



UNIVERSIDAD DE LAS FUERZAS ARMADAS (ESPE)

SEGUNDO SEMESTRE

CARRERAS TECNICAS

DEPARTAMENTO DE CIENCIAS EXACTAS

PRIMER PARCIAL

AUTORES:

DARWIN STEVEN GUERRERO CABRERA

SHAFFIK SAMIR SAMANDÉ MUJICA

NRC:

1322

DOCENTE:

LUIS ENRIQUE JARAMILLO MONTAÑO

SANGOLQUÍ – ECUADOR

1. Introducción

El UML (Lenguaje de Modelado Unificado) es un lenguaje gráfico utilizado para describir y diseñar sistemas de software. Se compone de varios tipos de diagramas y notaciones estandarizadas que se utilizan para representar la estructura, el comportamiento y la interacción de los componentes de un sistema de software. Es una herramienta útil para la planificación, diseño y comunicación en proyectos de desarrollo de software.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar cinco objetos con atributos, métodos y relaciones claras, representándolos mediante diagramas UML, acompañados de un resumen explicativo y el código correspondiente.

2.2 OBJETIVOS ESPECÍFICO

- Diseñar cinco objetos distintos, definiendo atributos, métodos y relaciones mediante diagramas UML que reflejan asociación, composición o agregación.
- Documentar un resumen explicativo que describa el diseño, las interacciones entre los objetos y las decisiones tomadas en el modelado.
- Desarrollar y entregar el código correspondiente que implemente los objetos diseñados, asegurando su funcionalidad según lo descrito.

3. MARCO TEÓRICO

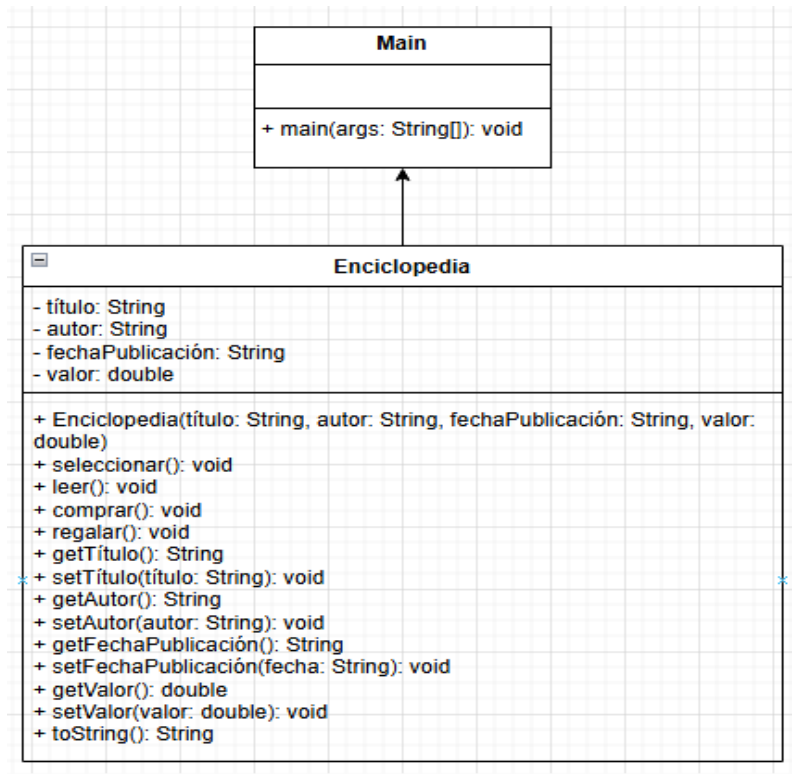
- A continuación, vamos a realizar una breve explicación de la actividad que tenemos que realizar.

ACTIVIDAD

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

Creación de Objetos y UML (Enciclopedia)

- Diagrama UML:



Explicación

Es la clase principal del programa, indicada por su nombre. Tiene un único método:

- `+ main(args: String()): void`

Es el método principal donde se inicia la ejecución del programa. Es público (+) y no devuelve ningún valor (void).

3.1 2. Clase Enciclopedia

Representa un objeto relacionado con enciclopedias, definido con atributos y métodos relativamente.

Atributos:

- - **Título: String:** Título de la enciclopedia.
- - **autor: String:** Autor de la enciclopedia.
- - **fechaPublicación: String:** Fecha en la que se publicó.
- - **valor: double:** Precio o valor asignado a la enciclopedia.

Todos los atributos tienen acceso privado (-), lo que significa que solo pueden ser manipulados mediante los métodos de la clase.

Métodos:

Constructores:

- + **Enciclopedia(título: String, autor: String, fechaPublicación: String, valor: double):**
Constructor público para inicializar un objeto con los valores de sus atributos.

Métodos funcionales:

- + **seleccionar(): void:** Realiza una acción específica sobre la enciclopedia (posiblemente elegir una para su uso o visualización).
- + **leer(): void:** Representa la acción de leer la enciclopedia.
- + **comparar(): void:** Posiblemente compara la enciclopedia con otra según algún criterio.
- + **regalar(): void:** Indica una acción para regalar la enciclopedia.

