UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

SEGUNDO SEMESTRE

CARRERAS TECNICAS

DEPARTAMENTO DE CIENCIAS EXACTAS

PRIMER PARCIAL

AUTOR:

NAYELI SCARLETH LOACHAMIN TIPAN

NRC:

1323

DOCENTE:

LUIS ENRIQUE JARAMILLO MONTAÑO

SANGOLQUI – ECUADOR

1. Introducción

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual desarrollado para proporcionar una notación común y detallada para la arquitectura, diseño e implementación de sistemas de software complejos, tanto en su estructura como en su comportamiento. Su utilidad no se limita al desarrollo de software ya que también puede aplicarse en áreas como la gestión de procesos en la fabricación. UML se asemeja a los planos utilizados en otras disciplinas y se compone de diversos tipos de diagramas que describen los límites, la estructura y el comportamiento de un sistema, así como los objetos que lo componen. Aunque UML no es un lenguaje de programación, existen herramientas que permiten generar código a partir de los diagramas. Este lenguaje está estrechamente vinculado al análisis y diseño.

2. Objetivos

2.1 Objetivo General

- Aprender sus conocimientos referentes al UML, Microsoft Viso y algunos conceptos; con el fin de brindar mejores profesionistas y actualizar los avances del Software.

2.2 Objetivos específicos

- Estudiar el lenguaje de modelado UML.
- Desarrollar por completo el diseño de un proyecto de software con el fin de comprender todo el proceso.
- Mostrar como UML crea un protocolo de comunicación estándar entre los desarrolladores

3. Marco Teórico

A continuación, realizare una breve explicación

Ejercicios

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

OBJETO LIBRO

Atributo:

Título: El título del libro

Autor: El autor del libro

Isbn: El código ISBN que identifica de forma única al libro

Método:

Presentar (): Método para presentar el libro

Devolver (): Método para devolver el libro

Relación:

Se relaciona con un usuario mediante composición ya que un usuario tiene una lista de libros prestados

Diagrama UML

LIBRO
Título: String Autor: String Isbn: String
+Prestar(): void +devolver(): void

Código

Resolución

Representa un libro con atributos como título, autor, isbn y métodos prestar () y devolver ().

OBJETO USUARIO

Atributo:

Nombre: Nombre del usuario

Libros prestados: Una lista de libros que el usuario ha tomado prestado

Método

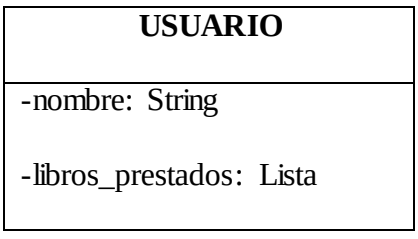
Tomar_prestado (libro): Método para tomar prestado un libro

Devolver_libro (libro): Método para devolver un libro

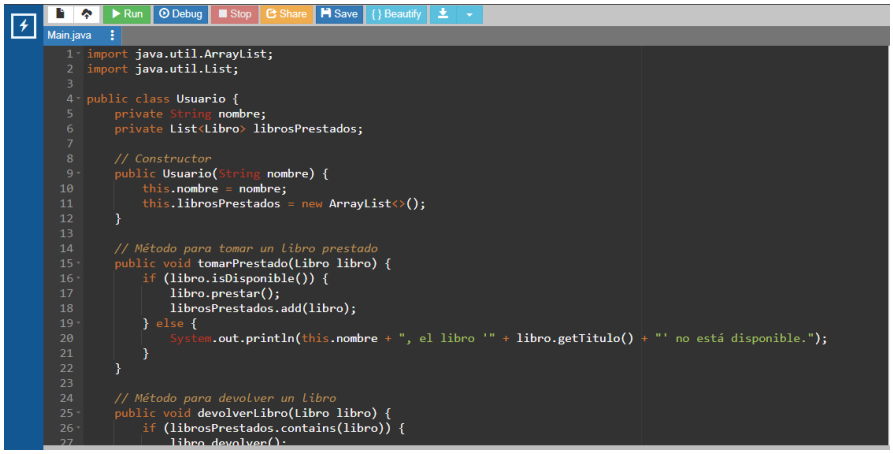
Relación

El objeto Usuario tiene una relación de composición con libro, ya que un usuario puede tener varios libros prestados.

