

Universidad Mariano Gálvez de Guatemala.

Ingeniero Luis Alfredo Alvarado

Programación II



Cinthia Yadira Robles Sotoj

7690-16-13986

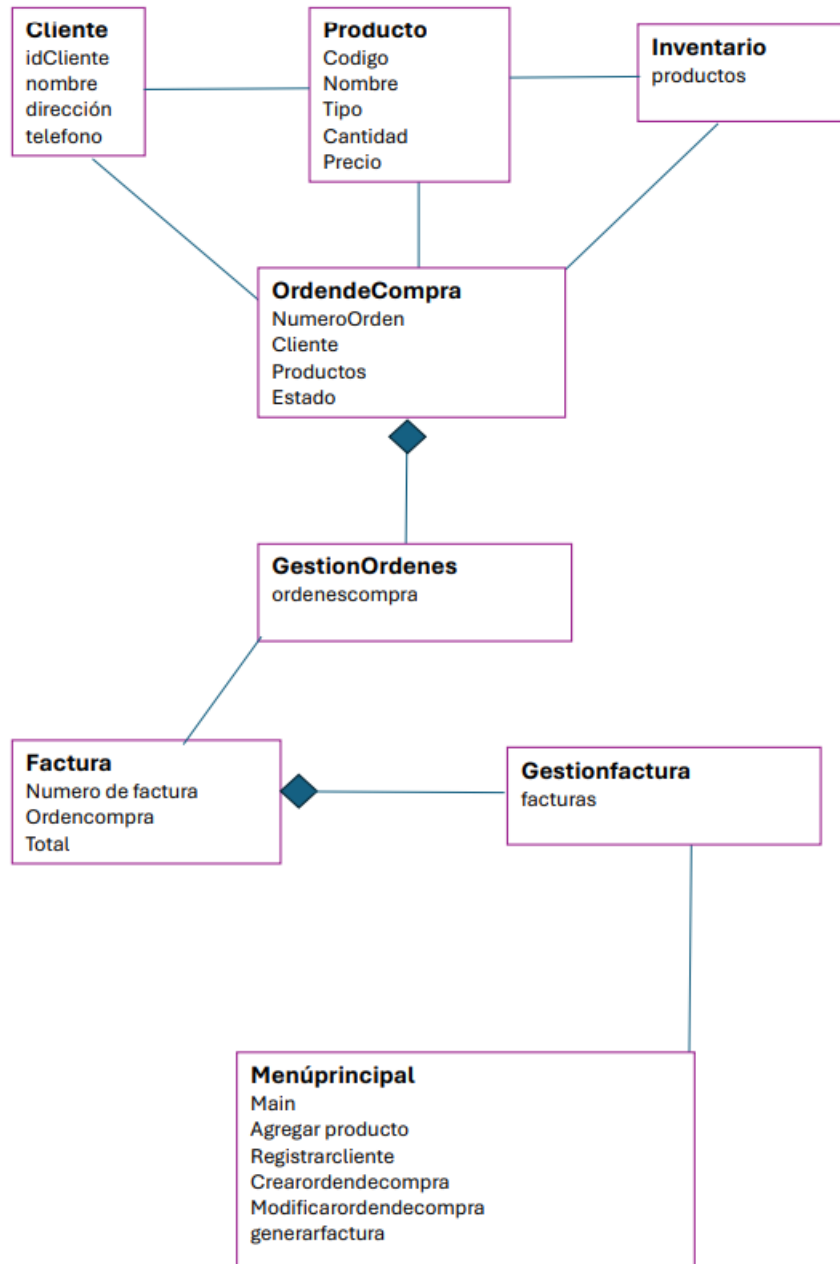
Boca del Monte

Introducción:

El presente proyecto consiste en el desarrollo de un programa utilizando el lenguaje de programación Java bajo el paradigma de la **programación orientada a objetos (POO)**. Este programa permite llevar un control de los productos en stock, sus existencias y la información de los productos en tiempo real.

Se realizó con el principio de responsabilidad única. Los productos se gestionan como objetos que contienen atributos como nombre, cantidad disponible, y categoría, lo que permite realizar operaciones como agregar nuevos productos, actualizar existencias y eliminar productos. También se utilizaron .json para poder almacenar información y llevar un control de productos.