Métodos de acceso (getters y setters):

Getters (+):

- + **getTítulo(): String**
- + **getAutor(): String**
- + **getFechaPublicación(): String**
- + **getValor(): double**

Setters (+):

- + **setTítulo(título: String): void**
- + **setAutor(autor: String): void**
- + **setFechaPublicación(fecha: String): void**
- + **setValor(valor: double): void**

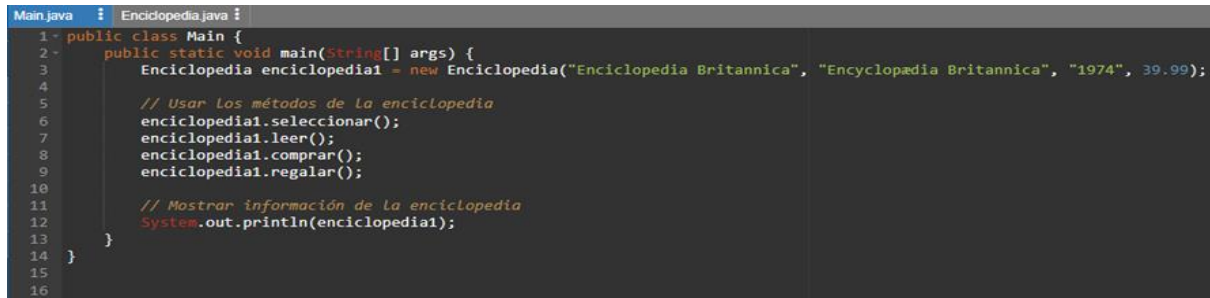
Estos métodos permiten obtener o modificar los valores de los atributos privados de la clase.

Otros métodos:

- + **toString(): String:** Devuelve una representación en texto del objeto Enciclopedia, generalmente incluyendo valores de sus atributos.

CÓDIGO

Main.java



```

1 public class Main {
2     public static void main(String[] args) {
3         Encyclopedia enciclopedia1 = new Encyclopedia("Enciclopedia Britannica", "Encyclopedia Britannica", "1974", 39.99);
4
5         // Usar Los métodos de La enciclopedia
6         enciclopedia1.seleccionar();
7         enciclopedia1.leer();
8         enciclopedia1.comprar();
9         enciclopedia1.regalar();
10
11        // Mostrar información de La enciclopedia
12        System.out.println(encyclopedia1);
13    }
14 }
15
16

```

Descripción

- **public class Main {**

- Declara la clase principal del programa.

- **public static void main(String[] args) {**

- Método principal donde inicia la ejecución del programa.

- **Encyclopedia enciclopedia1 = new Encyclopedia("Encyclopedia Britannica", "Encyclopedia Britannica", "1974", 39.99);**

Crea un objeto enciclopedia1 de la clase Encyclopedia con valores iniciales (título, autor, fecha y precio).

- **encyclopedia1.seleccionar();**

- Llama al método seleccionar() del objeto enciclopedia1.

- **encyclopedia1.leer();**

- Llama al método leer() para simular la lectura de la enciclopedia.

- **encyclopedia1.comprar();**

- Invoca el método comprar() para compararla con otro objeto.

- **encyclopedia1.regalar();**

Ejecuta el método regalar() que representa dar la enciclopedia.

- **System.out.println(encyclopedia1);**

- Imprime en la consola los detalles del objeto enciclopedia1 usando el método toString().

```

Main.java  Enciclopedia.java
1- public class Enciclopedia {
2-     // Atributos
3-     private String titulo;
4-     private String autor;
5-     private String fechaPublicacion;
6-     private double valor;
7-
8-     // Constructor
9-     public Enciclopedia(String titulo, String autor, String fechaPublicacion, double valor) {
10-         this.titulo = titulo;
11-         this.autor = autor;
12-         this.fechaPublicacion = fechaPublicacion;
13-         this.valor = valor;
14-     }
15-
16-     // Métodos
17-     public void seleccionar() {
18-         System.out.println("Has seleccionado la enciclopedia: " + titulo);
19-     }
20-
21-     public void leer() {
22-         System.out.println("Estás leyendo la enciclopedia: " + titulo);
23-     }
24- }

```

Descripción

- **public class Enciclopedia** {Define la clase Enciclopedia que representa una enciclopedia.
- **Atributos privados:**
 - private String titulo;** Guarda el título de la enciclopedia.
 - private String autor;** Guarda el nombre del autor.
 - private String fechaPublicacion;** Registra la fecha de publicación.
 - private double valor;** Define el precio o valor de la enciclopedia.
- **Constructor:**
 - public Enciclopedia(String titulo, String autor, String fechaPublicacion, double valor) {**
Inicializa un objeto de Enciclopedia con los valores para titulo, autor, fechaPublicacion y valor.
 - this.titulo = titulo;, this.autor = autor;, this.fechaPublicacion = fechaPublicacion;, this.valor = valor;** Asigna los parámetros recibidos a los atributos de la clase.
- **Método seleccionar():**
 - **public void seleccionar() {**: Método público sin retorno que muestra un mensaje al seleccionar la enciclopedia.
 - **System.out.println("Has seleccionado la enciclopedia: " + titulo);**: Imprime el mensaje incluyendo el título de la enciclopedia.

- **Método leer():**

public void leer() {: Método público sin retorno que simula leer la enciclopedia.