Diagrama de UML



Código:




```java
import java.util.ArrayList;
import java.util.List;

public class Usuario {
    private String nombre;
    private List<Libro> librosPrestados;

    // Constructor
    public Usuario(String nombre) {
        this.nombre = nombre;
        this.librosPrestados = new ArrayList<>();
    }

    // Método para tomar un libro prestado
    public void tomarPrestado(Libro libro) {
        if (libro.isDisponible()) {
            libro.prestar();
            librosPrestados.add(libro);
        } else {
            System.out.println(this.nombre + ", el libro '" + libro.getTitulo() + "' no está disponible.");
        }
    }

    // Método para devolver un libro
    public void devolverLibro(Libro libro) {
        if (librosPrestados.contains(libro)) {
            libro.devolver();
```

Resolución

Representa a los usuarios, quienes tienen una lista de libros prestados. Se relacionan como libro por composición.

OBJETO BIBLIOTECA

Atributos:

Catalogo_libros: Una lista de objetos libro que la biblioteca tiene disponible.

Usuarios: Una lista de objetos usuario registrado.

Método:

Agregar_libro (libro): Agrega un libro al catálogo.

Registrar_usuarios (usuario): Registra un usuario en la biblioteca

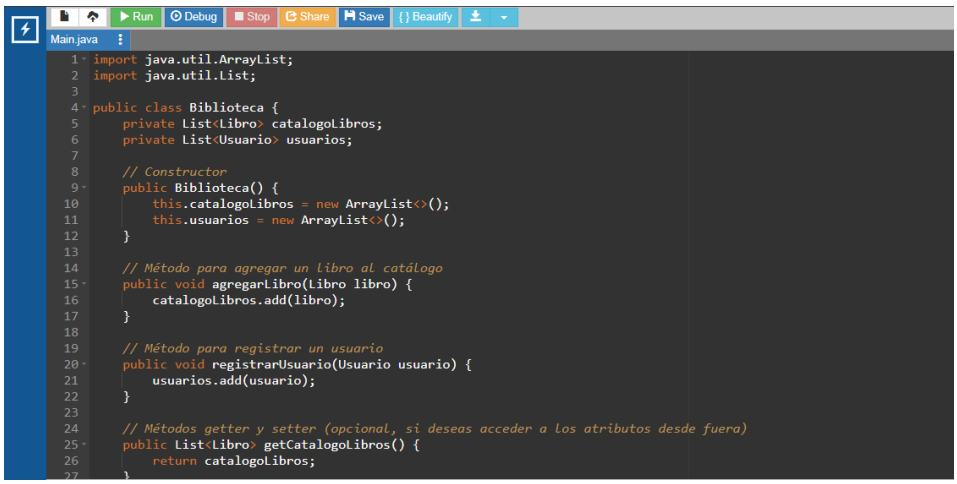
Relación:

La biblioteca tiene una relación de asociación con libro, ya que gestiona un catálogo libro.

Diagrama de UML

BIBLIOTECA
-catalogo_libros: List -usuarios: List
+Agregar_libro(Libros) +Registrar_usuario(usuarios)

Código:



Resolución:

Gestiona un catálogo de libros y una lista de usuarios. Se relaciona con libro por asociación.

OBJETO EMPLEADO

Atributo:

Nombre: nombre del empleado

Cargo: Cargo del empleado en la biblioteca

Método:

Gestionar_prestamo (usuario, libro): Método para gestionar un préstamo de libro.

Gestionar_devolucion (usuario, libro): Método para gestionar la devolución de un libro.

Relación

El empleado tiene una relación de asociación tanto con usuario como con libro.

Diagrama de UML

EMPLEADO
-nombre: String -cargo: String
+gestionar_prestamo(usuario, libro) +gestionar_devolucion(usuario, libro)

Código:

Resolución:

Se encarga de gestionar los préstamos y devoluciones que tiene relaciones de asociación con usuario y libro.