UML



Primera clase es Cliente.java, aquí están el nit, el nombre la dirección el celular del cliente para que puedan ser agregadas en nuestro Cliente.json.

```
Modificado ▶ src/main/java ▶ org.example ▶ Cliente ▶
1 package org.example;
2
3 import java.util.List;
4
5 public class Cliente {
6     private String nitCliente;
7     private String nombreCliente;
8     private String direccion;
9     private String celular;
10
11     private List<Pedido> pedidosLista;
12
13     public List<Pedido> getPedidosLista() {
14         return pedidosLista;
15     }
16
17     public void setPedidosLista(List<Pedido> pedidosLista) {
18         this.pedidosLista = pedidosLista;
19     }
20
21     public String getNitCliente() {
22         return nitCliente;
23     }
24
25     public void setNitCliente(String nitCliente) {
26         this.nitCliente = nitCliente;
27     }
28
29     public String getNombreCliente() {
30         return nombreCliente;
31     }
32
33     public void setNombreCliente(String nombreCliente) {
34         this.nombreCliente = nombreCliente;
35     }
36
37     public String getDireccion() {
38         return direccion;
39     }
40
41     public void setDireccion(String direccion) {
42         this.direccion = direccion;
43     }
44
45     public String getCelular() {
46         return celular;
47     }
48
49     public void setCelular(String celular) {
50         this.celular = celular;
51     }
52 }
```

En la clase Producto se agregará el nombre del producto, el id y el stock(existencia) y nuestra clase Producto.json se encargará de restar lo solicitado por el cliente

```
package org.example;

public class Producto {
    private Integer productoId;
    private String nombre;
    private Integer stock;

    private String estado;

    public Producto() {
    }

    public Producto(Integer idProducto, String nombre, Integer stock, String estado) {
        this.productoId = idProducto;
        this.nombre = nombre;
        this.stock = stock;
        this.estado = estado;
    }

    public String getEstado() {
        return estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public Integer getProductoId() {
        return productoId;
    }

    public void setProductoId(Integer productoId) {
        this.productoId = productoId;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public Integer getStock() {
        return stock;
    }

    public void setStock(Integer stock) {
        this.stock = stock;
    }
}
```

En la clase Pedido.java, se encuentra el estado la fecha y la orden del producto y en nuestra clase Productos.json se vera reflejado si el producto esta en existencia o ya se encuentra agotado.

```
package org.example;

import java.util.ArrayList;

public class Pedido {
    private Integer pedidoId;
    private Date fechaOrden;
    private String estado;

    private List<Producto> productos;

    public List<Producto> getProductos() {
        return productos;
    }

    public void agregarProductoAOrden(Producto producto) {
        if (productos == null) {
            productos = new ArrayList<>();
        }
        productos.add(producto);
    }

    public void modificarEstado(String estado) {
        this.estado = estado;
    }

    public Integer getPedidoId() {
        return pedidoId;
    }

    public void setPedidoId(Integer pedidoId) {
        this.pedidoId = pedidoId;
    }

    public Date getFechaOrden() {
        return fechaOrden;
    }

    public void setFechaOrden(Date fechaOrden) {
        this.fechaOrden = fechaOrden;
    }

    public String getEstado() {
        return estado;
    }
}
```

