) y listas.
- No se permite el uso de diccionarios ni estructuras avanzadas.

Sugerencias:

- Utilizar dos listas paralelas: una para los nombres de productos y otra para las cantidades.
- Para consultar el stock de un producto, buscar su índice en la lista de nombres y utilizar ese índice en la lista de cantidades.
- Si un producto está agotado (es decir, no tiene stock), debe mantenerse en ambas listas, asignando el valor 0 en su posición correspondiente dentro de la lista de cantidades.

Menú del programa

- Menú de opciones:
- 1. Agregar producto
- 2. Ver productos agotados
- 3. Actualizar stock
- 4. Ver todos los productos
- 5. Salir



Entregables

El estudiante deberá subir el archivo del programa en lenguaje Python a la plataforma institucional.

El código debe cumplir con:

- Todas las funcionalidades solicitadas reflejadas en el menú.
- Buena ejecución sin errores.
- Nomenclatura clara en el nombre de las variables.
- Legibilidad general y buenas prácticas de codificación.

Rúbrica de Evaluación – Práctica: Gestión de Stock con Listas en Python

Criterio	Excelente (10)	Bueno (7)	Regular (4)	Insuficiente (1)
Manejo de listas paralelas	Usa correctamente listas para nombres y cantidades, de forma paralela y coherente.	Usa listas con pequeños errores de paralelismo.	Usa listas pero con errores que afectan el funcionamiento.	No usa listas o las usa incorrectamente.
Control de flujo con bucles	Usa estructuras repetitivas de forma eficiente y funcional.	Usa bucles con errores menores o redundancias.	Usa bucles con errores lógicos.	No usa bucles o están mal implementados.
Entrada y salida de datos	Agrega, muestra y actualiza datos de manera clara e interactiva.	Entrada/salida funcional con pequeños errores de formato.	Entrada/salida poco clara o incompleta.	No implementa entrada/salida o no funciona.
Menú de opciones	Menú funcional y fácil de usar.	Menú funcional pero mejorable.	Menú confuso o con errores.	No hay menú o no funciona.
Buenas prácticas de programación	Código limpio, legible, con nombres adecuados y comentarios útiles.	Código legible, con algunos aspectos a mejorar.	Código desordenado o poco claro.	Código difícil de entender o desorganizado.



Ejemplo orientativo – Pseudocódigo

INICIAR dos listas vacías:

nombres \leftarrow []

cantidades \leftarrow []

INICIAR variable bandera:

salir = verdadero

MIENTRAS salir HACER:

MOSTRAR menú de opciones:

- 1. Agregar producto
- 2. Ver productos agotados
- 3. Actualizar stock
- 4. Ver todos los productos
- 5. Salir

LEER opción del usuario

SI opción = "1" ENTONCES:

PEDIR nombre del producto \rightarrow nombre

PEDIR cantidad en stock \rightarrow cantidad

AGREGAR nombre a lista nombres

AGREGAR cantidad a lista cantidades

MOSTRAR "Producto agregado con éxito"

SINO SI opción = "2" ENTONCES:

MOSTRAR "Productos agotados:"

agotados ← falso

PARA i DESDE 0 HASTA longitud(nombres) - 1:

SI cantidades[i] = 0 ENTONCES:



MOSTRAR nombres[i]

agotados ← verdadero

SI agotados = falso ENTONCES:

MOSTRAR "No hay productos agotados"

SINO SI opción = "3" ENTONCES:

PEDIR nombre del producto → producto

SI producto EXISTE en nombres ENTONCES:

index ← posición de producto en nombres

PEDIR nueva cantidad → nueva_cantidad

 $cantidades[index] \leftarrow nueva_cantidad$

MOSTRAR "Stock actualizado"

SINO:

MOSTRAR "Producto no encontrado"

SINO SI opción = "4" ENTONCES:

MOSTRAR "Listado de productos:"

PARA i DESDE 0 HASTA longitud(nombres) - 1:

MOSTRAR nombres[i], cantidades[i]

SINO SI opción = "5" ENTONCES:

MOSTRAR "Gracias por usar el sistema"

salir = falso

SINO:

MOSTRAR "Opción inválida"