

Universidad de las Fuerzas Armadas “ESPE”

1323- Programación Orientada a Objetos

Henry Bolivar Borja Milan

Control de lectura 1: Diseñe 5 objetos diferentes con su correspondiente diagrama UML, asegurándose de mostrar las relaciones entre ellos.

Ing. Luis Enrique Jaramillo Montaña

22 de noviembre del 2024

Objetivo

Comprender el Lenguaje Unificado de Modelado (UML), enfocándose en los seis tipos principales de relaciones que se pueden representar en los diagramas de clases.

Digramas UML

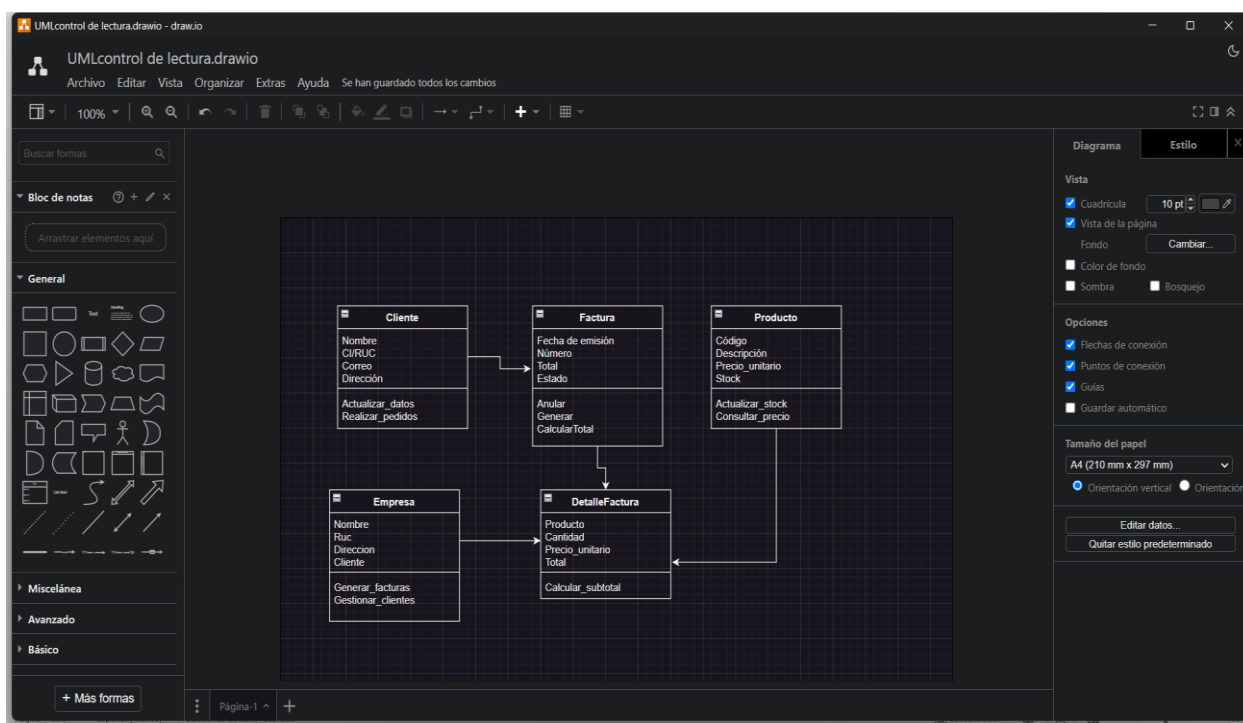
El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que permite la especificación, visualización, construcción y documentación de los componentes de un sistema de software. UML proporciona un conjunto estándar de notaciones para diagramas que representan tanto la estructura estática como el comportamiento dinámico de los sistemas. Su objetivo principal es estandarizar el modelado de sistemas de software, proporcionando un vocabulario y reglas para combinar y construir representaciones, modelos conceptuales y físicos del sistema. (Visual Pradigm, 2022)

Los seis tipos principales de relaciones son:

1. Herencia (Generalización): Describe una relación jerárquica entre una clase general (superclase) y una clase más específica (subclase). La subclase hereda atributos y métodos de la superclase, permitiendo la reutilización y extensión del comportamiento. Se representa con una línea sólida con una flecha vacía apuntando hacia la superclase.
2. Realización (Implementación): Indica que una clase concreta implementa una interfaz. La clase concreta proporciona la funcionalidad declarada por la interfaz. Se representa con una línea discontinua con una flecha vacía apuntando hacia la interfaz.
3. Composición: Es una forma fuerte de agregación que denota una relación "todo/parte" donde las partes no pueden existir independientemente del todo. Si el todo es destruido,

las partes también lo son. Se representa con una línea sólida con un rombo negro en el extremo de la clase que representa el todo.

