UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

CONTROL DE LECTURA 2

Nombre: ESTALYN DANIEL LICUY MECIAS.

ASIGNATURA:

Programación Orientada a Objetos

NRC: 1323

SANGOLQUÍ – ECUADOR

Pagina uno a mano

Vicerrectorado de Docencia
Unidad de Apoyo a la Formación Académica

Actividad de Aprendizaje Contacto Docente N.° 2 Primer Parcial

Tema de la Actividad:

Creación de Objetos y UML **Tipo**

de Actividad:

Diseño y Modelado

Descripción de la Actividad:

Diseñe 5 objetos diferentes con su correspondiente diagrama UML, asegurándose de mostrar las relaciones entre ellos.

1: Producto: Representa un producto en un sistema de inventario
- Atributos: nombre, precio, cantidad.
- metodos: actualizar cantidad(), calcular descuento()

2: Cliente: Representa al cliente que compra
- nombre, correo, dirección
- registrar Compra(), actualizar Datos()

3: Pedido: Representa un pedido echo por el cliente
- número de pedido, fecha, total
- calcular total(), confirmar Pedido()
- se relaciona con el cliente

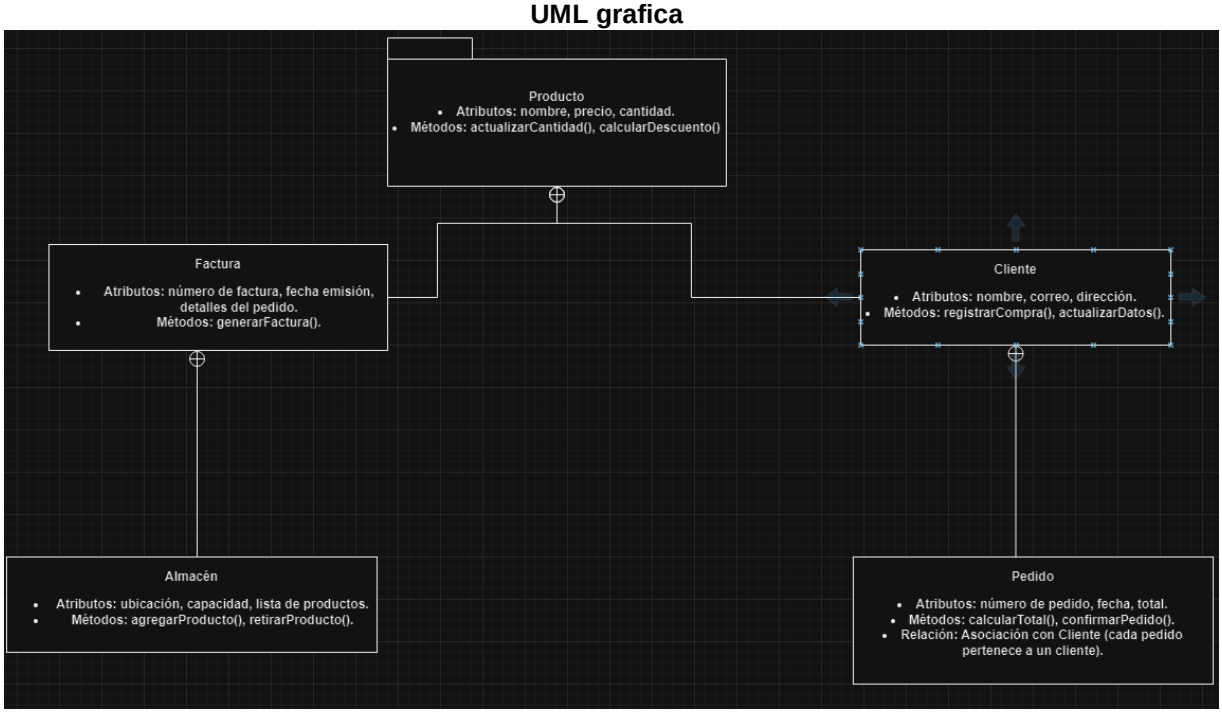
4: Factura; Documentación
- número de Factura, fecha, detalles
- generar factura()
- se relaciona con el pedido

5: Almacén: administra los productos
- capacidad, lista de productos.
- agregar Producto(), retirar Producto()
- Relación con el producto

Tema(s) n.°

Tema(s) n.°

UML grafica



Resumen de la resolución

La selección de los cinco objetos se realizó con base en un sistema típico de inventario y gestión de pedidos, considerando que este tipo de sistema es común y aplicable en múltiples contextos como tiendas, bibliotecas o almacenes. Los objetos elegidos fueron:

- **Producto:** Representa los bienes en inventario.
- **Cliente:** Modela a los usuarios que realizan pedidos.
- **Pedido:** Centraliza la información de las compras realizadas.
- **Factura:** Documenta oficialmente los pedidos efectuados.
- **Almacén:** Administra el stock de productos disponibles.

