



UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
PRIMER PARCIAL

TEMA:

“SISTEMA DE GESTIÓN DE PROYECTOS DE TIC’S”

INTEGRANTES:

Claudio Dahik Villalba Paladines

Brittany Naomi Loor Oña

Milton Joseph Yuxi Chicaiza

Luis Andrés Pujos Tanicuchi

Daniel Alexis Quilca Andrade

MATERIA:

Programación Orienta A Objetos

PARALELO:

NRC-1323

SANGOLQUI – ECUADOR

1. MARCO TEÓRICO

1.1 SISTEMA DE GESTIÓN DE PROYECTOS

Un sistema de gestión de proyectos puede entenderse de dos maneras. Por un lado, cuando se refiere a la manera en que los gestores llevan a cabo sus tareas. Por otro lado, si se considera como una herramienta utilizada en la gestión de proyectos, se define como un conjunto de recursos o herramientas diseñadas para apoyar a los gestores en el cumplimiento efectivo de sus funciones en cada fase del proyecto (SafetyCulture, 2024).

1.2 ¿QUÉ ES UN UML?

UML es una integración de diferentes notaciones orientadas a objetos, como el diseño orientado a objetos, la técnica de modelado de objetos y la ingeniería de software orientada a objetos. Aprovecha las ventajas de estos tres enfoques para ofrecer una metodología más coherente y fácil de utilizar (Lucidchart, s.f.).

1.3 UNA CLASE

Las clases son el componente central del diagrama y representan, como su nombre lo sugiere, una clase dentro del paradigma de la programación orientada a objetos. Una clase describe un conjunto de objetos que tienen en común características, propiedades y un significado específico. (DiagramasUML.com, 2024).

1.4 LOS ATRIBUTOS

Los atributos son las características que describen a una clase u otro tipo de elemento y representan las propiedades o datos internos que definen su información esencial. Estos atributos permiten identificar y diferenciar a los objetos de una clase, ya que almacenan valores específicos que determinan su estado o comportamiento dentro del sistema (Guía de Usuario de Enterprise Architect 7.0, 2007).