System.out.println("Estás leyendo la enciclopedia: " + titulo);: Imprime el mensaje incluyendo el título de la enciclopedia.

```
24
25 ~ public void comprar() {
26     System.out.println("Has comprado la enciclopedia: " + titulo);
27 }
28
29 ~ public void regalar() {
30     System.out.println("Has regalado la enciclopedia: " + titulo);
31 }
32
33 // Getters y Setters
34 ~ public String getTitulo() {
35     return titulo;
36 }
37
38 ~ public void setTitulo(String titulo) {
39     this.titulo = titulo;
40 }
41
42 ~ public String getAutor() {
43     return autor;
44 }
45
46 ~ public void setAutor(String autor) {
47     this.autor = autor;
48 }
```

Descripción

Método comprar():

Permite registrar la acción de compra de la enciclopedia mostrando un mensaje:

System.out.println("Has comprado la enciclopedia: " + titulo);

Método regalar():

Registra que la enciclopedia ha sido regalada mediante un mensaje:

System.out.println("Has regalado la enciclopedia: " + titulo);

Getters y Setters:

getTitulo(): Devuelve el valor actual del atributo titulo.

getTitulo(String titulo): Permite actualizar el valor del atributo titulo.

getAutor(): Devuelve el valor actual del atributo autor.

setAutor(String autor): Permite actualizar el valor del atributo autor.

```
49
50 ~ public String getFechaPublicacion() {
51     return fechaPublicacion;
52 }
53
54 ~ public void setFechaPublicacion(String fechaPublicacion) {
55     this.fechaPublicacion = fechaPublicacion;
56 }
57
58 ~ public double getValor() {
59     return valor;
60 }
61
62 ~ public void setValor(double valor) {
63     this.valor = valor;
64 }
```

```
65
66 // Método toString para mostrar información de la enciclopedia
67 @Override
68 ~ public String toString() {
69     return "Detalles de la Enciclopedia:\n" +
70         "Título: " + titulo + "\n" +
71         "Autor: " + autor + "\n" +
72         "Fecha de Publicación: " + fechaPublicacion + "\n" +
73         "Valor: $" + valor;
74 }
75 }
```

Descripción

- **getFechaPublicacion():** Devuelve el valor actual del atributo fechaPublicacion.
- **setFechaPublicacion(String fechaPublicacion):** Permite actualizar el valor del atributo fechaPublicacion.
- **getValor():** Devuelve el valor actual del atributo valor.
- **setValor(double valor):** Permite actualizar el valor del atributo valor.

Salida

Si ejecutas el programa, la salida será algo como esto:

Has seleccionado la enciclopedia: Enciclopedia Britannica

Estás leyendo la enciclopedia: Enciclopedia Britannica

Has comprado la enciclopedia: Enciclopedia Britannica

Has regalado la enciclopedia: Enciclopedia Britannica

Detalles de la Enciclopedia:

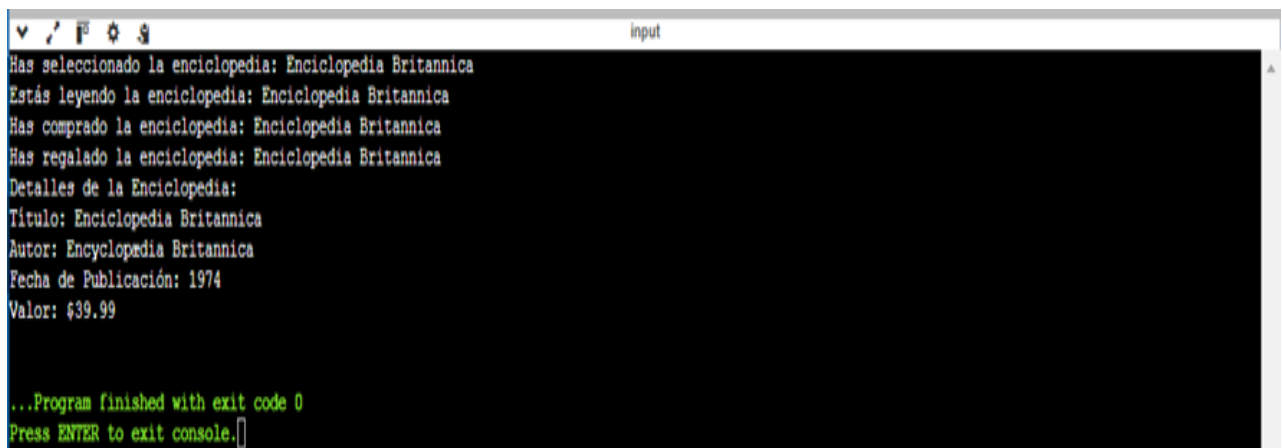
Título: Enciclopedia Britannica

Autor: Encyclopædia Britannica

Fecha de Publicación: 1974

Valor: \$39.99

El programa simula la interacción con una enciclopedia, mostrando cómo se podrían manejar objetos con atributos y comportamientos específicos.

