



**UNIVERSIDAD DE LAS FUERZAS ARMADAS
(ESPE)**



PROGRAMACIÓN ORIENTADA A OBJETOS

**Actividad de Aprendizaje Contacto Docente N.º 2
Primer Parcial**

AUTOR:

- Jefferson Jaya

PARALELO:

NRC-1322

DOCENTE:

Ing. Luis Jaramillo

PERÍODO:

Octubre 2024 – Marzo 2025

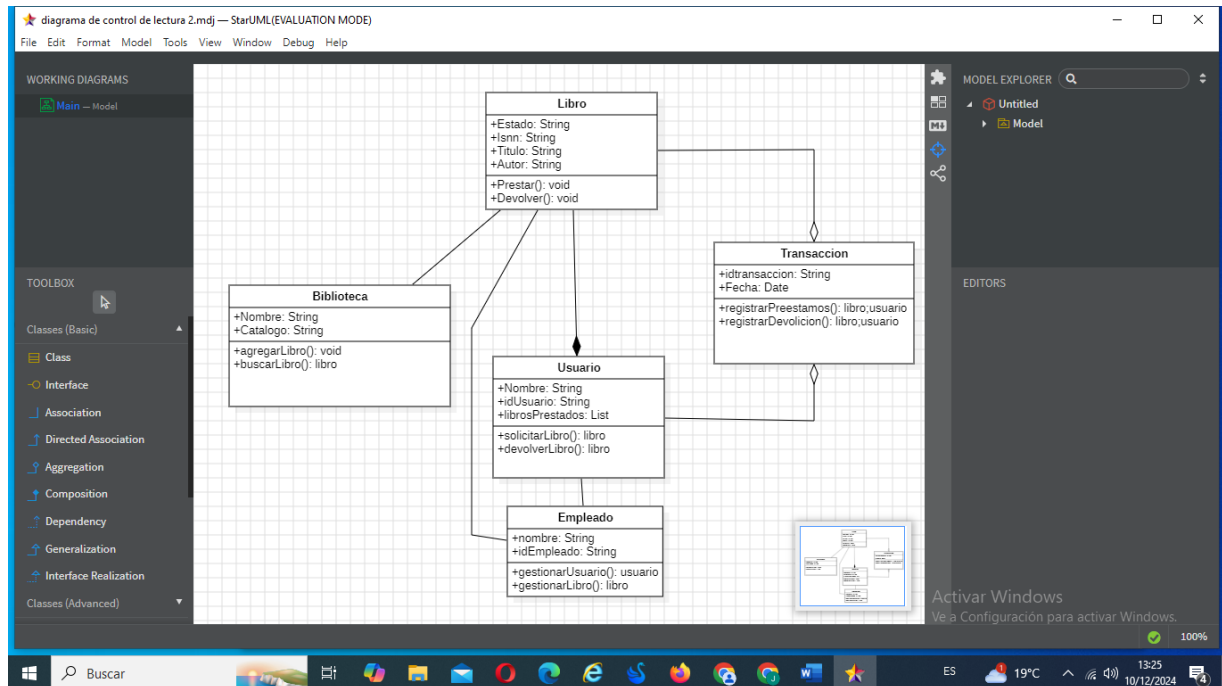
SANGOLQUI – ECUADOR

1. TEMA

Creación de objetos y UML

2. Diagrama UML

Creamos este diagrama utilizando la aplicación StartUML.



Explicación de cada clase

Clase libro

- Objetivo: En el sistema de biblioteca, simboliza un libro único.
- Atributos:

título: El nombre del libro

autor: Escritor del libro

isbn: Código único para identificar el libro

estado: Indica si el libro está disponible o prestado.

- Métodos:

prestar (): Altera la condición del libro a prestado

devolver (): Recuerda el estado actual del libro a la disponibilidad.

Clase Usuario:

- **Objetivo:** Simboliza a un integrante de la biblioteca capaz de pedir libros prestados.
- **Atributos:**

nombre: El nombre íntegro del usuario

idUsuario: Una única identificación del usuario en el sistema.

LibrosPrestados: Registro de los libros que actualmente tiene el usuario en su poder

- **Métodos:**

solicitarLibro(libro): Facilita al usuario la petición de un libro

devolverLibro(libro): Facilita que el usuario restituya un libro prestado.

Clase Biblioteca:

- **Objetivo:** Administra la totalidad de los libros y las operaciones generales.
- **Atributos:**

nombre: Identificación de la biblioteca

catálogo: Descripción completa de los libros presentes en la biblioteca

- **Métodos:**

agregarLibro(libro): Incorpora un libro reciente al catálogo.

buscarLibro(título): Busca un libro según su título.

Clase Empleado:

- **Objetivo:** Representa al equipo que gestiona la biblioteca.
- **Atributos:**

nombre: Nombre del empleado.

idEmpleado: Identificador exclusivo del trabajador.

- **Métodos:**

gestionarUsuarios(usuario): Gestiona los datos y acciones de los usuarios.

gestionarLibro(libro): Cambia la condición de un libro en el sistema.

Clase Transacción:

- **Objetivo:** Logra y seguimiento de los créditos y devoluciones de libros.
- **Atributos:**

idTransaccion: Un solo identificador de la transacción.

fecha: Fecha en la que sucede la transacción.

- Métodos:

registrarPrestamo(libro, usuario): Registrar cuando se presta un libro

registrarDevolucion (libro, usuario): Registro al devolver un libro.