4. Agregación: Similar a la composición, pero las partes pueden existir independientemente del todo. Representa una relación "todo/parte" más débil. Se representa con una línea sólida con un rombo blanco en el extremo de la clase que representa el todo.
5. Asociación: Indica una relación estructural entre clases, donde una clase está conectada a otra. Puede tener multiplicidad para mostrar cuántas instancias de una clase están relacionadas con una instancia de otra clase. Se representa con una línea sólida entre las clases involucradas.
6. Dependencia: Muestra que una clase depende de otra porque utiliza sus funciones. Es una relación más débil que la asociación y generalmente implica que un cambio en la clase independiente puede afectar a la clase dependiente. Se representa con una línea discontinua con una flecha apuntando hacia la clase de la que depende.



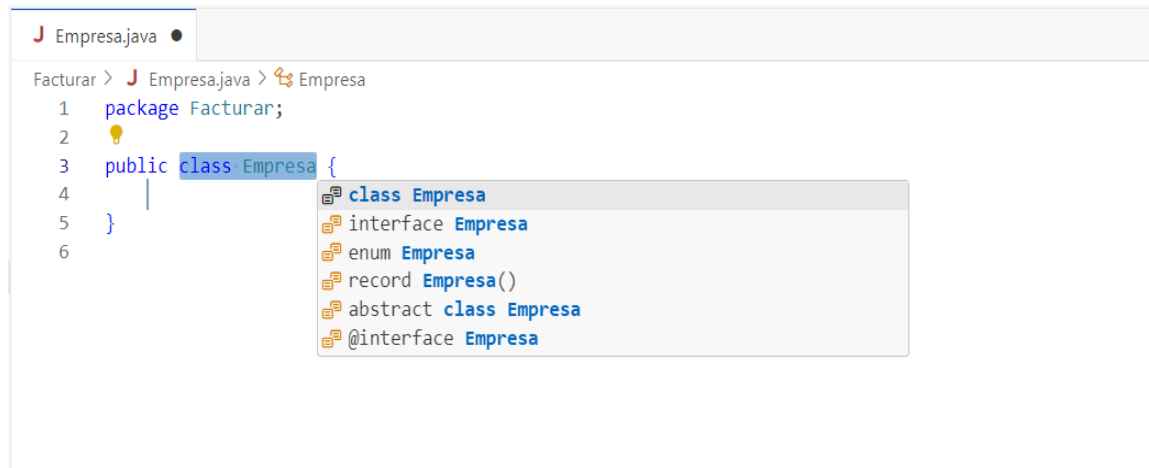
Comprender y aplicar correctamente estos tipos de relaciones en los diagramas de clases de UML es fundamental para modelar con precisión la estructura y las interacciones dentro de un sistema de software. (Izquierdo et al., s.f.)

El objetivo de esta actividad fue diseñar un sistema básico de facturación, incluyendo clases como Cliente, Factura, Producto, DetalleFactura y Empresa. El diseño se realizó en UML, mostrando las relaciones entre las clases, y se implementó el código en Java.

Se identificaron las clases principales y sus relaciones. Cada clase fue diseñada para representar una entidad del sistema. Luego, se creó un diagrama UML para representar las relaciones entre las clases. Finalmente, el diseño se implementó en código Java, asegurando que el código y el diagrama coincidieran.

El sistema permite gestionar clientes, productos y facturas de manera eficiente.