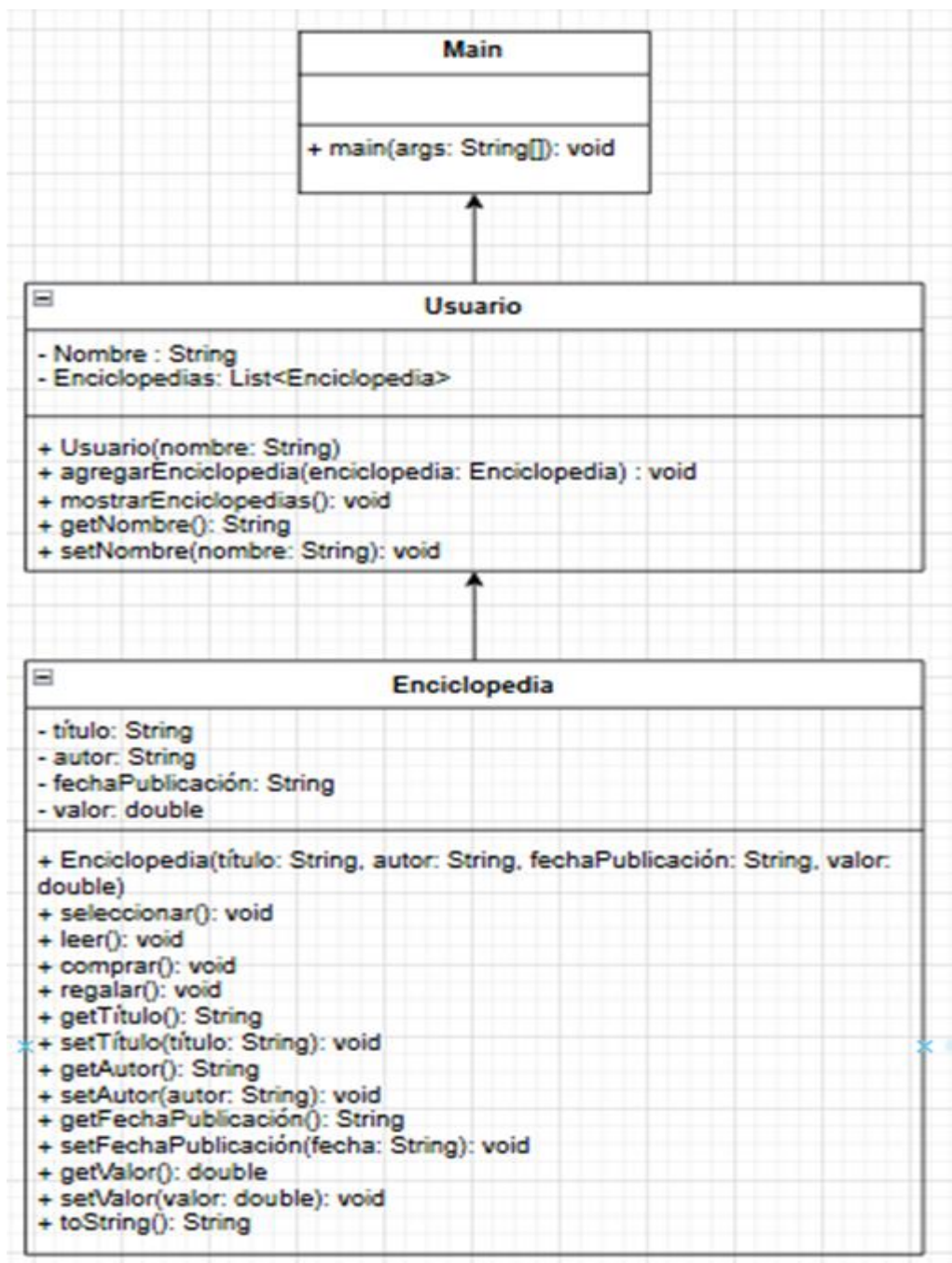


```
input
Has seleccionado la enciclopedia: Enciclopedia Britannica
Estás leyendo la enciclopedia: Enciclopedia Britannica
Has comprado la enciclopedia: Enciclopedia Britannica
Has regalado la enciclopedia: Enciclopedia Britannica
Detalles de la Enciclopedia:
Título: Enciclopedia Britannica
Autor: Encyclopædia Britannica
Fecha de Publicación: 1974
Valor: $39.99

...Program finished with exit code 0
Press ENTER to exit console.
```

- **Usuario:** Los usuarios pueden tener una lista de enciclopedias seleccionadas. Se relaciona con Enciclopedia mediante composición.

DIAGRAMA UML:



Explicación del diagrama UML

Clase Main

Es la clase principal del programa, desde donde se inicia la ejecución.

- **Atributo:** No tiene atributos definidos.
- **Método: `main(args: String[]): void`:** Método principal que inicia el programa. Es de tipo void porque no devuelve ningún valor.

Clase Usuario

Representa a un usuario que interactúa con enciclopedias.

- **Atributos:**

nombre: String: Almacena el nombre del usuario.

enciclopedias: List<Enciclopedia>: Lista que almacena objetos del tipo Enciclopedia, asociadas al usuario.

- **Métodos:**

Usuario(nombre: String): Constructor de la clase que inicializa el nombre del usuario.

agregarEnciclopedia(enciclopedia: Enciclopedia): void: Permite añadir una enciclopedia a la lista del usuario.

mostrarEnciclopedias(): void: Muestra todas las enciclopedias asociadas al usuario.

getNombre(): String: Devuelve el valor actual del atributo nombre.

setNombre(nombre: String): void: Permite actualizar el valor del atributo nombre.

Clase Enciclopedia

Representa una enciclopedia con detalles sobre su título, autor, fecha de publicación y valor.

- **Atributos:**

título: String: Almacena el título de la enciclopedia.

autor: String: Almacena el nombre del autor.

fechaPublicacion: String: Almacena la fecha de publicación.

valor: double: Almacena el precio o valor de la enciclopedia.

- **Métodos:**

- **Enciclopedia(título: String, autor: String, fechaPublicacion: String, valor: double):** Constructor que inicializa los atributos de la enciclopedia.
- **seleccionar(): void:** Registra la acción de seleccionar la enciclopedia.
- **leer(): void:** Permite registrar la acción de leer la enciclopedia.
- **comprar(): void:** Permite registrar la acción de compra.
- **regalar(): void:** Permite registrar que la enciclopedia ha sido regalada.
- **getTitulo(): String:** Devuelve el valor de título.
- **setTitulo(título: String): void:** Actualiza el valor de título.
- **getAutor(): String:** Devuelve el valor de autor.
- **setAutor(autor: String): void:** Actualiza el valor de autor.
- **getFechaPublicacion(): String:** Devuelve el valor de fechaPublicacion.
- **setFechaPublicacion(fecha: String): void:** Actualiza el valor de fechaPublicacion.
- **getValor(): double:** Devuelve el valor de valor.
- **setValor(valor: double): void:** Actualiza el valor de valor.