Estos objetos fueron seleccionados por su relevancia dentro de un sistema de gestión integral y por las relaciones naturales que existen entre ellos en escenarios reales.

Creación de los Diagramas UML

Los diagramas UML fueron diseñados siguiendo un enfoque sistemático:

1. **Definición de las Clases:** Se identificaron los atributos y métodos relevantes para cada clase, asegurando que reflejen sus propiedades y comportamientos en un sistema real.
2. **Establecimiento de Relaciones:** Se analizaron las dependencias entre las clases para definir relaciones como:
    - **Asociación:** Relación directa entre objetos independientes (ej. Pedido y Cliente).
    - **Agregación:** Relación "parte-todo" donde los componentes pueden existir por separado (ej. Factura y Pedido).

- **Composición:** Relación más fuerte donde los componentes no tienen sentido sin el objeto principal (ej. Almacén y Producto).

3. **Uso de Herramientas UML:** Se empleó software de modelado como Lucidchart o StarUML para generar el diagrama, asegurando claridad en la representación visual de las clases y sus relaciones.

Relaciones Entre los Objetos

1. **Cliente y Pedido:** Un cliente puede realizar múltiples pedidos, lo que se representa con una relación de **asociación.**
2. **Pedido y Producto:** Cada pedido contiene una lista de productos, lo que se refleja como una relación de **agregación.**
3. **Factura y Pedido:** Una factura documenta un pedido específico, representando otra **agregación.**
4. **Almacén y Producto:** El almacén administra los productos, reflejando una relación de **composición** ya que los productos no tienen sentido sin el contexto del almacén.

Este enfoque asegura que el diseño sea modular, escalable y fácil de entender, facilitando su implementación en cualquier lenguaje de programación orientado a objetos.

Codigo

Clase producto

```java
public class Producto {
    private String nombre;
    private double precio;
    private int cantidad;

    public Producto(String nombre, double precio, int cantidad) {
        this.nombre = nombre;
        this.precio = precio;
        this.cantidad = cantidad;
    }

    public void actualizarCantidad(int nuevaCantidad) {
        this.cantidad = nuevaCantidad;
    }

    public double calcularDescuento(double porcentaje) {
        return precio * (1 - porcentaje / 100);
    }

    // Getters y Setters
    public String getNombre() {
        return nombre;
    }

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

    public double getPrecio() {
        return precio;
    }

    public void setPrecio(double precio) {
        this.precio = precio;
    }

    public int getCantidad() {
        return cantidad;
    }

    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }
}
```

Clase Cliente

```java
import java.util.ArrayList;

public class Cliente {
    private String nombre;
    private String correo;
    private String direccion;
    private ArrayList<Pedido> pedidos;

    public Cliente(String nombre, String correo, String direccion) {
        this.nombre = nombre;
        this.correo = correo;
        this.direccion = direccion;
        this.pedidos = new ArrayList<>();
    }

    public void registrarCompra(Pedido pedido) {
        this.pedidos.add(pedido);
    }

    public void actualizarDatos(String nuevoCorreo, String nuevaDireccion) {
        this.correo = nuevoCorreo;
        this.direccion = nuevaDireccion;
    }

    // Getters y Setters
    public String getNombre() {
        return nombre;
    }

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

    public String getCorreo() {
        return correo;
    }

    public void setCorreo(String correo) {
        this.correo = correo;
    }
    public String getDireccion() {
        return direccion;
    }
    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }
}
```

Clase Pedido

```
import java.util.ArrayList;

public class Pedido {
    private int numeroPedido;
    private String fecha;
    private double total;
    private ArrayList<Producto> productos;

    public Pedido(int numeroPedido, String fecha) {
        this.numeroPedido = numeroPedido;
        this.fecha = fecha;
        this.productos = new ArrayList<>();
    }

    public void agregarProducto(Producto producto) {
        this.productos.add(producto);
    }

    public void calcularTotal() {
        this.total = productos.stream()
                              .mapToDouble(p -> p.getPrecio() *
p.getCantidad())
                              .sum();
    }

    public void confirmarPedido() {
        System.out.println("Pedido #" + numeroPedido + " confirmado.");
    }

    // Getters y Setters
    public int getNumeroPedido() {
        return numeroPedido;
    }

    public void setNumeroPedido(int numeroPedido) {
        this.numeroPedido = numeroPedido;
    }

    public String getFecha() {
        return fecha;
    }

    public void setFecha(String fecha) {
        this.fecha = fecha;
    }

    public double getTotal() {
```

```
        return total;
    }

    public void setTotal(double total) {
        this.total = total;
    }
}
```

