



UEA

UNIVERSIDAD

ESTATAL AMAZÓNICA

UNIVERSIDAD ESTATAL AMAZÓNICA



INGENIERÍA EN TECNOLOGÍAS DE INFORMACIÓN

PROGRAMACIÓN ORIENTADA A OBJETOS

TEMA:

SISTEMA DE GESTIÓN DE INVENTARIOS

ESTUDIANTE:

EDWIN FABIÁN NORIEGA BALDEON

DOCENTE:

ING. SANTIAGO ISRAEL NOGALES GUERRERO

NIVEL:

SEGUNDO NIVEL_PARALELO "A"

2025-2025

PUYO-ECUADOR



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

1. CÓDIGO DE GITHUB

https://github.com/EDFANOBA19/PROGRAMACION_ORIENTADA_A_OBJETOS

[OS.git](#)

2. REPOSITORIO DE GITHUB

The screenshot shows the GitHub repository page for 'PROGRAMACION_ORIENTADA_A_OBJETOS' by user EDFANOBA19. The repository is public and has 1 branch and 0 tags. A file named 'SEMANA_9_POO_Sistema de Gestión de Inventarios' is highlighted in red. The file content is a Python script for a class 'Producto' with attributes 'id_producto', 'nombre', 'cantidad', and 'precio'. The script includes methods for getters and setters.

```
1 # SISTEMA DE GESTION DE INVENTARIOS
2
3 # Clase Producto que representa un producto individual en el inventario
4 class Producto:
5     def __init__(self, id_producto, nombre, cantidad, precio):
6         # Uso atributos "privados" con _ para promover encapsulación
7         self._id = id_producto # ID único del producto, se asume que es string para mayor flexibilidad
8         self._nombre = nombre # Nombre descriptivo del producto
9         self._cantidad = cantidad # Cantidad disponible en inventario (int)
10        self._precio = precio # Precio unitario del producto (float)
11
12 # Métodos getters para acceder a atributos privados
13 def get_id(self):
14     return self._id
15
16 def get_nombre(self):
17     return self._nombre
18
19 def get_cantidad(self):
20     return self._cantidad
21
22 def get_precio(self):
23     return self._precio
24
25 # Métodos setters para modificar atributos (excepto ID que es único e inmutable)
26 def set_nombre(self, nombre):
27     self._nombre = nombre
```



2.1 CÓDIGO

SISTEMA DE GESTIÓN DE INVENTARIOS SIMPLE PARA UNA TIENDA

CLASE PRODUCTO:

Atributos: ID (único), nombre, cantidad y precio.

```
Sistema de Gestión de Inventarios.py x
1  # SISTEMA DE GESTION DE INVENTARIOS
2
3  # Clase Producto que representa un producto individual en el inventario
4  class Producto: 6 usages
5      def __init__(self, id_producto, nombre, cantidad, precio):
6          # Uso atributos "privados" con _ para promover encapsulación
7          self._id = id_producto      # ID único del producto, se asume que es string para mayor flexibilidad
8          self._nombre = nombre      # Nombre descriptivo del producto
9          self._cantidad = cantidad  # Cantidad disponible en inventario (int)
10         self._precio = precio      # Precio unitario del producto (float)
11
```

Métodos: Constructor, getters y setters para cada atributo.

```
12     # Métodos getters para acceder a atributos privados
13     def get_id(self): 2 usages (2 dynamic)
14         return self._id
15
16     def get_nombre(self): 1 usage (1 dynamic)
17         return self._nombre
18
19     def get_cantidad(self):
20         return self._cantidad
21
22     def get_precio(self):
23         return self._precio
24
25     # Métodos setters para modificar atributos (excepto ID que es único e immutable)
26     def set_nombre(self, nombre):
27         self._nombre = nombre
28
29     def set_cantidad(self, cantidad):
30         self._cantidad = cantidad
31
32     def set_precio(self, precio):
33         self._precio = precio
34
35     # Método especial para presentar el producto en forma legible (útil para mostrar en consola)
36     def __str__(self):
37         return f"ID: {self._id} | Nombre: {self._nombre} | Cantidad: {self._cantidad} | Precio: ${self._precio:.2f}"
```