Creacion de las clases



```
Facturar > J Empresa.java > Empresa
1 package Facturar;
2
3 public class Empresa {
4
5 }
6
```

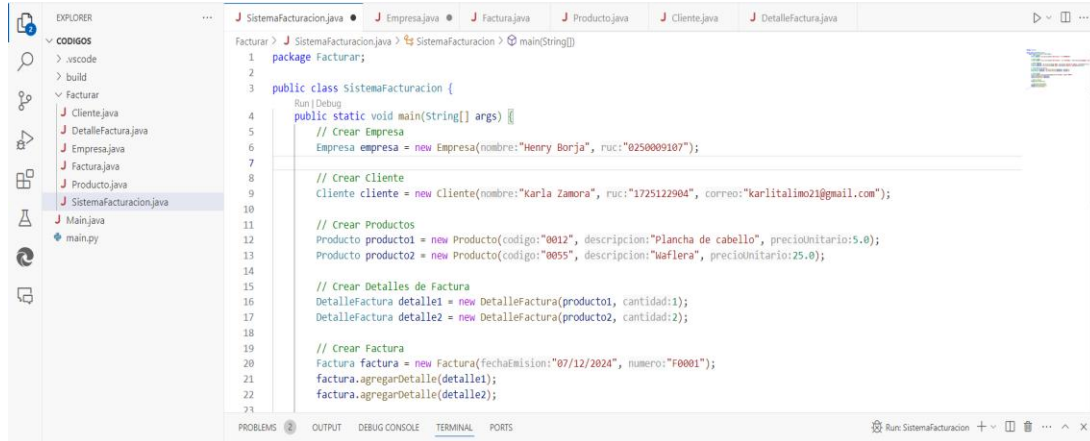
The screenshot shows an IDE window titled 'Empresa.java'. The editor displays the following code:

```
1 package Facturar;
2
3 public class Empresa {
4
5 }
6
```

A dropdown menu is open next to the 'class Empresa' line, showing the following options:

- class Empresa
- interface Empresa
- enum Empresa
- record Empresa()
- abstract class Empresa
- @interface Empresa

Pruebas de funcionamiento



```

1 package Facturar;
2
3 public class SistemaFacturacion {
4     // Crear Empresa
5     Empresa empresa = new Empresa(nombre:"Henry Borja", ruc:"0250009107");
6
7     // Crear Cliente
8     Cliente cliente = new Cliente(nombre:"Karla Zamora", ruc:"1725122904", correo:"karlitalimo21@gmail.com");
9
10    // Crear Productos
11    Producto producto1 = new Producto(codigo:"0012", descripcion:"Plancha de cabello", precioUnitario:5.0);
12    Producto producto2 = new Producto(codigo:"0055", descripcion:"Waflera", precioUnitario:25.0);
13
14    // Crear Detalles de Factura
15    DetalleFactura detalle1 = new DetalleFactura(producto1, cantidad:1);
16    DetalleFactura detalle2 = new DetalleFactura(producto2, cantidad:2);
17
18    // Crear Factura
19    Factura factura = new Factura(fechaEmision:"07/12/2024", numero:"F0001");
20    factura.agregarDetalle(detalle1);
21    factura.agregarDetalle(detalle2);
22
23

```

Resultados obtenidos:

```

Microsoft Windows [Versión 10.0.26100.2454]
(c) Microsoft Corporation. Todos los derechos reservados.

D:\CODIGOS> cmd /c ""C:\Program Files\Java\jdk-23\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\boliv\AppData\Roaming\Code\User\workspaceStorage\3096cf669f8af993892ca035c289c0de\redhat.java\jdt_ws\CODIGOS_f04b4c41\bin Facturar.SistemaFacturacion "
Factura N°: F0001
Fecha de Emisión: 07/12/2024
- Producto: Plancha de cabello | Cantidad: 1 | Subtotal: $5.0
- Producto: Waflera | Cantidad: 2 | Subtotal: $50.0
Total: $55.0
La empresa Henry Borja (RUC: 0250009107) está gestionando clientes.
Cliente Karla Zamora (RUC: 1725122904, Correo: karlitalimo21@gmail.com) está realizando un pedido.

```

Referencias

Izquierdo, C. E., Márquez, A. E., Plata, I. L., Batista, M. B., & Vega, J. M. (s.f.). *Tema 1*.

Introducción a UML.

https://campusvirtual.ull.es/ocw/pluginfile.php/14338/mod_resource/content/1/1.%20Introducci%C3%B3n.pdf

Visual Pradigm. (9 de febrero de 2022). *¿Cuáles Son Los Seis Tipos De Relaciones En Los Diagramas De Clases UML?* <https://blog.visual-paradigm.com/es/what-are-the-six-types-of-relationships-in-uml-class-diagrams/>