

# Solución actividad POO II

## Paso 1: Crear la clase **PrendaDeVestir**

```
// Clase PrendaDeVestir que representa una prenda en el inventario
public class PrendaDeVestir {

    // Atributos de la clase PrendaDeVestir
    private String codigo; // Identificador único de la prenda
    private String nombre; // Nombre o descripción de la prenda
    private String marca; // Marca de la prenda
    private String categoria; // Categoría a la que pertenece la prenda (ej. camisas, pantalones)
    private boolean disponibilidad; // Indica si la prenda está disponible en el inventario

    // Constructor de la clase PrendaDeVestir
    public PrendaDeVestir(String codigo, String nombre, String marca, String categoria, boolean disponibilidad) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.marca = marca;
        this.categoria = categoria;
        this.disponibilidad = disponibilidad;
    }

    // Método para registrar la venta de una prenda
    public void vender() {
```

```

        if (disponibilidad) {
            disponibilidad = false; // Marca la prenda como no
            o disponible
            System.out.println("Prenda vendida: " + nombre);
        } else {
            System.out.println("Prenda no disponible: " + nom
bre);
        }
    }

    // Método para reponer el inventario de la prenda
    public void reponerInventario() {
        disponibilidad = true; // Marca la prenda como dispon
ible nuevamente
        System.out.println("Prenda repuesta: " + nombre);
    }

    // Método para obtener la información de la prenda
    public String getInfo() {
        return "Código: " + codigo + ", Nombre: " + nombre +
", Marca: " + marca +
        ", Categoría: " + categoria + ", Disponible: "
+ (disponibilidad ? "Sí" : "No");
    }
}

```

## Paso 2: Crear la clase **Cliente**

```

/*
CLASE Cliente
*/
package tiendaropita;

```

```

public class Cliente {
    //Atributos
    private String numeroCliente;//id cliente
    private String nombre;
    private String prendasAdquiridas; //Prendas adquiridas por el cliente.

    //Constructor

    public Cliente(String numeroCliente, String nombre) {
        this.numeroCliente = numeroCliente;
        this.nombre = nombre;
        this.prendasAdquiridas = "";
    }
    //Métodos
    public void realizarCompra(PrendaDeVestir prenda){
        prenda.vender();//Llamando al método vender de PDV para la prenda quede en estado no disponible.
        prendasAdquiridas = prenda.getInfo();//Guardamos la info de la Prenda adquirida
        System.out.println( nombre + " ha comprado " + prendasAdquiridas);
    }
    public void devolverPrenda(PrendaDeVestir prenda){
        if (!prendasAdquiridas.isEmpty()){
            prenda.reponerInventario();//llama al método reponerInventario para dejar la prenda como disponible
            System.out.println(nombre + " ha devuelto " + prendasAdquiridas);
            prendasAdquiridas = "";//Reseteamos la prenda después de su devolución.
        }else{
            System.out.println( nombre +" no tiene prendas para devuelve, compre algo primero ");
        }
    }
}

```

```

    public String getInfo(){
        return "Número cliente: " + numeroCliente + ", Nombre: " + nombre + ", Prendas Adquiridas: " + prendasAdquiridas;
    }

```

### Paso 3: Crear la clase **ProveedorDeModa**

```

/*
CLASE ProveedorDeModa
*/
package tiendaropita;

public class ProveedorDeModa {
    //Atributos
    private String codigoProveedor;
    private String nombre;
    private String prendasSuministradas;
    //Constructor
    public ProveedorDeModa(String codigoProveedor, String nombre) {
        this.codigoProveedor = codigoProveedor;
        this.nombre = nombre;
        this.prendasSuministradas = "";
    }
    //Método para agregar el suministro de una prenda por el proveedor.
    public void suministrarPrenda(PrendaDeVestir prenda){
        prendasSuministradas = prenda.getInfo();//Guarda la info de la prenda.
    }
}

```

```

        System.out.println(nombre + " ha suministrado: " + pre
ndasSuministradas);

    }

    public String getInfo(){
        return "Código del proveedor: " + codigoProveedor +
"Nombre: " + nombre + "Prendas Suministradas: " + prendasSumi
nistradas;
    }

}

```

#### Paso 4: Crear la clase **SistemaDeGestionDeInventarios**

Esta clase gestionará las operaciones del sistema de inventario.