En las clases **LogicaCliente.java** y **LógicaProducto.java**, vamos a poder ver si ya se encuentran registrados los clientes o los productos y también mostrará si se agregan nuevos clientes o productos.

```
Modificado ▶ src/main/java ▶ org.example ▶ LogicaCliente ▶ cargarClientes() : void
package org.example;

import com.fasterxml.jackson.databind.ObjectMapper;

public class LogicaCliente {
    private static List<Cliente> clientes = new ArrayList<>();
    private static Map<String, String> map = new HashMap<>();

    public static void cargarClientes() throws IOException {
        URL url = new URL("file:src/main/resources/cliente.json");

        ObjectMapper mapper = new ObjectMapper();
        Cliente[] clientesTmp = mapper.readValue(new File(url.getPath()), Cliente[].class);

        for (Cliente cliente : clientesTmp) {
            if (map.get(cliente.getNitCliente()) != null) {
                System.out.println("Cliente con ID duplicado nit: " + cliente.getNitCliente() + " nombre: " + cliente.getNombreCliente());
                continue;
            }
            map.put(cliente.getNitCliente(), cliente.getNombreCliente());
            clientes.add(cliente);
        }

        System.out.println("clientes cargados: " + clientes.size());
    }

    public static void guardarClientes() throws IOException {
        URL url = new URL("file:src/main/resources/cliente.json");

        ObjectMapper mapper = new ObjectMapper();

        ObjectMapper mapper2 = new ObjectMapper();
        mapper2.writerWithDefaultPrettyPrinter().writeValue(new File(url.getPath()), clientes);
        System.out.println("clientes guardados: " + clientes.size());
    }

    public static void agregarCliente(Cliente cliente) {
        if (map.get(cliente.getNitCliente()) != null) {
            clientes.forEach(c -> {
                if (c.getNitCliente().equals(cliente.getNitCliente())) {
                    cliente.getPedidosLista().forEach(pedido -> c.getPedidosLista().add(pedido));
                }
            });
            return;
        }
        System.out.println("Agregando cliente id: " + cliente.getNitCliente() + " nombre: " + cliente.getNombreCliente());
        clientes.add(cliente);
    }

    public static List<Cliente> getClients() {
        return clientes;
    }
}
```

```

package org.example;

import com.fasterxml.jackson.databind.ObjectMapper;

public class LogicaProducto {
    private static List<Producto> productos = new ArrayList<>();
    private static Map<Integer, Integer> map = new HashMap<>();

    public static Producto despacharProducto(int idProducto, int cantidadRequerida) {
        Producto producto = getProducto(idProducto);
        Integer cantidad = producto.getStock();

        if (cantidad < cantidadRequerida) {
            throw new IllegalArgumentException("No hay suficiente cantidad de producto id: " + idProducto +
            );
        }

        producto.setStock(cantidad - cantidadRequerida);

        if (producto.getStock() == 0) {
            producto.setEstado("Agotado");
        }

        return new Producto(producto.getProductoId(), producto.getNombre(), cantidadRequerida, producto.get
    )
}

    public static void salvarProductos() throws IOException {
        URL url = new URL("file:src/main/resources/producto.json");

        ObjectMapper mapper = new ObjectMapper();
        mapper.writerWithDefaultPrettyPrinter().writeValue(new File(url.getPath()), productos);
    }

    public static void adjuntarProductos() throws IOException {
        URL url = new URL("file:src/main/resources/producto.json");

        ObjectMapper mapper = new ObjectMapper();
        Producto[] productosTemp = mapper.readValue(new File(url.getPath()), Producto[].class);

        Integer index = 0;
        for (Producto producto : productosTemp) {
            if (map.get(producto.getProductoId()) != null) {
                System.out.println("Producto con ID duplicado id: " + producto.getProductoId() + " nombre: "
                );
                continue;
            }
            map.put(producto.getProductoId(), index++);
            productos.add(producto);
        }
    }

    public static void agregarProducto(Producto producto) {
        System.out.println("Agregando producto id: " + producto.getProductoId() + " nombre: " + producto.getN
        );
        if (map.get(producto.getProductoId()) != null) {
            throw new IllegalArgumentException("Producto con ID duplicado id: " + producto.getProductoId() +
            );
        }
        productos.add(producto);
    }

    public static List<Producto> getProductos() {
        return productos;
    }

    private static Producto getProducto(int idProducto) {
        Integer productoId = map.get(idProducto);
        if (productoId == null) {
            throw new IllegalArgumentException("Producto no encontrado id: " + idProducto);
        }
        return productos.get(productoId);
    }
}