Clase Factura

```
public class Factura {
    private int numeroFactura;
    private String fechaEmision;
    private Pedido detallesPedido;

    public Factura(int numeroFactura, String fechaEmision, Pedido
detallesPedido) {
        this.numeroFactura = numeroFactura;
        this.fechaEmision = fechaEmision;
        this.detallesPedido = detallesPedido;
    }

    public void generarFactura() {
        System.out.println("Factura #" + numeroFactura + " generada el " +
fechaEmision);
        System.out.println("Detalles del Pedido: " +
detallesPedido.getNumeroPedido());
    }

    // Getters y Setters
    public int getNumeroFactura() {
        return numeroFactura;
    }

    public void setNumeroFactura(int numeroFactura) {
        this.numeroFactura = numeroFactura;
    }

    public String getFechaEmision() {
        return fechaEmision;
    }

    public void setFechaEmision(String fechaEmision) {
        this.fechaEmision = fechaEmision;
    }

    public Pedido getDetallesPedido() {
        return detallesPedido;
    }
```

```
    public void setDetallesPedido(Pedido detallesPedido) {
        this.detallesPedido = detallesPedido;
    }
}
```

Clase Almacén

```
import java.util.ArrayList;

public class Almacen {
    private String ubicacion;
    private int capacidad;
    private ArrayList<Producto> listaDeProductos;

    public Almacen(String ubicacion, int capacidad) {
        this.ubicacion = ubicacion;
        this.capacidad = capacidad;
        this.listaDeProductos = new ArrayList<>();
    }

    public void agregarProducto(Producto producto) {
        if (listaDeProductos.size() < capacidad) {
            listaDeProductos.add(producto);
        } else {
            System.out.println("Almacén lleno. No se puede agregar más
productos.");
        }
    }

    public void retirarProducto(Producto producto) {
        listaDeProductos.remove(producto);
    }

    // Getters y Setters
    public String getUbicacion() {
        return ubicacion;
    }

    public void setUbicacion(String ubicacion) {
        this.ubicacion = ubicacion;
    }

    public int getCapacidad() {
        return capacidad;
    }
    public void setCapacidad(int capacidad) {
        this.capacidad = capacidad;
    }
}
```

Informe

Introducción

El presente informe documenta la actividad de diseño y modelado de objetos y la creación de un diagrama UML que representa la relación entre cinco clases en un sistema orientado a objetos. Además, se incluye el desarrollo del código para cada clase, seguido de un análisis de las relaciones establecidas en el diagrama UML y su implementación en código.

Objetivos

- Diseñar cinco objetos relacionados entre sí y sus respectivas relaciones en un diagrama UML.
- Implementar el código para cada clase en el lenguaje de programación Java.
- Analizar las relaciones entre las clases y documentar la lógica del sistema.
- Presentar un informe que incluya el diagrama UML, código, y un resumen del diseño.