Código

Main.java

```
Main.java  Enciclopedia.java  Usuario.java
1 public class Main {
2     public static void main(String[] args) {
3         // Crear el primer usuario
4         Usuario usuario1 = new Usuario("Samir Samandé");
5
6         // Crear algunas enciclopedias
7         Enciclopedia enciclopedia1 = new Enciclopedia("Enciclopedia Britannica", "Encyclopedia Britannica", "1974", 39.99);
8         Enciclopedia enciclopedia2 = new Enciclopedia("Enciclopedia Moderna", "John Doe", "1980", 29.99);
9
10        // Agregar enciclopedias a la lista del primer usuario
11        usuario1.agregarEnciclopedia(enciclopedia1);
12        usuario1.agregarEnciclopedia(enciclopedia2);
13
14        // Mostrar enciclopedias seleccionadas por el primer usuario
15        usuario1.mostrarEnciclopedias();
16
17        // Usar métodos de la primera enciclopedia
18        enciclopedia1.seleccionar();
19        enciclopedia1.leer();
20        enciclopedia1.comprar();
21        enciclopedia1.regalar();
22
23        // Mostrar información de la primera enciclopedia
24        System.out.println(enciclopedia1);
25
26        // Crear el segundo usuario
27        Usuario usuario2 = new Usuario("Steven Guerrero");
28
29        // Crear más enciclopedias
30        Enciclopedia enciclopedia3 = new Enciclopedia("Enciclopedia Larousse - Pequeño Larousse ilustrado", "Raquel GU", "2019", 9.99);
31        Enciclopedia enciclopedia4 = new Enciclopedia("Enciclopedia de Primaria", "PARRAMON EDICIONES, S.A.", "2011", 29.99);
32
33        // Agregar enciclopedias a la lista del segundo usuario
34        usuario2.agregarEnciclopedia(enciclopedia3);
35        usuario2.agregarEnciclopedia(enciclopedia4);
36    }
```

Descripción

La clase Main representa el punto de entrada del programa, donde se ejecutan las interacciones entre los usuarios y sus enciclopedias. Dentro del método principal, se crean instancias de las clases Usuario y Enciclopedia, y se realiza una serie de acciones para gestionar estas relaciones.

- **Creación del primer usuario:** Se instancia un objeto de la clase Usuario llamado usuario1 con el nombre "Samir Samandé".

Usuario usuario1 = new Usuario("Samir Samandé");

- **Creación de enciclopedias:** Se crean dos objetos de la clase Enciclopedia, enciclopedia 1 y enciclopedias 2, cada uno inicializado con su título, autor, fecha de publicación y valor.

Enciclopedia enciclopedia1 = new Enciclopedia("Enciclopedia Británica", "Enciclopedia Británica", "1974", 39.99);

Enciclopedia enciclopedia2 = new Enciclopedia("Enciclopedia Moderna", "John Doe", "1980", 29.99);

- **Asociación de enciclopedias al primer usuario:** Las enciclopedias creadas se agregan a la lista de enciclopedias del usuario usuario1 mediante el método

agregarEnciclopedia.

usuario1.agregarEnciclopedia(enciclopedia1);

usuario1.agregarEnciclopedia(enciclopedia2);

- **Visualización de enciclopedias del primer usuario:** Se utiliza el método `mostrarEnciclopedias` para listar las enciclopedias asociadas a `usuario1`.
usuario1.mostrarEnciclopedias();
- **Acciones sobre la primera enciclopedia:** Se invocan los métodos `seleccionar`, `leer`, `comprar` y `regalar` en el objeto `enciclopedia1`, registrando interacciones con esta enciclopedia.

enciclopedia1.seleccionar();

enciclopedia1.leer();

enciclopedia1.comprar();

enciclopedia1.regalar();

- **Muestra la información de la primera enciclopedia:** Se imprime la representación de `enciclopedia1` en consola.

System.out.println(enciclopedia1);

- **Creación del segundo usuario:** Se instancia otro objeto de la clase `Usuario`, `usuario2`, con el nombre "Steven Guerrero".

Usuario usuario2 = new Usuario("Steven Guerrero");

- **Creación de más enciclopedias:** Se crean dos nuevas instancias de la clase `Enciclopedia`, `enciclopedia3` y `enciclopedia4`, con diferentes valores para sus atributos.

Enciclopedia enciclopedia3 = new Enciclopedia("Enciclopedia

Larousse - Pequeño Larousse ilustrado", "Raquel GU", "2019", 9.99);

**Enciclopedia enciclopedia4 = new Enciclopedia("Enciclopedia de Primaria",
"PARRAMON EDICIONES, S.A.", "2011", 29.99);**

- **Asociación de enciclopedias al segundo usuario:** Las nuevas enciclopedias se añaden a la lista de `usuario2`.

usuario2.agregarEnciclopedia(enciclopedia3);

usuario2.agregarEnciclopedia(enciclopedia4);

```

36
37      // Mostrar enciclopedias seleccionadas por el segundo usuario
38      usuario2.mostrarEnciclopedias();
39
40      // Usar métodos de la primera enciclopedia del segundo usuario
41      enciclopedia3.seleccionar();
42      enciclopedia3.leer();
43      enciclopedia3.comprar();
44      enciclopedia3.regalar();
45
46      // Mostrar información de la primera enciclopedia del segundo usuario
47      System.out.println(enciclopedia3);
48  }
49 }

```

Descripción

Mostrar enciclopedias seleccionadas por el segundo usuario:

Este bloque llama al método `mostrarEnciclopedias()` del objeto `usuario2`, el cual imprime la lista de todas las enciclopedias asociadas al segundo usuario.

usuario2.mostrarEnciclopedias();