```


Esta es la clase Main.java, en esta podremos agregar clientes y verificar los productos en existencia y que podrán ser llamados a través de otras clases.

```
import java.util.*;

public class Main {
    public static void main(String[] args) throws Exception {
        LogicaProducto.adjuntarProductos();
        LogicaCliente.cargarClientes();
        Scanner scanner = new Scanner(System.in);
        Integer opcion = 0;
        do {
            System.out.println("1. Seleccione un producto");
            System.out.println("2. Elegir productos");
            System.out.println("3. Ver las compras realizadas");
            System.out.println("4. Agregar un producto");
            System.out.println();
            System.out.println("- Ingrese 0 para salir");

            opcion = scanner.nextInt();
            switch (opcion) {
                case 1: {
                    Cliente cliente = new Cliente();
                    System.out.println("Ingrese su nit");
                    cliente.setNitCliente(scanner.next());
                    System.out.println("Ingrese su nombre");
                    cliente.setNombreCliente(scanner.next());
                    System.out.println("Ingrese su celular");
                    cliente.setCelular(scanner.next());
                    System.out.println("Ingrese su direccion");
                    cliente.setDireccion(scanner.next());
                }
            }
        } while (opcion != 0);
    }
}
```

```

Pedido pedido = new Pedido();
Random random = new Random();
pedido.setPedidoId((random.nextInt(1000)));
pedido.setFechaOrden(new Date());
pedido.modificarEstado("Generado");

while (true) {
    LogicaProducto.getProductos().forEach(producto -> {
        System.out.println("Id: " + producto.getProductoId() + " Nombre: " + producto.getNombre() + " Esta");
    });
    scanner.nextLine();

    System.out.println("Ingrese id del producto que desea comprar");
    Integer idProducto = scanner.nextInt();
    System.out.println("Ingrese la cantidad que desea comprar");
    Integer cantidad = scanner.nextInt();

    Producto producto = LogicaProducto.despacharProducto(idProducto, cantidad);
    pedido.agregarProductoAOrden(producto);

    System.out.println("Desea seguir agregando productos? (si/no)");
    String respuesta = scanner.next();
    if ("no".equalsIgnoreCase(respuesta)) {
        break;
    }
}

cliente.setPedidosLista(List.of(pedido));
System.out.println("Pedido realizado");
LogicaCliente.agregarCliente(cliente);
LogicaCliente.guardarClientes();
LogicaProducto.salvarProductos();
break;

2: {
LogicaProducto.getProductos().forEach(producto -> System.out.println("Id del producto: " + producto.getProcc
System.out.println());
break;

3: {
LogicaCliente.getClientes().forEach(cliente -> {
    System.out.println("Cliente: " + cliente.getNombreCliente());
    System.out.println("Nit: " + cliente.getNitCliente());
    System.out.println("Direccion: " + cliente.getDireccion());

    System.out.println("Pedidos: ");
    cliente.getPedidosLista().forEach(pedido -> {
        System.out.println("    Id Pedido: " + pedido.getPedidoId());
        System.out.println("    Estado: " + pedido.getEstado());
        System.out.println("    Fecha de la venta: " + pedido.getFechaOrden());
        List<Producto> productos = pedido.getProductos();
        if (productos != null) {
            for (Producto producto : productos) {
                System.out.println("        Producto: " + producto.getNombre());
                System.out.println("        Cantidad: " + producto.getStock());
                System.out.println("        Id del producto: " + producto.getProductoId());
                Svstem.out.println();
            }
        }
    });
}

```

```
        });  
    });  
    break;  
}  
case 4: {  
    Producto producto = new Producto();  
    Random random = new Random();  
    producto.setProductoId(random.nextInt(1000));  
    System.out.println("Ingrese el nombre del producto");  
    producto.setNombre(scanner.next());  
    System.out.println("Ingrese el stock del producto");  
    producto.setStock(scanner.nextInt());  
    if (producto.getStock() > 0) {  
        producto.setEstado("Existencia");  
    } else {  
        producto.setEstado("Agotado");  
    }  
  
    LogicaProducto.agregarProducto(producto);  
    LogicaProducto.salvarProductos();  
    break;  
}  
default:  
    System.out.println("Opcion no valida");  
}  
} while (opcion != 0);  
scanner.close();  
}  
}
```