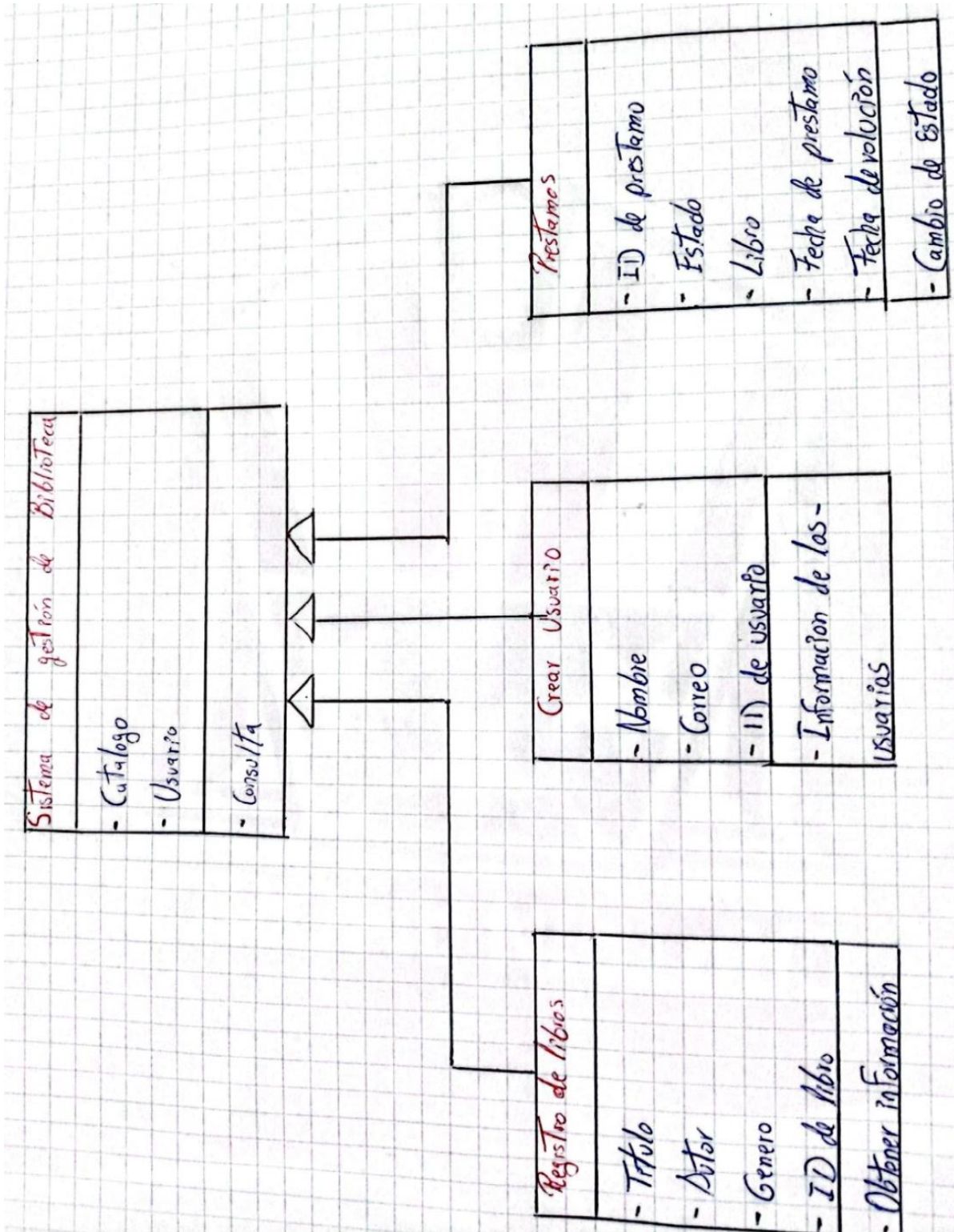
1.5 UN MÉTODO

Un método es una forma clara de organizar el pensamiento y las acciones de una persona. Además, el método guía al usuario indicando qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo, a diferencia del lenguaje de modelado, que no incluye estas instrucciones. Los métodos incluyen modelos, los cuales se utilizan para describir algo y comunicar los resultados obtenidos al aplicar el método (Román Zamitiz, s.f.).

2. RESUMEN – SISTEMA DE GESTION DE BIBLIOTECA

El proyecto del sistema de Gestión de Biblioteca fue desarrollado de manera organizada y eficiente; gracias a la colaboración y coordinación del equipo. Antes de la implementación del código, se diseñó un diagrama UML que permitió establecer la estructura del sistema, las clases son: Registro de libros, Crear usuarios, Prestamos, Sistema de gestión de biblioteca. Este proceso inicial fue fundamental para garantizar que el código estuviera bien estructurado y cumpliera con todo lo necesario. Finalmente, el código se realizó con la participación de todo el equipo, donde cada miembro se le designó desarrollar una de las clases mencionadas, el equipo realizó una verificación conjunta del funcionamiento del programa, asegurándose de que cada uno se ejecutara correctamente. La buena comunicación, coordinación y planificación permitieron que esta actividad experimental se completara sin inconvenientes, obteniendo un buen resultado.

3. UML- SISTEMA DE GESTION DE BIBLIOTECA



4. CÓDIGO EXPLICADO

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

// Primera parte: creacion de Registros de Libro, Identificadores de ID
// Registro de Libro Biblioteca
class Libro {
    private String titulo;
    private String autor;
    private String genero;
    private int idLibro;

    public Libro(String titulo, String autor, String genero, int idLibro) {
        this.titulo = titulo;
        this.autor = autor;
        this.genero = genero;
        this.idLibro = idLibro;
    }

    public String obtenerInformacion () {
        return ID "+idLibro+", Titulo: "+titulo+", Autor: "+autor+",
        genero: "+genero;
    }

    public int getIdLibro() {
        return idLibro;
    }
}

// segunda parte: creacion para registro de Usuario para la biblioteca.
// Identificacion para correo institucional.
// Registro de Usuarios Biblioteca.
class Usuario {
    private String nombre;
    private String correo;
    private int idUsuario;

    public Usuario(String nombre, String correo, int idUsuario) {
        this.nombre = nombre;
        this.correo = correo;
    }
}
```