Usar métodos de la primera enciclopedia del segundo usuario:

- En esta sección, se realizan diversas acciones sobre `enciclopedia3` invocando los métodos disponibles en la clase `Enciclopedia`:
- **Seleccionar la enciclopedia:** Marca que se está trabajando con esta enciclopedia.
 java
 Copiar código
`enciclopedia3.seleccionar();`
- **Leer la enciclopedia:** Representa la acción de lectura.
`enciclopedia3.leer();`
- **Comprar la enciclopedia:** Simula la acción de adquirir la enciclopedia.
`enciclopedia3.comprar();`
- **Regalar la enciclopedia:** Registra que la enciclopedia ha sido regalada.
`enciclopedia3.regalar();`

Mostrar información de la primera enciclopedia del segundo usuario:

- Finalmente, este bloque imprime los detalles de la enciclopedia `enciclopedia3` utilizando el método `System.out.println`, que probablemente llama al método `toString` de la clase `Enciclopedia` para mostrar su información.
System.out.println(enciclopedia3);

Usuario.java


```

Main.java : Enciclopedia.java : Usuario.java :
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class Usuario {
5     // Atributos
6     private String nombre;
7     private List<Enciclopedia> enciclopediasSeleccionadas;
8
9     // Constructor
10    public Usuario(String nombre) {
11        this.nombre = nombre;
12        this.enciclopediasSeleccionadas = new ArrayList<>();
13    }
14
15    // Métodos para gestionar enciclopedias
16    public void agregarEnciclopedia(Enciclopedia enciclopedia) {
17        enciclopediasSeleccionadas.add(enciclopedia);
18        System.out.println(enciclopedia.getTitulo() + " ha sido añadida a la lista de " + nombre);
19    }
20
21    public void mostrarEnciclopedias() {
22        System.out.println("Lista de enciclopedias seleccionadas por " + nombre + ":");
23        for (Enciclopedia enciclopedia : enciclopediasSeleccionadas) {
24            System.out.println(enciclopedia.getTitulo());
25        }
26    }
27
28    // Getters y Setters
29    public String getNombre() {
30        return nombre;
31    }
32
33    public void setNombre(String nombre) {
34        this.nombre = nombre;
35    }
36 }

```

Código de la clase Usuario

```

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

```

- **Importaciones:** Estas líneas importan las clases ArrayList y List de la biblioteca de Java.

List: Es una interfaz que define listas genéricas.

ArrayList: Es una implementación de la interfaz List, que permite almacenar elementos en una lista dinámica

public class Usuario {

- **Definición de la clase:** Declara una clase pública llamada Usuario. Representa a un usuario que puede seleccionar enciclopedias.

private String nombre;

private List<Enciclopedia> enciclopediasSeleccionadas;

- **Atributos privados:**

nombre: Almacena el nombre del usuario como una cadena de texto.

enciclopediasSeleccionadas: Es una lista de objetos de tipo Enciclopedia asociada al usuario.

```
public Usuario(String nombre) {  
    this.nombre = nombre;  
    this.enciclopediasSeleccionadas = new ArrayList<>();  
}
```

- **Constructor:**

String nombre: Recibe como parámetro el nombre del usuario y lo asigna al atributo nombre con `this.nombre = nombre`.

Inicialización de la lista: Crea una nueva lista vacía de enciclopedias (ArrayList) para el usuario.

```
public void agregarEnciclopedia(Enciclopedia enciclopedia) {  
    enciclopediasSeleccionadas.add(enciclopedia);  
    System.out.println(enciclopedia.getTitulo() + " ha sido añadida a la lista  
de " + nombre);  
}
```

- **Método agregarEnciclopedia:**

Enciclopedia: Recibe un objeto Enciclopedia como parámetro.

Añade la enciclopedia a la lista `enciclopediasSeleccionadas` usando el método `add()`.

Imprime un mensaje indicando que la enciclopedia ha sido añadida a la lista del usuario.

```

public void mostrarEnciclopedias() {
    System.out.println("Lista de enciclopedias seleccionadas por " + nombre
+ ":");
    for (Enciclopedia enciclopedia : enciclopediasSeleccionadas) {
        System.out.println(enciclopedia.getTitulo());
    }
}

```

- **Método mostrarEnciclopedias:**

Imprime un encabezado indicando que se mostrarán las enciclopedias seleccionadas por el usuario.

Usa un bucle for para recorrer la lista enciclopediasSeleccionadas e imprime el título de cada enciclopedia usando getTitulo().

```

public String getNombre() {
    return nombre;
}

```

- **Método getNombre:** Devuelve el valor del atributo nombre.

```

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