Estas son las clases Cliente.json y Producto.json.

```
1 [ {
2   "productoId" : 1,
3   "nombre" : "Aguacates",
4   "stock" : 100,
5   "estado" : "Existencia"
6 }, {
7   "productoId" : 2,
8   "nombre" : "Frijoles",
9   "stock" : 50,
10  "estado" : "Existencia"
11 }, {
12   "productoId" : 3,
13   "nombre" : "Queso",
14   "stock" : 100,
15   "estado" : "Existencia"
16 }, {
17   "productoId" : 4,
18   "nombre" : "Leche",
19   "stock" : 200,
20   "estado" : "Existencia"
21 }, {
22   "productoId" : 5,
23   "nombre" : "Huevos",
24   "stock" : 150,
25   "estado" : "Existencia"
26 }, {
27   "productoId" : 6,
28   "nombre" : "Pan",
29   "stock" : 300,
30   "estado" : "Existencia"
31 }, {
32   "productoId" : 7,
33   "nombre" : "Azucar",
34   "stock" : 100,
35   "estado" : "Existencia"
36 }, {
37   "productoId" : 8,
38   "nombre" : "Sal",
39   "stock" : 50,
40   "estado" : "Existencia"
41 }, {
42   "productoId" : 9,
43   "nombre" : "Aceite",
44   "stock" : 100,
45   "estado" : "Existencia"
46 }, {
47   "productoId" : 10,
48   "nombre" : "Arroz",
49   "stock" : 200,
50   "estado" : "Existencia"
51 }, {
52   "productoId" : 11,
53   "nombre" : "mesa",
54   "stock" : 76,
55   "estado" : "Existencia"
56 }, {
57   "productoId" : 90,
58   "nombre" : "sillas",
59   "stock" : 88,
60   "estado" : "Existencia"
61 }, {
```

```

Main.java cliente.json x producto.json LogicaClien... Producto.java Pedido.java LogicaProduc... »
1 [ {
2   "nitCliente" : "2323434",
3   "nombreCliente" : "aaaa",
4   "direccion" : "test",
5   "celular" : "56897412",
6   "pedidosLista" : [ {
7     "pedidoId" : 883781400,
8     "fechaOrden" : 1727222683086,
9     "estado" : "Despachado",
10    "productos" : [ {
11      "productoId" : 2,
12      "nombre" : "Frijoles",
13      "stock" : 25,
14      "estado" : "Existencia"
15    } ]
16  } ]
17 }, {
18   "nitCliente" : "90",
19   "nombreCliente" : "Cinthia",
20   "direccion" : "37",
21   "celular" : null,
22   "pedidosLista" : [ {
23     "pedidoId" : 127,
24     "fechaOrden" : 1727752450873,
25     "estado" : "Generado",
26     "productos" : [ {
27       "productoId" : 1,
28       "nombre" : "Aguacates",
29       "stock" : 8,
30       "estado" : "Existencia"
31     } ]
32   } ]

```

```

Main.java cliente.json x producto.json LogicaClien... Producto.java Pedido.java LogicaProduc... »
34   "nitCliente" : "5689",
35   "nombreCliente" : "cinthia",
36   "direccion" : "37calle",
37   "celular" : null,
38   "pedidosLista" : [ {
39     "pedidoId" : 696,
40     "fechaOrden" : 1727752564529,
41     "estado" : "Generado",
42     "productos" : [ {
43       "productoId" : 1,
44       "nombre" : "Aguacates",
45       "stock" : 67,
46       "estado" : "Agotado"
47     } ]
48   } ]
49 }, {
50   "nitCliente" : "5695",
51   "nombreCliente" : "CinthiaRobles",
52   "direccion" : "7ma_avnida",
53   "celular" : "548995",
54   "pedidosLista" : [ {
55     "pedidoId" : 773,
56     "fechaOrden" : 1727754025175,
57     "estado" : "Generado",
58     "productos" : [ {
59       "productoId" : 871,
60       "nombre" : "sillon",
61       "stock" : 7,
62       "estado" : "Existencia"
63     } ]
64   } ]
65 } ]

```

Y por ultimo esta clase pom.xml, en la cual se utilizo Maven con Java 17

```
http://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation with catalog)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>org.example</groupId>
8     <artifactId>Walter</artifactId>
9     <version>1.0-SNAPSHOT</version>
10    <dependencies>
11        <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
12        <dependency>
13            <groupId>com.fasterxml.jackson.core</groupId>
14            <artifactId>jackson-databind</artifactId>
15            <version>2.17.2</version>
16        </dependency>
17    </dependencies>
18    <properties>
19        <maven.compiler.source>17</maven.compiler.source>
20        <maven.compiler.target>17</maven.compiler.target>
21        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
22    </properties>
23
24 </project>
```