

RESPUESTAS CUESTIONARIO

Mariana Vallejo Bustillo

1. ¿Qué estructura condicional utilizaste para validar que el ID del producto sea único? Explica el proceso de validación.

Para asegurar que cada producto tenga un ID irrepetible, utilicé un contador que siempre guarda el último ID asignado. Cada vez que se registra un nuevo artículo, ese contador incrementa en uno y se utiliza como ID. Aunque se borren productos, el generador de IDs continúa desde el número más alto previamente usado, por lo que nunca se repite un identificador.

2. En el sistema de ventas, ¿cómo implementaste la condición para verificar si hay suficiente stock antes de realizar una venta?

Luego de confirmar que el ID existe en el inventario, recorremos los productos hasta encontrarlo. Cuando coincide el ID, reviso el valor del campo "stock"; si es menor a 1, se muestra un mensaje indicando que no es posible agregarlo por falta de existencias. Solo si el stock es mayor a cero se permite continuar con la venta.

3. Proporciona un ejemplo de uso de elif en tu código y explica por qué fue necesario.

Durante la sección de ordenamiento del inventario se emplea una cadena de elif como esta:

```
if opt == 1:
    inv = ordenar_inventario(inv, "nombre")
elif opt == 2:
    inv = ordenar_inventario(inv, "nombre", True)
elif opt == 3:
    inv = ordenar_inventario(inv, "proveedor")
...
```

Este patrón era necesario porque cada opción del menú desencadena un tipo distinto de ordenamiento. Usar varios elif permite cubrir cada caso de forma clara y directa.

4. ¿Qué tipo de ciclo utilizaste para el menú principal y por qué elegiste ese tipo?

Para el menú general utilicé un while True:. Este tipo de ciclo funciona muy bien en menús, porque permanece activo indefinidamente hasta que el usuario elige explícitamente salir. Así no hace falta definir una condición externa.

5. Explica la diferencia entre usar for y while en el contexto de tu proyecto. Da un ejemplo de cada uno.

El ciclo for lo usé para recorrer colecciones como el inventario o el carrito, por ejemplo:

```
for llave, producto in inv.items():  
    imprimir_producto_inventario(producto)
```

En cambio, while lo apliqué cuando necesitaba repetir una acción hasta recibir datos válidos o hasta que el usuario decida detenerse. Ejemplo:

```
while True:  
    cad = input("> ")  
    if cad.strip() != "":  
        return cad
```

6. ¿Cómo implementaste la búsqueda de productos usando ciclos? ¿Qué pasaría si tuvieras 10,000 productos?

La búsqueda se hace recorriendo todo el inventario:

```
for id_prod, prod in inv.items():  
    if texto in prod[criterio].lower():  
        productos.append((id_prod, prod))
```

Cada coincidencia se guarda en una lista auxiliar. Con miles de productos, el proceso tardaría un poco más, pero seguiría funcionando porque es una operación lineal y depende principalmente del rendimiento del equipo.

7. Lista las cinco funciones más importantes de tu sistema y describe brevemente qué hace cada una.

- **ver_productos_inventario()**
Muestra todos los productos organizados y con formato en consola.
 - **detectar_valor_entero_positivo()**
Verifica que el usuario introduzca valores válidos para menús y cantidades.
 - **detectar_existencia_inventario()**
Comprueba que el ID ingresado corresponda a un producto existente.
 - **finalizar_compra()**
Realiza el total de la venta y crea un ticket con la información del carrito.
 - **menu_producto()**
Permite editar atributos o eliminar un producto específico.
-

8. ¿Por qué es importante usar funciones en lugar de escribir todo el código en el programa principal?

El uso de funciones evita que el programa se vuelva un bloque gigante difícil de mantener. Permiten organizar tareas, reutilizar procesos y localizar errores más fácilmente. Además, mejora la lectura general del código y lo mantiene ordenado.

9. Describe una función que reciba parámetros y retorne un valor. ¿Qué ventaja tiene usar return?

Ejemplo:

```
def obtener_mayor_id(inv):  
    if not inv: # si está vacío  
        return -1  
    return max(inv.keys())+1
```

El return permite enviar de vuelta el resultado calculado, lo cual hace posible reutilizar ese valor en cualquier parte del programa sin depender de variables globales o efectos secundarios.

10. ¿Qué estructura de datos elegiste para almacenar los productos? Justifica tu elección.

Opté por un diccionario cuyos valores también son diccionarios. Esta estructura permite manejar cada producto mediante un ID único y acceder rápidamente a su información. Es más flexible que las listas para este tipo de datos, ya que evita IDs duplicados y permite búsquedas directas.

11. Explica cómo agregaste un nuevo producto al inventario usando operaciones con arreglos.

Para añadir un nuevo artículo, primero valido los datos del usuario. Luego asigno un ID libre y coloco el producto así:

```
inv[identificador] = {  
    "ID": identificador,  
    "nombre": nombre,  
    "categoria": categoria,  
    "precio": precio,  
    "proveedor": proveedor,  
    "stock": stock  
}
```

Python crea automáticamente la nueva entrada del inventario cuando se escribe sobre una llave que aún no existe. Finalmente, incremento el ID para mantener la secuencia.

12. ¿Cómo implementaste la eliminación de un producto? ¿Qué método utilizaste?

Una vez que el usuario confirma la eliminación, uso `.pop()` para quitar el producto:

```
inventario.pop(prod["ID"])
```

Primero verifico que el producto exista y luego utilizo el ID almacenado dentro de la propia estructura del producto para eliminarlo de forma segura.

13. ¿Cuál fue el mayor desafío técnico que enfrentaste durante el desarrollo y cómo lo resolviste?

La parte más complicada fue el diseño de las tablas para mostrar los productos, aunque me guíe con tablas prediseñadas en internet, el acomodar la longitud de los espacios para cada atributo y buscar utilizar caracteres fue un buen reto. Después de mucha prueba y error al final pude dar con las cantidades a utilizar para los espacios, las siguientes tablas ya fueron más sencillas, aunque aún tardadas.

14. Si tuvieras que agregar una nueva funcionalidad (por ejemplo, sistema de descuentos), ¿cómo la integrarías al código existente?

Sería un sistema de descuentos por fecha, dependiendo del mes o hasta día de la semana, ciertos artículos podrían tener un descuento aplicado. Esto se lograría consiguiendo la fecha actual y comparándola con algún atributo dentro de los productos que indique las fechas a tener descuento disponible.

15. ¿Qué aprendiste sobre trabajo en equipo durante este proyecto? ¿Cómo se organizaron para evitar conflictos en el código?

Aprendí que dividir responsabilidades acelera el desarrollo y reduce errores. Cada integrante tomó un módulo específico y, cuando era necesario integrarlos, se revisaban en conjunto para detectar incompatibilidades. Tener áreas bien delimitadas evitó que sobrescribiéramos el trabajo de otros.