```

Salida

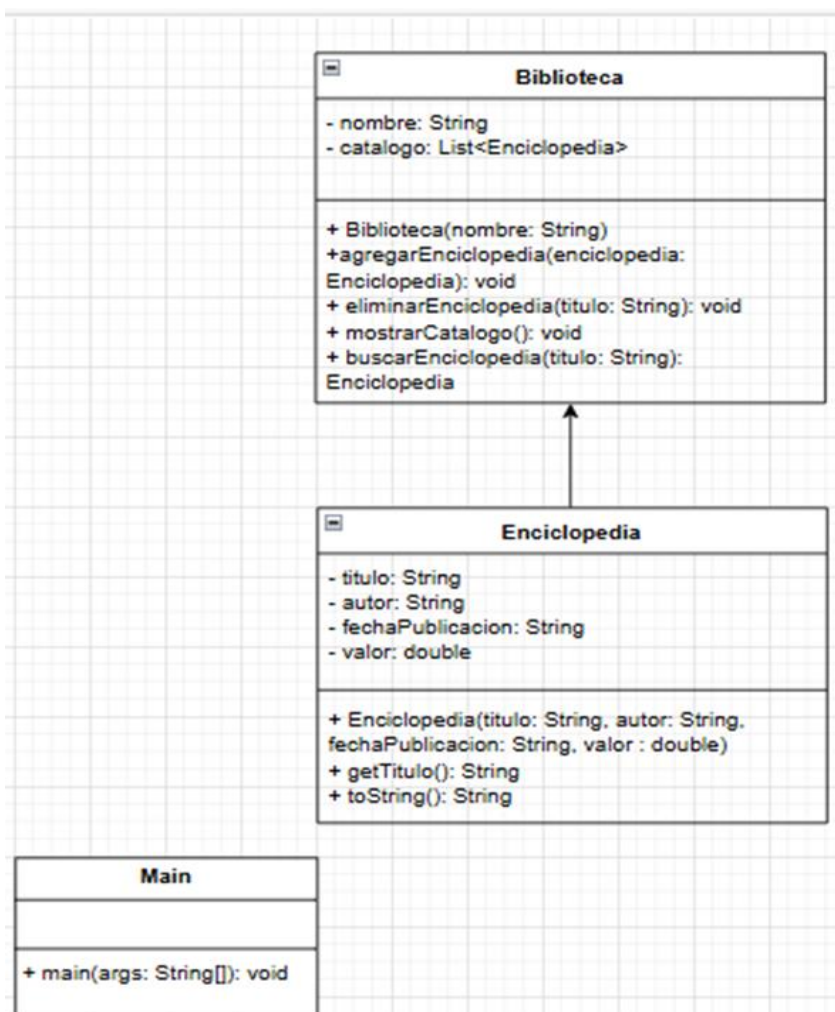
```

Enciclopedia Britannica ha sido añadida a la lista de Samir Samandé
Enciclopedia Moderna ha sido añadida a la lista de Samir Samandé
Lista de enciclopedias seleccionadas por Samir Samandé:
Enciclopedia Britannica
Enciclopedia Moderna
Has seleccionado la enciclopedia: Enciclopedia Britannica
Estás leyendo la enciclopedia: Enciclopedia Britannica
Has comprado la enciclopedia: Enciclopedia Britannica
Has regalado la enciclopedia: Enciclopedia Britannica
Detalles de la Enciclopedia:
Título: Enciclopedia Britannica
Autor: Encyclopædia Britannica
Fecha de Publicación: 1974
Valor: $39.99
Enciclopedia Larousse - Pequeño Larousse ilustrado ha sido añadida a la lista de Steven Guerrero
Enciclopedia de Primaria ha sido añadida a la lista de Steven Guerrero
Lista de enciclopedias seleccionadas por Steven Guerrero:
Enciclopedia Larousse - Pequeño Larousse ilustrado
Enciclopedia de Primaria
Has seleccionado la enciclopedia: Enciclopedia Larousse - Pequeño Larousse ilustrado
Estás leyendo la enciclopedia: Enciclopedia Larousse - Pequeño Larousse ilustrado
Has comprado la enciclopedia: Enciclopedia Larousse - Pequeño Larousse ilustrado
Has regalado la enciclopedia: Enciclopedia Larousse - Pequeño Larousse ilustrado
Detalles de la Enciclopedia:
Título: Enciclopedia Larousse - Pequeño Larousse ilustrado
Autor: Raquel GU
Fecha de Publicación: 2019
Valor: $9.99
...Program finished with exit code 0
Press ENTER to exit console.

```

- **Biblioteca:** Administra un catálogo de enciclopedias y se relaciona con Enciclopedia por asociación.

DIAGRAMA UML:



Explicación del Diagrama UML

Clase Enciclopedia

Esta clase representa una enciclopedia y contiene los siguientes atributos y métodos:

Atributos (privados, indicados por el símbolo -):

- **titulo:** Almacena el título de la enciclopedia (tipo **String**).
- **autor:** Almacena el nombre del autor de la enciclopedia (tipo **String**).
- **fechaPublicacion:** Representa la fecha de publicación de la enciclopedia (tipo **String**).
- **valor:** Representa el valor monetario de la enciclopedia (tipo **double**).

Métodos (públicos, indicados por el símbolo +):

- **Constructor Enciclopedia:** Recibe como parámetros **titulo**, **autor**, **fechaPublicacion**, y **valor** para inicializar los atributos de la clase.
- **getTitulo(): String:** Devuelve el título de la enciclopedia.
- **toString(): String:** Proporciona una representación textual de la enciclopedia, probablemente incluyendo todos sus atributos.

Clase Biblioteca

Esta clase gestiona un catálogo de enciclopedias.

Atributos (privados, indicados por el símbolo -):

- **nombre:** Almacena el nombre de la biblioteca (tipo **String**).
- **catalogo:** Es una lista (**List<Enciclopedia>**) que contiene las enciclopedias disponibles en la biblioteca.

Métodos (públicos, indicados por el símbolo +):

- **Constructor Biblioteca:** Recibe un parámetro **nombre** para inicializar el nombre de la biblioteca y probablemente inicializa la lista **catalogo** como vacía.
- **agregarEnciclopedia(enciclopedia: Enciclopedia): void:** Permite añadir una nueva enciclopedia al catálogo.
- **eliminarEnciclopedia(titulo: String): void:** Permite eliminar una enciclopedia del catálogo usando su título como referencia.
- **mostrarCatalogo(): void:** Muestra todas las enciclopedias disponibles en el catálogo.
- **buscarEnciclopedia(titulo: String): Enciclopedia:** Busca y devuelve una enciclopedia del catálogo según su título.