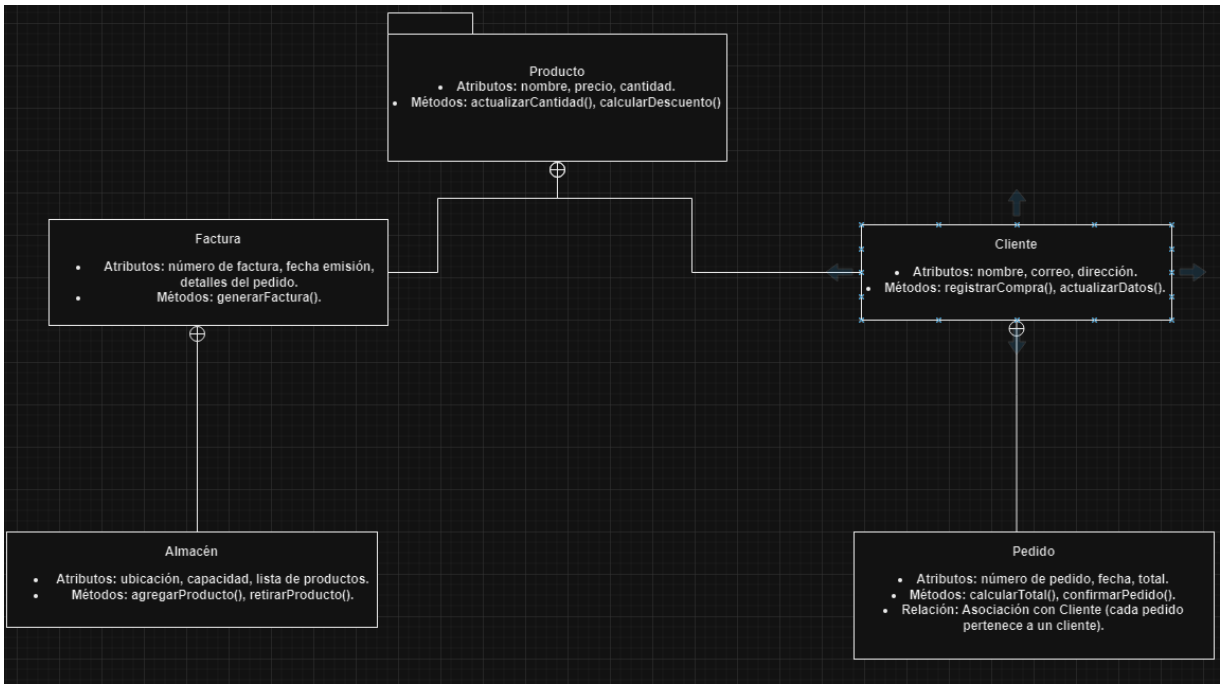
Marco Teórico

El diseño orientado a objetos (OO) es una metodología ampliamente utilizada en el desarrollo de software que organiza un sistema en términos de "clases" y "objetos." Una clase define la estructura y comportamiento compartido de un conjunto de objetos, mientras que un objeto es una instancia concreta de una clase. Para representar estas relaciones y estructuras, se emplea el Lenguaje Unificado de Modelado (UML), una herramienta estándar para modelar sistemas complejos de software.

En UML, las relaciones como asociación, agregación y composición permiten describir las interacciones y dependencias entre clases. La **asociación** denota un vínculo general entre dos clases, mientras que la **agregación** indica una relación "parte-todo" más débil. Por otro lado, la **composición** establece una relación más fuerte, en la que los componentes no pueden existir de forma independiente del todo. Estas relaciones facilitan la planificación de arquitecturas de software robustas y escalables, alineando los requisitos funcionales con los principios de encapsulación, cohesión y reutilización de código.

Diagrama UML

A continuación, se presenta el diagrama UML diseñado para los objetos:



Descripción de las Clases y Relaciones

1. Clase Producto: Representa los productos en inventario con atributos como nombre, precio y cantidad, y métodos para actualizar la cantidad y calcular descuentos.
    - Relación: Composición con Almacén.
2. Clase Cliente: Representa a los clientes del sistema con atributos personales y una lista de pedidos realizados.
    - Relación: Asociación con Pedido.
3. Clase Pedido: Define los pedidos realizados por clientes, con atributos como número de pedido, fecha, total, y una lista de productos.
    - Relación: Asociación con Cliente, Agregación con Factura.
4. Clase Factura: Documenta los pedidos realizados, asociando un número de factura y los detalles del pedido.
    - Relación: Agregación con Pedido.
5. Clase Almacén: Administra los productos almacenados, con métodos para agregar y retirar productos.
    - Relación: Composición con Producto.