```

        this.IdUsuario = idUsuario;
    }
    public String informacionDeLosUsuarios() {
        return "ID: " + usuario + ", Nombre: " + nombre + ", Correo: " + correo;
    }
    public String informacion e int getIdUsuario() {
        return IdUsuario;
    }
}

```

II Tercera Parte: Prestamo dentro del usuario, prestacion de libros con fechas

1. Verificaciones de Prestamo

```

class Prestamo {
    private int idPrestamo;
    private Libro libro;
    private Usuario usuario;
    private String fechaPrestamo;
    private String fechaDevolucion;
    private String estado;

    public Prestamo(int idPrestamo, Libro libro, Usuario usuario, String
        fechaPrestamo) {
        this.idPrestamo = idPrestamo;
        this.libro = libro;
        this.usuario = usuario;
        this.fechaPrestamo = fechaPrestamo;
        this.estado = "Activo"; // estado inicial del Prestamo
    }
}

```

II Método Prestamo Biblioteca

```

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

```

```

public String obtenerInformacion() {
    return "ID prestamo: " + idPrestamo + ", Libro: " + libro.obtenerInformacion() +
        ", Usuario: " + usuario.informacionDeLosUsuarios() +
        ", Fecha Prestamo: " + fechaPrestamo +
        ", Fecha Devolucion: " + (fechaDevolucion != null ? fechaDevolucion + "Pendiente" :

```



```
    "Estado: " + estado;
```

```
    }  
    public void setFechaDevolucion(String fechaDevolucion) {  
        this.fechaDevolucion = fechaDevolucion;  
    }  
}
```

// Cuarta Parte: Proceso de Adquisición de libros dentro de la Biblioteca

```
class SistemaDeGestionDeBiblioteca {
```

```
    private List<Libros> catalogo;
```

```
    private List<Usuario> usuarios;
```

```
    private List<Prestamo> prestamos;
```

```
    private int idPrestamoCounter = 1;
```

```
    public SistemaDeGestionDeBiblioteca() {
```

```
        this.catalogo = new ArrayList<>();
```

```
        this.usuarios = new ArrayList<>();
```

```
        this.prestamos = new ArrayList<>();
```

```
    }  
    // Metodos para gestionar libros
```

```
    public void agregarLibro(String titulo, String autor, String genero, int idLibro) {
```

```
        catalogo.add(new Libro(titulo, autor, genero, idLibro));
```

```
    }
```

```
    public void mostrarCatalogo() {
```

```
        System.out.println("Catalogo de Libros:");
```

```
        for (Libro libro : catalogo) {
```

```
            System.out.println(libro.obtenerInformacion());
```

```
        }
```

```
    }
```

```
    // Metodos para gestionar usuarios.
```

```
    public void registrarUsuario(String nombre, String correo, int idUsuario) {
```

```
        usuarios.add(new Usuario(nombre, correo, idUsuario));
```

```
    }
```

```
    public void mostrarUsuarios() {
```

```
        System.out.println("Usuarios Registrados:");
```

```
        for (Usuario usuario : usuarios) {
```



```
System.out.println(usuario.informacionDeLosUsuarios());
```

```
}
```

```
}
```

// Metodos para gestionar Prestamos.

```
public void registrarPrestamo (int idLibro, int idUsuario, String fechaPrestamo) {
```

```
    Libro libro = catalogo.stream().filter(L -> L.getIdLibro() == idLibro).
```

