

CÓDIGO FUENTE GENERADO A PARTIR DEL DIAGRAMA DE CLASES

PRESENTADO A:

INST. ANDRES FERNANDO SANCHEZ SOLARTE

PRESENTADO POR:

GRUPO NUMERO 3

CARLOS JOSE DELGADO GONZALEZ

DURLEY SANDRITH GALVAN JIMENEZ

JORGE ANDERSON CORTES TORRES

YASSER LEONARDO PACHECO CAÑIZARES

PROGRAMA:

**PROGRAMACION DE APLICACIONES PARA DISPOSITIVOS MOVILES
(2977832)**

SERVICIO NACIONAL DE APRENDIZAJE

SENA

2024

INTRODUCCIÓN

En el siguiente informe nos permitimos explicar y analizar el código fuente en lenguaje JAVA sobre un software que va a facilitar la administración de inventario, gestión de clientes, manejo de pedidos y control de proveedores en una joyería artesanal. A lo largo de este trabajo, exploraremos la estructura del sistema, sus principales componentes y cómo estos interactúan para proporcionar una solución integral a los desafíos que enfrenta una joyería artesanal en el ámbito de la gestión y organización de dicho negocio.

DESARROLLO

Resumen del Código Java

El código Java presentado define un sistema de gestión para una joyería artesanal, estructurado en varias clases principales:

1. **Articulo:** Representa las joyas o productos de la joyería. Incluye atributos como id, nombre, descripción, precio, material y stock. Ofrece métodos para crear joyas, actualizar stock, calcular precios y listar artículos.
2. **Cliente:** Una clase interna de Articulo que maneja la información de los clientes, incluyendo métodos para registrar clientes, actualizar información y realizar pedidos.
3. **Catalogo:** Otra clase interna de Articulo que gestiona la colección de artículos disponibles. Permite agregar o eliminar artículos, actualizar precios y buscar artículos específicos.
4. **Pedido:** Maneja la información relacionada con los pedidos de los clientes, incluyendo id, fecha, estado y total. Proporciona métodos para crear pedidos y actualizar su estado.
5. **Inventario:** Controla el stock de productos, con funcionalidades para actualizar cantidades y verificar niveles de stock.
6. **Proveedor:** Gestiona la información de los proveedores, permitiendo realizar pedidos y actualizar información de contacto.

El sistema está diseñado para interactuar entre estas clases, facilitando operaciones como la creación de artículos, gestión de pedidos, control de inventario y manejo de relaciones con proveedores y clientes.

Este diseño modular permite una fácil expansión y mantenimiento del sistema, adaptándose a las necesidades cambiantes de una joyería artesanal en crecimiento.

CÓDIGO FUENTE GENERADO A PARTIR DEL DIAGRAMA DE CLASES

```
import java.util.Date;
import java.util.List;
import java.util.ArrayList;
import java.io.File;

public class Diagrama {
    public static void main(String[] args) {
        //Codigo fuente para Diagrama de Clases sobre una app para una joyeria Artesanal
    }

    private static class Articulo {
        // Atributos
        private int id;
        private String nombre;
        private String descripcion;
        private double precio;
        private String material;
        private int stock;

        // Constructor
        public Articulo(int id, String nombre, String descripcion, double precio, String material, int
stock) {
            this.id = id;
            this.nombre = nombre;
            this.descripcion = descripcion;
            this.precio = precio;
            this.material = material;
            this.stock = stock;
        }

        // Métodos
        public void crearJoya() {
            System.out.println("Joya creada: " + nombre);
        }

        public void actualizarStock(int cantidad) {
            if (cantidad >= 0) {
                this.stock = cantidad;
                System.out.println("Stock actualizado a: " + stock);
            } else {
                System.out.println("Cantidad inválida para el stock");
            }
        }
    }
}
```

```

public double calcularPrecio() {
    return this.precio;
}

public void listarArticulo() {
    // Lógica para listar la información del artículo
    System.out.println("ID: " + id);
    System.out.println("Nombre: " + nombre);
    System.out.println("Descripción: " + descripcion);
    System.out.println("Precio: " + precio);
    System.out.println("Material: " + material);
    System.out.println("Stock: " + stock);
}

public void eliminarArticulo() {
    System.out.println("Artículo " + nombre + " eliminado.");
}

class Cliente {
    // Atributos
    private int id;
    private String nombre;
    private String email;
    private String telefono;

    // Constructor (opcional)
    public Cliente(int id, String nombre, String email, String telefono) {
        this.id = id;
        this.nombre = nombre;
        this.email = email;
        this.telefono = telefono;
    }

    // Métodos
    public void registrarCliente() {
    }
    public void actualizarInformacion() {
    }
    public void verCatalogo() {
    }
    public boolean realizarPedidos() {
        return true; // Retorna true o false según el éxito del pedido
    }
    public List<Pedido> verHistorialPedidos() {
        List<Pedido> historialPedidos = obtenerHistorialPedidos();
        return historialPedidos;
    }
}