Código Fuente

```java
public class Producto {
    private String nombre;
    private double precio;
    private int cantidad;

    public Producto(String nombre, double precio, int cantidad) {
        this.nombre = nombre;
        this.precio = precio;
        this.cantidad = cantidad;
    }

    public void actualizarCantidad(int nuevaCantidad) {
        this.cantidad = nuevaCantidad;
    }

    public double calcularDescuento(double porcentaje) {
        return precio * (1 - porcentaje / 100);
    }

    // Getters y Setters
    public String getNombre() {
        return nombre;
    }

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

    public double getPrecio() {
        return precio;
    }

    public void setPrecio(double precio) {
        this.precio = precio;
    }

    public int getCantidad() {
        return cantidad;
    }

    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }
}
```

```java
import java.util.ArrayList;

public class Cliente {
    private String nombre;
    private String correo;
    private String direccion;
    private ArrayList<Pedido> pedidos;

    public Cliente(String nombre, String correo, String direccion) {
        this.nombre = nombre;
        this.correo = correo;
        this.direccion = direccion;
        this.pedidos = new ArrayList<>();
    }

    public void registrarCompra(Pedido pedido) {
        this.pedidos.add(pedido);
    }

    public void actualizarDatos(String nuevoCorreo, String nuevaDireccion) {
        this.correo = nuevoCorreo;
        this.direccion = nuevaDireccion;
    }

    // Getters y Setters
    public String getNombre() {
        return nombre;
    }

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

    public String getCorreo() {
        return correo;
    }

    public void setCorreo(String correo) {
        this.correo = correo;
    }

    public String getDireccion() {
        return direccion;
    }

    public void setDireccion(String direccion) {
```

```java
public class Pedido {
    public Pedido(int numeroPedido, String fecha) {
    }

    public void agregarProducto(Producto producto) {
        this.productos.add(producto);
    }

    public void calcularTotal() {
        this.total = productos.stream()
                              .mapToDouble(p -> p.getPrecio() * p.getCantidad())
                              .sum();
    }

    public void confirmarPedido() {
        System.out.println("Pedido #" + numeroPedido + " confirmado.");
    }

    // Getters y Setters
    public int getNumeroPedido() {
        return numeroPedido;
    }

    public void setNumeroPedido(int numeroPedido) {
        this.numeroPedido = numeroPedido;
    }

    public String getFecha() {
        return fecha;
    }

    public void setFecha(String fecha) {
        this.fecha = fecha;
    }

    public double getTotal() {
        return total;
    }

    public void setTotal(double total) {
        this.total = total;
    }
}
```

```java
public class Factura {
    private int numeroFactura;
    private String fechaEmision;
    private Pedido detallesPedido;

    public Factura(int numeroFactura, String fechaEmision, Pedido detallesPedido) {
        this.numeroFactura = numeroFactura;
        this.fechaEmision = fechaEmision;
        this.detallesPedido = detallesPedido;
    }

    public void generarFactura() {
        System.out.println("Factura #" + numeroFactura + " generada el " + fechaEmision);
        System.out.println("Detalles del Pedido: " + detallesPedido.getNumeroPedido());
    }

    // Getters y Setters
    public int getNumeroFactura() {
        return numeroFactura;
    }

    public void setNumeroFactura(int numeroFactura) {
        this.numeroFactura = numeroFactura;
    }

    public String getFechaEmision() {
        return fechaEmision;
    }

    public void setFechaEmision(String fechaEmision) {
        this.fechaEmision = fechaEmision;
    }

    public Pedido getDetallesPedido() {
        return detallesPedido;
    }

    public void setDetallesPedido(Pedido detallesPedido) {
        this.detallesPedido = detallesPedido;
    }
}
```

```java
public class Almacen {
    private String ubicacion;
    private int capacidad;
    private ArrayList<Producto> listaDeProductos;

    public Almacen(String ubicacion, int capacidad) {
        this.ubicacion = ubicacion;
        this.capacidad = capacidad;
        this.listaDeProductos = new ArrayList<>();
    }

    public void agregarProducto(Producto producto) {
        if (listaDeProductos.size() < capacidad) {
            listaDeProductos.add(producto);
        } else {
            System.out.println("Almacén lleno. No se puede agregar más productos.");
        }
    }

    public void retirarProducto(Producto producto) {
        listaDeProductos.remove(producto);
    }

    // Getters y Setters
    public String getUbicacion() {
        return ubicacion;
    }

    public void setUbicacion(String ubicacion) {
        this.ubicacion = ubicacion;
    }

    public int getCapacidad() {
        return capacidad;
    }

    public void setCapacidad(int capacidad) {
        this.capacidad = capacidad;
    }
}
```

Conclusiones

1. La implementación del diagrama UML y las clases correspondientes permitió consolidar conocimientos sobre el diseño orientado a objetos, destacando la importancia de las relaciones como asociación, agregación y composición en la planificación de sistemas.
2. Este ejercicio práctico fortaleció habilidades en la modelación de sistemas mediante UML y la codificación en Java, asegurando una correcta implementación de las interacciones entre los objetos. Además, se evidenció cómo el diseño previo facilita la codificación y reduce errores en el desarrollo del software.

Bibliografía

Diagrama UML: Qué es, cómo hacerlo y ejemplos | Miro. (s. f.). https://miro.com/.

https://miro.com/es/diagrama/que-es-diagrama-uml/

Qué es una clase en Java y cómo crearla (con ejemplos). (2023, 16 mayo). *Blog.Hubspot*.

https://blog.hubspot.es/website/que-es-clase-en-java