

Ejercicio 1. Diseñar una aplicación Python que trabaje con objetos de la clase “Producto”. Esta clase tendrá las siguientes características:

- Atributos: identificador, nombre del producto, categoría, precio de venta, stock y stock mínimo.
- Métodos.
 - **actualizar_stock**. Recibe como argumento un número de unidades y las sumará al stock existente.
 - **esta_en_minimos**. Retorna True si el stock del producto es inferior al stock mínimo, False en caso contrario.
 - Define el método mágico que creas conveniente para mostrar el identificador, nombre y stock del producto.
 - Define el método mágico que consideres oportuno para determinar si dos productos son iguales. Se consideran iguales dos productos si tienen el mismo identificador.

Prueba las clases creadas de la siguiente forma:

- Instancia dos objetos.
- Imprime ambos objetos.
- Actualiza las unidades del primero.
- Imprime el primer objeto.
- Comprueba si el segundo está en mínimos.
- Comprueba si son iguales.

Forma de entrega.

- Crea una carpeta con dos archivos Python, uno con la clase creada y otro con las pruebas.

Ejercicio 2. Diseñar una aplicación Python que sirva para gestionar un almacén de productos (utiliza la clase del ejercicio anterior). Esta gestión se realizará mediante una clase llamada “**GestionAlmacen**” que tendrá las siguientes características.

- Atributo de clase en el que guardar los productos. Debe contener varios productos de inicio.
- Métodos.
 - **nuevo_producto**. Recibe datos de un producto como argumentos y lo añade a la lista, siempre y cuando su identificador no exista en la misma. Retorna True si el producto se añadió a la lista, False en caso contrario.
 - **buscar_por_id**. Recibe un identificador de producto como argumento y lo busca en la lista. En caso de encontrarlo lo retorna, si no lo encuentra debe retornar el valor None.
 - **bajo_minimos**. Retorna una lista con los productos que estén bajo mínimos.
 - **listado**. Muestra en pantalla un listado de los productos existentes en el almacén.

Prueba el código de la siguiente forma:

- Llama al método "listado" y muestra los productos existentes.
- Usando el método "nuevo_producto", añade dos productos nuevos. El segundo debe tener el mismo identificador que el primero. Comprueba si se añaden los dos o uno solo.
- Llama al método "bajo_mínimos" y guarda en una lista los productos que estén en mínimos. Muestra el contenido de la lista.

Forma de entrega.

- Crea una carpeta con dos archivos Python, uno con las clase creadas y otro con las pruebas.

Ejercicio 3. Crea un paquete redistribuible con las clases "Producto" y "GestionAlmacen". Para realizar este ejercicio consulta el Anexo IV.

Forma de entrega.

- Entrega el archivo "tar.gz" generado.
- Instala el paquete en tu computadora y pruébalo.
- Indica cómo deben realizarse las importaciones para que yo pueda importar esas clases.

Importante:

- En la resolución de los ejercicios de la tarea se deben utilizar las sentencias, estructuras y formas de trabajo vistas en clase.

Criterios de calificación.

- Ejercicio 1. 3 puntos.
 - Diseño de la clase, 2 puntos.
 - Pruebas, 1 punto.
- Ejercicio 2. 6 puntos.
 - Diseño de la clase GestionAlmacen, 4 puntos.
 - Pruebas, 2 puntos.
- Ejercicio 3. 1 punto.