

Evaluación Final Modulo 1 Data Analytics

A lo largo de esta evaluación tenes que crear una clase llamada: *Tienda Online*

Tienda Online

Class *TiendaOnline*: → Estoy creando una tienda, donde guardo mis productos

Método constructor, se ejecuta automáticamente. **def __init__(self)** → Self es como una etiqueta de identificación

Para que se guarde dentro de la tienda **self.inventario = []** → estantería de la tienda

self.ventas_totales = 0.0 → caja registradora

Método 1: agregar producto

for producto in self.inventario: → Bucle for para recorrer la lista inventario

Comprobación ← **if producto["nombre"] == nombre:**

Modificación de cantidad ← **producto["cantidad"] = producto["cantidad"] + cantidad**

Actualización del precio ← **producto["precio"] = precio**

else → Lo uso para salir del bucle cuando el producto ya existe (anti-duplicado).

(nes por si no encuentro el producto, para crear uno nuevo) → **new_product = {"nombre": nombre, "precio": precio, "cantidad": cantidad}**.

Para esto se usa el **.append()**

Método 2: ver inventario

inventario(self): → Nos sirve para ver información,

del otro. **for producto in self.inventario:** → Bucle for para recorrer la lista inventario

Es primordial ← **print("Nombre:", producto["nombre"], ", ", "Precio:", producto["precio"], ", ", "Cantidad:", producto["cantidad"])**

Método 3: buscar producto

buscar_producto(self, nombre):

for producto in self.inventario: → bucle for busca un producto concreto

Si encuentro, imprimo datos y no return → **if producto["nombre"] == nombre:** → Compruebo

print("Nombre:", producto["nombre"], ", ", "Precio:", producto["precio"], ", ", "Cantidad:", producto["cantidad"])

return "Producto no encontrado"

return es como decir: Dara aquí.

Tras para ver pares key-value del diccionario:

for producto in self.inventario

for key, value in producto.items()

print(key, ":", value)

print("-----")

Método 4:

Actualizar stock

- stock (self, nombre, cantidad) : → Este método lo vamos a usar para sumar o restar unidades a un producto que ya existe.

for producto in self.inventario:

if producto ["nombre"] == nombre:

productos ["cantidad"] = producto ["cantidad"] + cantidad → Así actualizamos el stock

return → Cuando encuentra el producto y lo actualiza, llamamos el stock para el método

print ("Producto no encontrado")

Método 5:

Eliminar producto producto (self, nombre):

for producto in self.inventario: → Lo usaremos para borrar productos de la tienda

if producto ["nombre"] == nombre:

self.inventario.remove (producto) → Remove es para quitar el producto de la lista del inventario.

return → Para el método cuando lo encuentra

print ("Producto no encontrado") → Solo lo mostraremos si el for termina y no encuentra nada

Método 6:

Calcular el valor del almacén-inventario (self)

valor_total = 0 → Creo la variable para ir guardando la suma total

for producto in self.inventario: → Recorro la lista de inventario ((cada producto es un dic)

precio = producto ["precio"] → copio el precio y lo guardo en una variable

cantidad = producto ["cantidad"] → copio la cantidad y la guardo en una variable

valor_producto = precio * cantidad → Para hacer el cálculo

valor_total = valor_total + valor_producto → sumo el valor en lo que ya tengo

return valor_total → Me devuelve el resultado aún estando fuera del método.