```

```

    }
    // Método auxiliar (opcional, depende de cómo se maneje el historial de pedidos)
    private List<Pedido> obtenerHistorialPedidos() {
        return null; // Reemplazar con la lógica adecuada
    }
}

public class Catalogo {
    // Atributos
    private List<Articulo> articulos;
    private String nombre;
    private Date fechaActualizacion;

    // Constructor
    public Catalogo(String nombre) {
        this.articulos = new ArrayList<>();
        this.nombre = nombre;
        this.fechaActualizacion = new Date();
    }

    // Métodos privados
    private void agregarArticulo(Articulo articulo) {
        articulos.add(articulo);
        actualizarFecha();
    }

    private boolean eliminarArticulo(Articulo articulo) {
        boolean eliminado = articulos.remove(articulo);
        if (eliminado) {
            actualizarFecha();
        }
        return eliminado;
    }

    private void actualizarFecha() {
        this.fechaActualizacion = new Date();
    }

    private boolean actualizarPrecio(Articulo articulo, double nuevoPrecio) {
        for (Articulo art : articulos) {
            if (art.equals(articulo)) {
                art.setPrecio(nuevoPrecio);
                actualizarFecha();
                return true;
            }
        }
        return false;
    }

    // Métodos públicos
    public Articulo buscarArticulo(String nombreArticulo) {
        for (Articulo art : articulos) {

```

```

        if (art.getNombre().equalsIgnoreCase(nombreArticulo)) {
            return art;
        }
    }
    return null;
}

public File obtenerCatalogoPDF() {
    // Implementación para generar un PDF del catálogo.
    // Retornar un archivo PDF generado
    return new File("catalogo.pdf");
}

public List<Articulo> mostrarNovedades() {
    List<Articulo> novedades = new ArrayList<>();
    Date haceUnMes = new Date(System.currentTimeMillis() - 30L * 24 * 60 * 60 * 1000);
    for (Articulo art : articulos) {
        if (art.getFechaIngreso().after(haceUnMes)) {
            novedades.add(art);
        }
    }
    return novedades;
}

}

private Date getFechaIngreso() {
    return null;
}

private String getNombre() {
    return null;
}

private void setPrecio(double nuevoPrecio) {
}

}

```

```

private static class Pedido {
    // Atributos
    private int id;
    private Date fecha;
    private String estado;
    private double total;

    // Constructor
    public Pedido(int id, Date fecha) {
        this.id = id;
        this.fecha = fecha;
        this.estado = "Pendiente"; // Estado inicial por defecto
        this.total = 0.0;
    }
}

```

```

// Métodos públicos
public void crearPedido() {
    this.fecha = new Date();
    this.estado = "Creado";
    // Calcular el total al crear el pedido
    this.total = calcularTotal();
}
public void actualizarEstado(String nuevoEstado) {
    this.estado = nuevoEstado;
}
public double calcularTotal() {
    return this.total; // Devolver el total calculado
}
public int getId() {
    return id;
}
public Date getFecha() {
    return fecha;
}
public String getEstado() {
    return estado;
}
public double getTotal() {
    return total;
}
}

private static class Inventario {
    // Atributos privados
    private int id;
    private String nombre;
    private int cantidad;
    private int umbralMinimo;
    // Constructor
    public Inventario(int id, String nombre, int cantidad, int umbralMinimo) {
        this.id = id;
        this.nombre = nombre;
        this.cantidad = cantidad;
        this.umbralMinimo = umbralMinimo;
    }
    // Método para actualizar la cantidad
    public void actualizarCantidad(int nuevaCantidad) {
        this.cantidad = nuevaCantidad;
    }
    // Método para verificar el stock
    public boolean verificarStock() {

```



```

        return this.cantidad >=
        this.umbralMinimo;  }

    public int getId() {

        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public int getCantidad() {
        return cantidad;
    }
    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }
    public int getUmbralMinimo() {
        return umbralMinimo;
    }
    public void setUmbralMinimo(int umbralMinimo) {
        this.umbralMinimo = umbralMinimo;
    }
}

private static class Proveedor {

    // Atributos privados
    private int id;
    private String nombre;
    private String contacto;

    // Constructor
    public Proveedor(int id, String nombre, String contacto) {
        this.id = id;
        this.nombre = nombre;
        this.contacto = contacto;
    }

    // Método para realizar un pedido
    public void realizarPedido() {

```

```

        System.out.println("Pedido realizado por el proveedor: " + nombre);
    }
    // Método para actualizar la información del proveedor
    public void actualizarInformacion(String nuevoNombre, String nuevoContacto) {
        this.nombre = nuevoNombre;
        this.contacto = nuevoContacto;
        System.out.println("Información del proveedor actualizada.");
    }

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getContacto() {
        return contacto;
    }
    public void setContacto(String contacto) {
        this.contacto = contacto;
    }
}

```

EN LA SIGUIENTE CAPTURA SE EVIDENCIA LA SALIDA DEL ARCHIVO CON EXTENSION .CLASS COMO RESULTADO DE LA CORRECTA EJECUCION DEL CODIGO FUENTE(MAIN) – ELABORADO EN IDE INTELLIJ.