Dado que la clase **SistemaDeGestionDeInventarios** maneja varias operaciones relacionadas con el registro y manejo de prendas, clientes y proveedores, puede ser suficiente utilizar métodos específicos para cada tipo de registro o actualización, sin necesidad de un estado inicial complejo que requiera configuración a través de un constructor. Esto permite que los objetos de esta clase se inicialicen en un estado "vacío" y se configuren gradualmente mediante llamadas a sus métodos.

```

/*
CLASE SistemaDeGestionDeInventarios
*/
package tiendaropita;

public class SistemaDeGestionDeInventarios {

```

```

//Atributos
private String catalogoPrendas;
private String clientesRegistrados;
private String proveedoresActivos;

//Métodos para registrar una prenda
public void registrarPrenda(PrendaDeVestir prenda){
    catalogoPrendas = prenda.getInfo();//Almaceno la info
del catálogo
    System.out.println("Prenda registrada: " + catalogoPr
endas);
}
//Método para registrar Cliente
public void registrarCliente(Cliente cliente){
    clientesRegistrados = "Cliente: " + cliente.getInfo
();
    System.out.println("Cliente registrado" + clientesReg
istrados);
}
//Método para registrar un Proveedor
public void registrarProveedor(ProveedorDeModa proveedor)
{
    proveedoresActivos = "Proveedor"+ proveedor.getInfo
();//Guardamos la info del proveedor.
    System.out.println("Proveedor registrado: " + proveed
oresActivos);
}

//Método para verificar la disponibilidad de una prenda
public void verificarDisponibilidad(PrendaDeVestir prenda){
    System.out.println("verificando disponibilidad de: "+
prenda.getInfo());
}
//Método para generar un informe de ventas
public void generarInformeVentas(){

```

```

        System.out.println("Generando infome de ventas");
        //Acá va la lógica para crear un informe....
    }

```

## Paso 5: Implementar el código en `main` para probar el sistema

Este paso incluye la creación de instancias de las clases anteriores y la simulación de operaciones en el sistema.

```

/*
MAIN PRINCIPAL
*/
package tiendaropita;

public class TiendaRopita {
    public static void main(String[] args) {
        //Crear instancias de clases objetos
        //2 Objetos prendas
        PrendaDeVestir prenda1 = new PrendaDeVestir("484848",
"Polera de algodón", "marca: chanchito","poleras",true);
        PrendaDeVestir prenda2 = new PrendaDeVestir("858585",
"Gorro", "marca: chanchito","accesorios",true);
        //2 Clientes
        Cliente cliente1 = new Cliente("Cl001","Juan Perez");
        Cliente cliente2 = new Cliente("Cl002","Jhon Doe");
        //Proveedor de moda
        ProveedorDeModa proveedor1 = new ProveedorDeModa("Pr0
202", "Nike");
        ProveedorDeModa proveedor2 = new ProveedorDeModa("Pr0
303", "Adidas");
        //Crear un objeto SDGDI = sistema Este objeto será el
encargado de manejar todas las operaciones relacionadas con e
l inventario, clientes y proveedores.
        SistemaDeGestionDeInventarios sistema = new SistemaDe

```

```

GestionDeInventarios();
    //Registramos las prendas
    sistema.registrarPrenda(prenda1);
    sistema.registrarPrenda(prenda2);
    //Registramos los clientes
    sistema.registrarCliente(cliente1);
    sistema.registrarCliente(cliente2);
    //Registrar Proveedor
    sistema.registrarProveedor(proveedor1);
    sistema.registrarProveedor(proveedor2);
    //Realizar una compra
    cliente1.realizarCompra(prenda1);
    //Realizar una devolución
    cliente1.devolverPrenda(prenda1);
    //Proveedor suministra prenda
    proveedor1.suministrarPrenda(prenda1);
    //Verificar la disponibilidad de una prenda
    sistema.verificarDisponibilidad(prenda1);
    //Generar un informe de ventas
    sistema.generarInformeVentas();
    //Registrar prendas, cleintes y proveedores en el si
sistema
    }
}

```

1. **Clase** `PrendaDeVestir` : Maneja la información de una prenda, incluyendo su código, nombre, marca, categoría y disponibilidad. Los métodos permiten vender y reponer la prenda, así como obtener su información detallada.
2. **Clase** `Cliente` : Representa a un cliente en el sistema, permitiendo registrar compras y devoluciones de prendas. Los métodos utilizan la información de la prenda adquirida o devuelta para actualizar el estado del cliente.
3. **Clase** `ProveedorDeModa` : Representa a un proveedor, con métodos para registrar el suministro de prendas. La información de la prenda suministrada se guarda



y muestra.

4. **Clase** `SistemaDeGestionDeInventarios` : Centraliza la gestión del inventario, incluyendo el registro de prendas, clientes y proveedores, la verificación de disponibilidad, y la generación de informes de ventas.
5. **Clase** `Main` : Implementa un ejemplo de uso del sistema, demostrando cómo se crean instancias de las clases, se registran en el sistema, y se simulan operaciones como compras, devoluciones y suministro de prendas.