Clase Inventario:

Atributos: Una lista de productos.

```
39 # Clase Inventario que agrupa y gestiona una colección de objetos Producto
40 class Inventario: 1 usage
41     def __init__(self):
42         self.productos = [] # Lista que almacenará objetos Producto
43
```



Métodos:

Añadir nuevo producto (asegurarse de que el ID sea único).

```
44     # Añadir producto al inventario siempre que el ID no exista (para evitar duplicados)
45     def agregar_producto(self, producto): 2 usages
46         if self.buscar_producto_por_id(producto.get_id()) is not None:
47             return False # No se añade si ya existe un producto con dicho ID
48         self.productos.append(producto)
49         return True
50
```

Eliminar producto por ID.

```
51     # Eliminar producto mediante identificador único
52     def eliminar_producto(self, id_producto): 1 usage
53         producto = self.buscar_producto_por_id(id_producto)
54         if producto:
55             self.productos.remove(producto)
56             return True
57         return False
```

Actualizar cantidad o precio de un producto por ID.

```
59     # Actualizar cantidad y/o precio del producto identificado por su ID
60     def actualizar_producto(self, id_producto, cantidad=None, precio=None): 1 usage
61         producto = self.buscar_producto_por_id(id_producto)
62         if producto:
63             # Solo actualizamos si se pasa un nuevo valor válido para cada atributo
64             if cantidad is not None:
65                 producto.set_cantidad(cantidad)
66             if precio is not None:
67                 producto.set_precio(precio)
68             return True
69         return False
70
```

Buscar producto(s) por nombre (puede haber nombres similares).

```
71     # Buscar producto por ID único (retorna objeto Producto o None si no existe)
72     def buscar_producto_por_id(self, id_producto): 5 usages
73         for producto in self.productos:
74             if producto.get_id() == id_producto:
75                 return producto
76         return None
77
78     # Buscar productos que contengan un texto en su nombre (búsqueda insensible a mayúsculas)
79     def buscar_productos_por_nombre(self, nombre): 1 usage
80         nombre = nombre.lower()
81         return [p for p in self.productos if nombre in p.get_nombre().lower()]
82
83     # Devuelve lista con todos los productos almacenados para mostrar
84     def mostrar_productos(self): 1 usage
85         return self.productos
86
```