```
        findFirst().orElse(null);
```

```
    Usuario usuario = usuarios.stream().filter(L -> L.getIdUsuario()
```

```
        == idUsuario).findFirst().orElse(null);
```

```
    if (libro == null || usuario == null) {
```

```
        System.out.println("Libro o usuario no encontrado.");
```

```
        return;
```

```
}
```

```
    Prestamo prestamo = new Prestamo(idPrestamo(counter++), libro,
```

```
        usuario, fechaPrestamo);
```

```
    prestamos.add(prestamo);
```

```
    System.out.println("Prestamo registrado exitosamente.");
```

```
}
```

```
public void mostrarPrestamos() {
```

```
    System.out.println("Prestamos:");
```

```
    for (Prestamo prestamo : prestamos) {
```

```
        System.out.println(prestamo.obtenerInformacion());
```

```
    }
```

```
}
```

```
}
```

// Ultima Parte: Registro Final dentro de Main, representacion dentro de la galeria Final

```
public class Main {
```

```
    public static void main (String[] args) {
```

```
        SistemaDeGestionDeBiblioteca sistema = new SistemaDeGestionBiblioteca();
```

// Registrar libros Resultados

```
sistema.agregarLibro("El principito", "Antoine de Saint-Exupéry",
```

```
    "Infantil", 1);
```

```
sistema.agregarLibro("Don Quijote de la Mancha", "Miguel
```

```
    de Cervante", "Clasico", 2);
```



```

// registrar usuarios Resultados
sistema.registrarUsuario ("Miton", "myyuxi@espe.edu.ec", 1);
sistema.registrarUsuario ("Dahik", "cdvillalba2@espe.edu.ec", 2);
sistema.registrarUsuario ("Britanny", "bnloae@espe.edu.ec", 3);
sistema.registrarUsuario ("Daniel", "daguilca@espe.edu.ec", 4);
// Mostrar catalogo y usuario Resultados
system.out.println("\nCatalogo:");
sistema.mostrarCatalogo();
system.out.println("\nUsuarios:");
sistema.mostrarUsuarios();
// Registrar prestamos Resultados
sistema.registrarPrestamo (1, 1, "2024-12-08");
sistema.registrarPrestamo (2, 2, "2024-12-09");
// Mostrar prestamos Resultados
system.out.println("\nPrestamos:");
sistema.mostrarPrestamos();
}

```

5. CONCLUSIONES

Para concluir, el proyecto del Sistema de Gestión de Biblioteca fue un éxito rotundo gracias a la adecuada organización y coordinación del equipo, y al uso de un diseño UML que permitió estructurar el sistema de manera clara y coherente. Cada miembro del equipo asumió la responsabilidad de una clase específica, lo que facilitó el desarrollo y la integración del código. Además, el sistema cumple con los objetivos establecidos, logrando gestionar de manera eficiente los registros de libros, usuarios y préstamos. Las pruebas realizadas demostraron que todas las funcionalidades operan correctamente, asegurando que el sistema sea viable y útil para su propósito. En resumen, este proyecto fue una experiencia valiosa que evidenció la importancia de la planificación y la colaboración en el desarrollo de software.

6. REFERENCIAS

DiagramasUML.com. (15 de Noviembre de 2024). *DiagramasUML*. Diagrama de clases. Teoria y ejemplo:

<https://diagramasuml.com/diagrama-de-clases/>

Guía de Usuario de Enterprise Architect 7.0. (2007). Atributos:

<http://www.sparxsystems.com.ar/download/ayuda/index.html?attributes.htm>

Lucidchart. (s.f.). *Lucidchart*. Qué es el lenguaje unificado de modelado (UML):

[https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-](https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=UML%20es%20una%20combinaci%C3%B3n%20de,sea%20m%C3%A1s%20sencilla%20de%20usar.)

[uml#:~:text=UML%20es%20una%20combinaci%C3%B3n%20de,sea%20m%C3%A1s%20sencilla%20de%20usar.](https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#:~:text=UML%20es%20una%20combinaci%C3%B3n%20de,sea%20m%C3%A1s%20sencilla%20de%20usar.)

Román Zamitiz, C. (s.f.). *EL LENGUAJE UNIFICADO DE MODELADO (UML)*. Facultad de Ingeniería,

UNAM: <http://profesores.fi-b.unam.mx/carlos/aydoo/toc.html>

SafetyCulture. (15 de Enero de 2024). *SafetyCulture*. Sisitema de gestión de proyectos:

<https://safetyculture.com/es/temas/gestion-de-proyectos/>