

Registro de ventas

Programación orientada a objetos

Emilio Muciño Segura

1. Identifica y enlista las clases, atributos y métodos de dichas clases.

Clases:

1. Producto
2. PuntoDeVentas Atributos:

1. Clase Producto: -

- Id
- nombre
 - precio
 - stock

2. Clase PuntoDeVentas:

- Productos
- Carrito
- ventas_del_dia

Métodos:

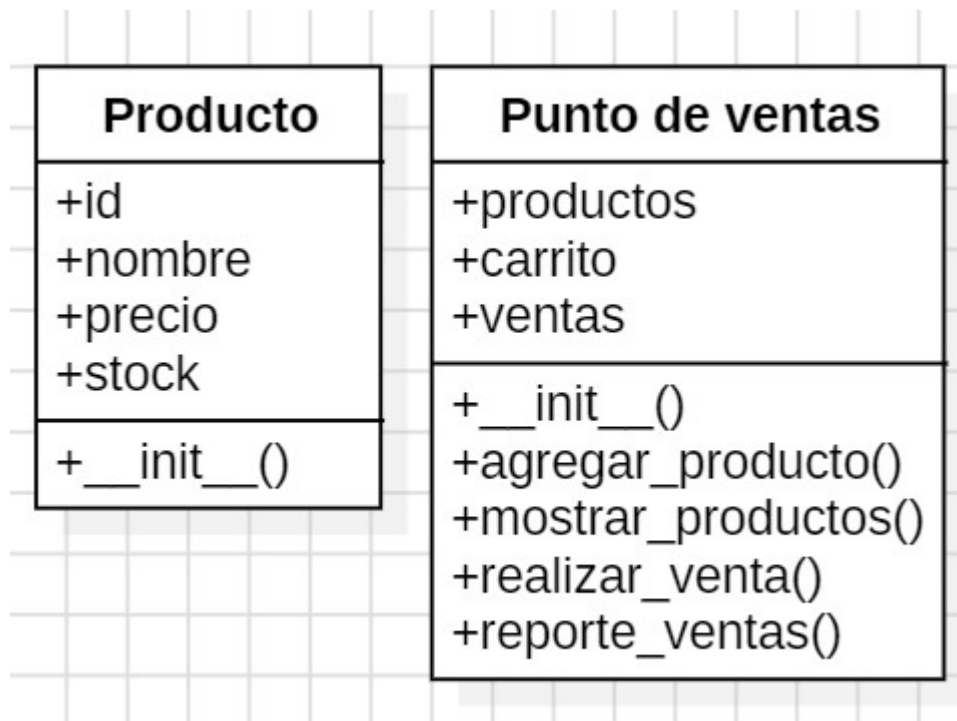
1. Clase Producto:

- __init__()
- ### 2. Clase PuntoDeVentas: - __init__()
- agregar_producto()
 - mostrar_productos()
 - realizar_venta()
 - reporte_ventas()

2. Identifica las relaciones entre los objetos.

- En la clase `PuntoDeVentas`, se almacenan instancias de la clase `Producto` en el atributo `productos`.
- En el método `realizar_venta()` de la clase `PuntoDeVentas`, se relaciona con la clase `Producto` al modificar los atributos `stock` de los productos en función de las ventas.
- En el método `realizar_venta()` de la clase `PuntoDeVentas`, se relaciona con la clase `Producto` al agregar productos al carrito de compras.
- En el método `reporte_ventas()` de la clase `PuntoDeVentas`, se accede a los atributos de las ventas del día, que contienen instancias de la clase `Producto`.
- En el programa principal, se crean instancias de la clase `Producto` y se llaman a métodos de la clase `PuntoDeVentas`, estableciendo relaciones entre las instancias de ambas clases.

3. Genera el diagrama de clases de la solución.



4. Genera el código de las clases y el código funcional para llegar a una solución a lo planteado en el caso.

Clase Producto

```
class Producto:
    def __init__(self, id, nombre, precio, stock):
        self.id=id
        self.nombre=nombre
        self.precio=precio
        self.stock=stock
```