Implementación en Java:

```

1- public class SistemaBiblioteca{
2-     public static void main(String[] args) {
3         Biblioteca biblioteca=new Biblioteca ("Biblioteca rumiñahui 17011",Calle Prin
           Venezuela");
4
5         Libro libro1=new Libro ("El extranjero","Albert Camus","isbn-001");
6         Libro libro2=new Libro ("La colmena","Camilo Cela", "isbn-002");
7
8
9         Biblioteca.registrarLibro (libro1);
10        Biblioteca.registrarLibro (libro2);
11
12        Usuario usuario=new Usuario("Javier Tapia", "user-001");
13        Usuario empleado=new Empleado("Maria Tobar", "emp001", "Bibliotecaria");
14        Transaccion prestamo=new Transaccion("TRANS-001", libro, usuario, Transaccion
           .TipoTransaccion.Prestamo);
15        Prestamo.registrar Prestamo (biblioteca);
16
17        Empleado.gestionarUsuarios (usuario, biblioteca);
18
19        Transaccion devolucion=new Transaccion("TRANS-002", libro, usuario, Transacci
           on.TipoTransaccion.DEVOLUCION);
20        Devolucion.registrarDevolucion (biblioteca);
21
22    }
23 }
24
25

```

```

1- public class Usuario {
2     public String nombre;
3     public String idUsuario;
4     public List<Libro> listaLibros Prestados;
5
6
7     public Usuario (String nombre, String idUsuario) {
8         this.nombre = nombre;
9         this.idUsuario = idUsuario;
10        this.listaLibros Prestados = new ArrayList<>();
11
12    }
13    public void solicitarPrestamo (Libro libro) {
14        if (libro.verificarDisponibilidad()) {
15            libro.prestar();
16            listaLibrosPrestados.add(libro);
17            System.out.println(nombre + "ha tomado prestado el libro: " + libro.titulo);
18        } else {
19            System.out.println("El libro no está disponible para préstamo.");
20        }
21
22    }
23    public void devolverLibro (Libro libro) {
24        if (listaLibrosPrestados.contains(libro)) {
25            libro.devolver();
26            listaLibrosPrestados.remove(libro);
27            System.out.println(nombre + "ha devuelto el libro: " + libro.titulo);
28        } else {
29            System.out.println("Este libro no fue prestado por " + nombre);
30        }
31    }
32 }
33
34
35

```

```

1 public class Biblioteca {
2     public String nombre;
3     public List<Libro> catalogo;
4
5
6     public Biblioteca (String nombre, String direccion) {
7         this.nombre = nombre;
8         this.direccion = direccion;
9         this.catalogo = new ArrayList<>();
10    }
11
12    public Libro buscarLibro (String titulo) {
13        return catalogo.stream()
14            .filter (libro -> libro.titulo.equalsIgnoreCase (titulo))
15            .findFirst()
16            .orElse (null);
17    }
18
19
20    public void registrarLibro (Libro libro) {
21        if (!catalogo.contains(libro)) {
22            catalogo.add(libro);
23            System.out.println("Libro registrado: " + libro.titulo);
24        } else {
25            System.out.println("El libro ya existe en el catálogo.");
26        }
27    }
28
29    public void eliminarLibro (Libro libro) {
30        if (catalogo.remove(libro)) {
31            System.out.println("Libro eliminado: " + libro.titulo);
32        } else {
33            System.out.println("El libro no se encuentra en el catálogo.");
34        }
35    }
36 }

```

close ad [x]

Activar Windows
Ve a Configuración para activar Windows.

```

1 public class Empleado {
2     public String nombre;
3     public String idEmpleado;
4
5
6     public Empleado (String nombre, String idEmpleado) {
7         this.nombre = nombre;
8         this.idEmpleado = idEmpleado;
9     }
10
11
12    public void gestionarUsuarios (Usuario usuario, Biblioteca biblioteca) {
13        // Ejemplo de gestión: imprimir información del usuario
14        System.out.println("Gestionando usuario: " + usuario.nombre);
15        System.out.println("Libros prestados: usuario.listaLibros Prestados.size());
16    }
17
18    public void actualizarEstadoLibro (Libro libro, Libro. EstadoLibro nuevoEstado) {
19        // Cambiar manualmente el estado de un libro
20        if (nuevoEstado == Libro. EstadoLibro.DISPONIBLE) {
21            libro.devolver();
22        } else {
23            libro.prestar();
24        }
25    }
26 }

```

close ad [x]

Activar Windows
Ve a Configuración para activar Windows.

```

1 public class Transaccion {
2     public String idTransaccion;
3     public LocalDate fecha;
4
5     public enum TipoTransaccion {
6         PRESTAMO, DEVOLUCION
7     }
8
9
10    public Transaccion (String idTransaccion, Libro libro, Usuario usuario, TipoTransaccion tipo) {
11        this.idTransaccion = idTransaccion;
12        this.fecha = LocalDate.now();
13    }
14
15    public void registrarPrestamo (Biblioteca biblioteca) {
16        if (libro.verificarDisponibilidad()) {
17            usuario.solicitarPrestamo (libro);
18            System.out.println("Préstamo registrado: libro.titulo +
19                " " + usuario.nombre +
20                " [" + fecha + "]");
21        }
22    }
23
24    public void registrarDevolucion (Biblioteca biblioteca) {
25        usuario.devolverLibro(libro);
26        System.out.println("Devolución registrada: libro.titulo +
27            " " + usuario.nombre +
28            " [" + fecha + "]");
29    }
30 }
31 }

```

close ad [x]

Activar Windows
Ve a Configuración para activar Windows.