OBJETO DE TRANSACCIÓN

Atributo:

Usuario: usuario relacionado con la transacción

Libro: libro relacionado con la transacción

Fecha: fecha de la transacción

Tipo: tipo de transacción (préstamo o devolución)

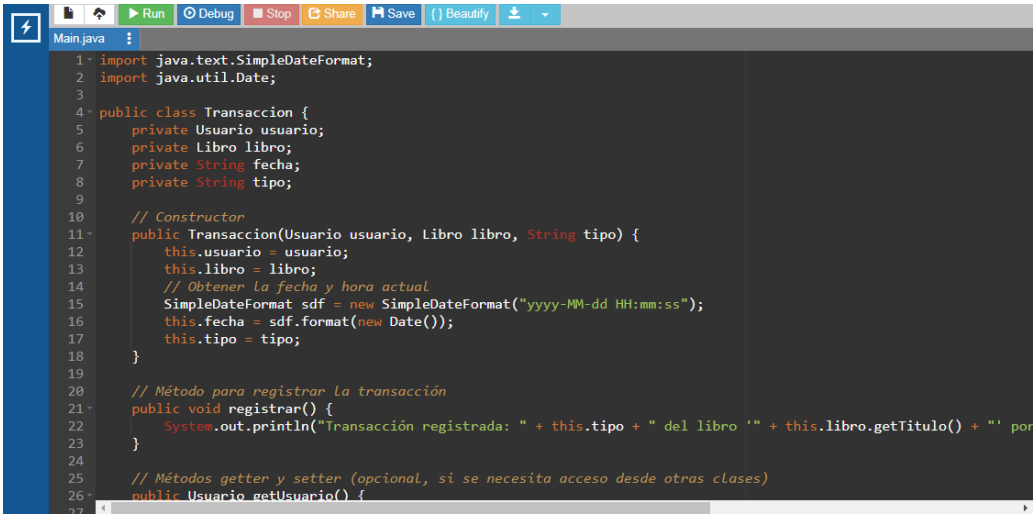
Método:

Registrar (): registra la transacción de un préstamo o devolución

Relación: Se relaciona con usuario y libro mediante agregación ya que la transacción involucra tanto a un usuario como a un libro pero no depende de ello.

Diagrama UML

Código:



Resolución:

Registra las transacciones de préstamo y devolución que se relaciona con usuario y libro de agregación.

4. Conclusiones

- Estandarización del diseño: UML proporciona un lenguaje visual estandarizado que facilita la comunicación entre los miembros del equipo, reduciendo malentendidos durante las fases de diseño y desarrollo.
- Versatilidad y adaptabilidad: UML puede aplicarse a una amplia gama de proyectos, desde sistemas complejos hasta aplicaciones pequeñas, gracias a su capacidad de modelar diferentes aspectos del sistema, comportamiento, estructura y procesos.
- Facilita la documentación: Al utilizar UML, se genera una documentación clara y estructurada que sirve como referencia para desarrolladores actuales y futuros.

5. Recomendaciones

- Adapta los diagramas a las necesidades del proyecto: No es necesario usar todos los tipos de diagramas UML. Selecciona los que mejor representen el sistema que estás diseñando (por ejemplo, diagramas de clases para estructuras o diagramas de actividad para flujos de trabajo).
- Capacita al equipo en UML: Asegúrate de que todos los miembros del equipo comprendan los fundamentos de UML para que puedan interpretar y contribuir a los diagramas de manera efectiva.
- Utiliza herramientas de software adecuadas: Usa software especializado (como Lucidchart, Enterprise Architect o Visual Paradigm) que facilite la creación y mantenimiento de diagramas UML.

Bibliografía

Qué es el lenguaje unificado de modelado (UML). (s. f.). Lucidchart. https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=un%20diagrama%20UML-,%C2%BFQu%C3%A9%20es%20UML%3F,en%20estructura%20como%20en%20comportamiento.%20(s.f.).%20Obtenido%20de%20https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=un%20diagrama%20UML-,%C2%BFQu%C3%A9%20es%20UML%3F,en%20estructura%20como%20en%20comportamiento.

Royercuno. (s. f.). *OBJETIVO GENERAL.docx*. Scribd. https://es.scribd.com/document/313814081/OBJETIVO-GENERAL-docx

(N.d.). Edu.Pe. Retrieved December 10, 2024, from https://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/basic/mendoza_nj/Conclu.pdf

(N.d.-b). Udlap.Mx. Retrieved December 10, 2024, from http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rea_c_ji/capitulo6.pdf