Clase PuntoDeVentas

```
class PuntoDeVentas:
    def __init__(self):
```

```

        self.productos=[]
self.carrito=[]
self.ventas_del_dia=[]
        def agregar_producto(self,
producto):
self.productos.append(producto)
        def
mostrar_productos(self):
        print("\nLista de productos:")
        for idx,
producto in enumerate(self.productos, start=1):
            print(f"{idx}. {producto.nombre} \tstock: {producto.stock}
\tprecio: ${producto.precio}")
        def
realizar_venta(self):
        print("\nCarrito de compras")
while True:
            self.mostrar_productos()
opcion=input("\nTerminar ventas (TV) o agregar otro producto
(AP): ").upper()
if opcion=='TV':
            break
elif opcion=='AP':
            id_producto=input("Ingresar código de producto: ")
producto=next((p for p in self.productos if
p.id==id_producto), None)
            if
producto and producto.stock>0:
                cantidad=int(input("Cantidad: "))
if cantidad<=producto.stock:
                    total=cantidad*producto.precio
self.carrito.append((producto, cantidad, total))
producto.stock-=cantidad
                    print(f"Total a pagar: ${total:.2f}")
else:
                    print("*** Error: ya no hay stock disponible ***")
else:
                    print("*** Error: producto no encontrado o sin stock
disponible ***")
                    else:
                        print("*** Opción no válida ***")
                        tventa=sum(item[2] for item in self.carrito)
tarticulos=sum(item[1] for item in self.carrito)
self.ventas_del_dia.append((len(self.carrito), tventa, tarticulos))
self.carrito.clear()

```

```

        print(f"\nVenta realizada exitosamente:\nImporte total de ventas:
${tventa:.2f}\nCantidad de artículos: {tarticulos}\n")
    def
    reporte_ventas(self):
        print("\nReporte de ventas del día:")
    tventas=len(self.ventas_del_dia)          timporte=sum(venta[1]
    for venta in self.ventas_del_dia)          tarticulos=sum(venta[2]
    for venta in self.ventas_del_dia)          tpiezas=sum(venta[0]
    for venta in self.ventas_del_dia)
        print(f"Cantidad de ventas: {tventas}")
    print(f"Importe total de ventas: ${timporte:.2f}")
    print f"Cantidad total de artículos: {tarticulos}"
    print f"Total de piezas vendidas: {tpiezas}"

```

Programa

Principal

```

from ClaseProducto import Producto from
ClasePuntoDeVentas import PuntoDeVentas
if __name__ ==
"__main__":
    punto_de_ventas=PuntoDeVentas()
    punto_de_ventas.agregar_producto(Producto("1", "Silla de madera",
550.50, 5))
    while True:
        print("\nCarrito de compras\n")          opcion=input("A. Listar
productos\nB. Agregar producto\nC. Realizar venta\nS. Salir\n\nElige una
opción: ").upper()
        if
    opcion=="A":
        punto_de_ventas.mostrar_productos()
    elif opcion=="B":
        id=input("\nID: ")          nombre=input("Nombre del
producto: ")          precio=float(input("Precio: "))
        stock=int(input("Stock: "))
        punto_de_ventas.agregar_producto(Producto(id, nombre, precio, stock))
        print("\n** Producto agregado exitosamente en el carrito **")
    elif opcion=="C":
        punto_de_ventas.realizar_venta()
    elif opcion=="S":
        punto_de_ventas.reporte_ventas()
    print("\nSaliendo del programa...")

```

```
        break
    else:
        print("Opción inválida. Por favor, selecciona una opción válida.")
```

Vista del programa:

```
Carrito de compras

A. Listar productos
B. Agregar producto
C. Realizar venta
S. Salir

Elige una opción: A

Lista de productos:
1. Silla de madera      stock: 5      precio: $550.5

Carrito de compras

A. Listar productos
B. Agregar producto
C. Realizar venta
S. Salir

Elige una opción: B

ID: 2
Nombre del producto: Mesa de madera
Precio: 609.99
Stock: 8

** Producto agregado exitosamente en el carrito **
```

Carrito de compras

- A. Listar productos
- B. Agregar producto
- C. Realizar venta
- S. Salir

Elige una opción: C

Carrito de compras

Lista de productos:

- | | | |
|--------------------|----------|------------------|
| 1. Silla de madera | stock: 5 | precio: \$550.5 |
| 2. Mesa de madera | stock: 8 | precio: \$609.99 |

Terminar ventas (TV) o agregar otro producto (AP): AP

Ingresar código de producto: 2

Cantidad: 1

Total a pagar: \$609.99

Lista de productos:

- | | | |
|--------------------|----------|------------------|
| 1. Silla de madera | stock: 5 | precio: \$550.5 |
| 2. Mesa de madera | stock: 7 | precio: \$609.99 |

Terminar ventas (TV) o agregar otro producto (AP): AP

Ingresar código de producto: 1

Cantidad: 3

Total a pagar: \$1651.50

Lista de productos:

1. Silla de madera	stock: 2	precio: \$550.5
2. Mesa de madera	stock: 7	precio: \$609.99

Terminar ventas (TV) o agregar otro producto (AP): TV

Venta realizada exitosamente:

Importe total de ventas: \$2261.49

Cantidad de artículos: 4

Carrito de compras

- A. Listar productos
- B. Agregar producto
- C. Realizar venta
- S. Salir

Elige una opción: S

Reporte de ventas del día:

Cantidad de ventas: 1

Importe total de ventas: \$2261.49

Cantidad total de artículos: 4

Total de piezas vendidas: 2

Saliendo del programa...