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

Mostrar todos los productos en el inventario.

```
87 # Función para mostrar menú interactivo de opciones disponibles al usuario
88 def mostrar_menu(): 1usage
89     print("\n--- Sistema de Gestión de Inventarios ---")
90     print("1. Añadir nuevo producto")
91     print("2. Eliminar producto por ID")
92     print("3. Actualizar cantidad o precio de un producto")
93     print("4. Buscar producto(s) por nombre")
94     print("5. Mostrar todos los productos")
95     print("6. Salir")
96
97 # Función para pedir un número entero positivo al usuario con validación
98 def pedir_entero(mensaje, permitir_cero=False): 1usage
99     while True:
100         try:
101             valor = int(input(mensaje))
102             if valor < 0 or (valor == 0 and not permitir_cero):
103                 print("Por favor, ingresa un número válido.")
104                 continue
105             return valor
106         except ValueError:
107             print("Error: debes ingresar un número entero.")
108
109 # Función para pedir un número flotante positivo, con validación, para precios por ejemplo
110 def pedir_flotante(mensaje): 1usage
111     while True:
112         try:
113             valor = float(input(mensaje))
114             if valor < 0:
115                 print("Por favor, ingresa un número positivo.")
116                 continue
117             return valor
118         except ValueError:
119             print("Error: debes ingresar un número válido.")
120
121 def main(): 1usage
122     inventario = Inventario()
123
124     # Agrego productos de ejemplo para facilitar pruebas sin entrada constante del usuario
125     productos_ejemplo = [
126         Producto(id_producto: "P001", nombre: "Manzanas", cantidad: 50, precio: 0.5),
127         Producto(id_producto: "P002", nombre: "Naranjas", cantidad: 75, precio: 0.75),
128         Producto(id_producto: "P003", nombre: "Leche", cantidad: 30, precio: 1.25),
129         Producto(id_producto: "P004", nombre: "Pan", cantidad: 100, precio: 0.8),
130         Producto(id_producto: "P005", nombre: "Huevos", cantidad: 200, precio: 0.1),
131     ]
132     for p in productos_ejemplo:
133         inventario.agregar_producto(p)
134
135     print("Sistema iniciado con productos de ejemplo.\n")
136
137     # Bucle principal que muestra el menú y atiende la opción seleccionada
138     while True:
139         mostrar_menu()
140         opcion = input("Selecciona una opción: ")
141
142         if opcion == '1':
143             # Añadir nuevo producto, confirmando que el ID es único para mantener integridad
144             print("\n--- Añadir nuevo producto ---")
145             id_producto = input("Ingresa ID único: ").strip()
146             if inventario.buscar_producto_por_id(id_producto):
147                 print("Error: ya existe un producto con ese ID.")
```



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

```
148         continue
149     nombre = input("Nombre del producto: ").strip()
150     cantidad = pedir_entero("Cantidad: ")
151     precio = pedir_flotante("Precio: $")
152     producto = Producto(id_producto, nombre, cantidad, precio)
153     inventario.agregar_producto(producto)
154     print("Producto agregado correctamente.")
155
156 elif opcion == '2':
157     # Eliminar producto por su ID único, mostrando mensaje según resultado
158     print("\n-- Eliminar producto --")
159     id_producto = input("Ingresa el ID del producto a eliminar: ").strip()
160     if inventario.eliminar_producto(id_producto):
161         print("Producto eliminado correctamente.")
162     else:
163         print("No se encontró un producto con ese ID.")
164
165 elif opcion == '3':
166     # Actualizar cantidad y/o precio, permitiendo dejar campos vacíos para no actualizar
167     print("\n-- Actualizar producto --")
168     id_producto = input("ID del producto a actualizar: ").strip()
169     producto = inventario.buscar_producto_por_id(id_producto)
170     if producto is None:
171         print("No se encontró un producto con ese ID.")
172         continue
173     print(f"Producto actual: {producto}")
174     print("Deja en blanco el campo que no deseas actualizar.")
175     cantidad_input = input("Nueva cantidad: ")
176     precio_input = input("Nuevo precio: $")
177     cantidad = None
178     precio = None
179     if cantidad_input != "":
180         try:
181             cantidad = int(cantidad_input)
182             if cantidad < 0:
183                 print("La cantidad no puede ser negativa.")
184                 continue
185         except ValueError:
186             print("Cantidad inválida.")
187             continue
188     if precio_input != "":
189         try:
190             precio = float(precio_input)
191             if precio < 0:
192                 print("El precio no puede ser negativo.")
193                 continue
194         except ValueError:
195             print("Precio inválido.")
196             continue
197     if cantidad is None and precio is None:
198         print("No actualizaste ningún dato.")
199         continue
200     inventario.actualizar_producto(id_producto, cantidad, precio)
201     print("Producto actualizado correctamente.")
202
203 elif opcion == '4':
204     # Búsqueda parcial por nombre para permitir encontrar varios productos similares
205     print("\n-- Buscar productos por nombre --")
206     nombre_búsqueda = input("Ingresa parte o todo el nombre: ").strip()
207     resultados = inventario.buscar_productos_por_nombre(nombre_búsqueda)
```



UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

```
208         if resultados:
209             print(f"Se encontraron {len(resultados)} producto(s):")
210             for prod in resultados:
211                 print(prod)
212         else:
213             print("No se encontraron productos con ese nombre.")
214
215     elif opcion == '5':
216         # Mostrar todo el inventario, o notificar que está vacío
217         print("\n--- Lista de todos los productos ---")
218         productos = inventario.mostrar_productos()
219         if productos:
220             for prod in productos:
221                 print(prod)
222         else:
223             print("El inventario está vacío.")
224
225     elif opcion == '6':
226         print("Saliendo del sistema. ¡Hasta luego!")
227         break
228
229     else:
230         print("Opción no válida. Intenta nuevamente.")
231
232 # Punto de entrada del programa, garantiza que main() se ejecute solo si este archivo corre directamente
233 if __name__ == "__main__":
234     main()
235
```





INTERFAZ DE USUARIO EN LA CONSOLA:

```
Run Sistema de Gestión de Inventarios x
"C:\Users\hp\PycharmProjects\SEMANA_2_P00_PILARES DE PROGRAMACION ORIENTADA A OBJETOS\.venv\Scripts\python.exe" "C:\Users\hp\PycharmProjects\PROGRAMACION_ORIENTADA_A_OBJETOS\SE
Sistema iniciado con productos de ejemplo.

--- Sistema de Gestión de Inventarios ---
1. Añadir nuevo producto
2. Eliminar producto por ID
3. Actualizar cantidad o precio de un producto
4. Buscar producto(s) por nombre
5. Mostrar todos los productos
6. Salir
Selecciona una opción:
```

Crear un menú interactivo en la consola que permita al usuario realizar todas las operaciones mencionadas anteriormente

Añadir nuevo producto

```
Selecciona una opción: 1

--- Añadir nuevo producto ---
Ingresa ID único: P006
Nombre del producto: PAN
Cantidad: 5
Precio: $0.5
Producto agregado correctamente.
```

Eliminar producto por ID

```
Selecciona una opción: 2

--- Eliminar producto ---
Ingresa el ID del producto a eliminar: P006
Producto eliminado correctamente.
```




UEA

UNIVERSIDAD ESTATAL AMAZÓNICA

Actualizar cantidad o precio de un producto por ID

```
Selecciona una opción: 3

--- Actualizar producto ---
ID del producto a actualizar: P004
Producto actual: ID: P004 | Nombre: Pan | Cantidad: 100 | Precio: $0.80
Deja en blanco el campo que no deseas actualizar.
Nueva cantidad:
Nuevo precio: $0.50
Producto actualizado correctamente.
```

Buscar producto(s) por nombre (puede haber nombres similares)

```
Selecciona una opción: 4

--- Buscar productos por nombre ---
Ingresa parte o todo el nombre: PAN
Se encontraron 1 producto(s):
ID: P004 | Nombre: Pan | Cantidad: 100 | Precio: $0.50
```

```
Selecciona una opción: 4

--- Buscar productos por nombre ---
Ingresa parte o todo el nombre: pan
Se encontraron 2 producto(s):
ID: P004 | Nombre: Pan | Cantidad: 100 | Precio: $0.50
ID: P008 | Nombre: Pan | Cantidad: 5 | Precio: $0.60
```





UEA

UNIVERSIDAD
ESTATAL AMAZÓNICA

Mostrar todos los productos en el inventario

Selecciona una opción: 5

--- Lista de todos los productos ---

ID: P001		Nombre: Manzanas		Cantidad: 50		Precio: \$0.50
ID: P002		Nombre: Naranjas		Cantidad: 75		Precio: \$0.75
ID: P003		Nombre: Leche		Cantidad: 30		Precio: \$1.25
ID: P004		Nombre: Pan		Cantidad: 100		Precio: \$0.50
ID: P005		Nombre: Huevos		Cantidad: 200		Precio: \$0.10

