

Pruebas Conceptuales de Algoritmos de Ordenamiento.

```
#Entidad Usuario
class Producto:

    def __init__(self, id = 0, nombre = "", descripcion = "", stock = 0,
                 precio = 0, categoria = "", usuario = None):
        self.id = id
        self.nombre = nombre
        self.descripcion = descripcion
        self.stock = stock
        self.precio = precio
        self.categoria = categoria
        self.usuario = usuario

#Controlador de Producto
class ProductoControlador:

    def run (self):
        self.inventario = []
        self.inventario.append(Producto(1, "Franela Roja", "Franela roja
deportiva", 10, 30.0, "Ropa Deportiva"))
        self.inventario.append(Producto(2, "Zapatos Rojo", "Zapatos Rojos
Deportivos", 5, 80.0, "Zapatos"))
        self.inventario.append(Producto(3, "Camisa Azul", "Camisa Azul
Manga Larga", 20, 20.0, "Ropa de Vestir"))
        self.inventario.append(Producto(4, "Franela Azul", "Franela azul
deportiva", 10, 30.0, "Ropa Deportiva"))
        self.inventario.append(Producto(5, "Pantalon Azul", "Pantalon Azul
de Vestir", 5, 40.0, "Ropa de Vestir"))
        self.inventario.append(Producto(6, "laptop hp", "computadora
portatil i5 2 TB", 6, 1200.00, "Computacion y Electronica"))
        self.inventario.append(Producto(7, "Pentdrive Azul 50GB", "Pent
drive de color azul, capacidad: 50gb", 15, 15.00, "Computacion y
Electronica"))
        self.inventario.append(Producto(8, "laptop Dell", "computadora
portatil i5 500GB", 3, 600.00, "Computacion y Electronica"))
        self.inventario.append(Producto(9, "Iphone", "", 10, 300.00,
"Computacion y Electronica"))

controlador = ProductoControlador()
controlador.run()
print(controlador.inventario[0].nombre)
```