Clase Main

Es el punto de entrada principal del programa, representada por:

Método principal:

- **main(args: String[]): void**: Método estándar en Java que inicia la ejecución del programa. Aquí se realizarían acciones como crear bibliotecas, añadir enciclopedias y manipular el catálogo.

CÓDIGO:

```
1 public class Main {  
2     public static void main(String[] args) {  
3         Biblioteca biblioteca = new Biblioteca("Biblioteca Central");  
4  
5         Enciclopedia enciclopedia1 = new Enciclopedia("Enciclopedia Britannica", "Encyclopædia Britannica", "1974", 29.99);  
6         Enciclopedia enciclopedia2 = new Enciclopedia("Enciclopedia Moderna", "John Doe", "1980", 39.99);  
7  
8         biblioteca.agregarEnciclopedia(enciclopedia1);  
9         biblioteca.agregarEnciclopedia(enciclopedia2);  
10  
11        biblioteca.mostrarCatalogo();  
12  
13        biblioteca.eliminarEnciclopedia("Enciclopedia Britannica");  
14        biblioteca.mostrarCatalogo();  
15    }  
16 }
```

Clase Main

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

- **Definición de la clase y método principal**: Declara la clase Main y su método principal main. Este método es el punto de entrada del programa.

```
Biblioteca biblioteca = new Biblioteca("Biblioteca Central");
```

- **Crear una instancia de la clase Biblioteca**:

Se crea un objeto biblioteca con el nombre "Biblioteca Central".

Inicializa el catálogo vacío (según la definición de la clase Biblioteca).

```
Enciclopedia enciclopedia1 = new Enciclopedia("Enciclopedia Britannica",  
"Encyclopaedia Britannica", "1974", 29.99);  
Enciclopedia enciclopedia2 = new Enciclopedia("Enciclopedia Moderna",  
"John Doe", "1980", 39.99);
```

- **Crear objetos de la clase Enciclopedia:**

enciclopedia1: Se crea con título "Enciclopedia Britannica", autor "Encyclopaedia Britannica", fecha de publicación "1974", y valor 29.99.

enciclopedia2: Se crea con título "Enciclopedia Moderna", autor "John Doe", fecha de publicación "1980", y valor 39.99.

```
biblioteca.agregarEnciclopedia(enciclopedia1);  
biblioteca.agregarEnciclopedia(enciclopedia2);
```

- **Añadir enciclopedias al catálogo de la biblioteca:**

Se llama al método `agregarEnciclopedia` para añadir `enciclopedia1` y `enciclopedia2` al catálogo de la biblioteca.

```
biblioteca.mostrarCatalogo();
```

- **Mostrar el catálogo de enciclopedias:**

Se llama al método `mostrarCatalogo`, que imprime los títulos y detalles de todas las enciclopedias contenidas en el catálogo.

```
biblioteca.eliminarEnciclopedia("Enciclopedia Britannica");
```

- **Eliminar una enciclopedia del catálogo:**

Se llama al método `eliminarEnciclopedia` para buscar y eliminar la enciclopedia con el título "Enciclopedia Britannica" del catálogo.

```
biblioteca.mostrarCatalogo();
```

- **Mostrar nuevamente el catálogo:**

Se llama otra vez al método `mostrarCatalogo` para verificar que la enciclopedia eliminada ya no está en la lista.

```
Main.java | Enciclopedia.java | Biblioteca.java |
1 public class Enciclopedia {
2     private String titulo;
3     private String autor;
4     private String fechaPublicacion;
5     private double valor;
6
7     // Constructor
8     public Enciclopedia(String titulo, String autor, String fechaPublicacion, double valor) {
9         this.titulo = titulo;
10        this.autor = autor;
11        this.fechaPublicacion = fechaPublicacion;
12        this.valor = valor;
13    }
14
15    // Métodos Getter
16    public String getTitulo() {
17        return titulo;
18    }
19
20    @Override
21    public String toString() {
22        return "Titulo: " + titulo + ", Autor: " + autor + ", Fecha de Publicación: " + fechaPublicacion + ", Valor: " + valor;
23    }
24 }
```

Descripcion

- **Encabezado y atributos**

public class Enciclopedia {

Declara una clase pública llamada Enciclopedia.

private String titulo;

Define un atributo privado de tipo String llamado titulo. Este almacenará el título de la enciclopedia.

private String autor;

Define un atributo privado de tipo String llamado autor. Este almacenará el nombre del autor.

private String fechaPublicacion;

Define un atributo privado de tipo String llamado fechaPublicacion. Este representará la fecha de publicación.

private double valor;

Define un atributo privado de tipo `double` llamado `valor`. Este guardará el valor monetario de la enciclopedia.

- **Constructor**

public Enciclopedia(String titulo, String autor, String fecha Publicación, double valor) {

Declara un constructor público de la clase `Enciclopedia` que recibe cuatro parámetros (`titulo`, `autor`, `fecha Publicacion`, `valor`) para inicializar los atributos de la clase.

this.titulo = titulo;

Asigna el valor del parámetro `titulo` al atributo `titulo` de la instancia actual (`this`).

this.autor = autor;

Asigna el valor del parámetro `autor` al atributo `autor` de la instancia actual.

this.fecha Publicación = fechaPublicacion;

Asigna el valor del parámetro `fechaPublicacion` al atributo `fechaPublicacion` de la instancia actual.

this.valor = valor;

Asigna el valor del parámetro `valor` al atributo `valor` de la instancia actual.

- **Métodos**

public String getTitulo() {

Declara un método público llamado `getTitulo` que devuelve un valor de tipo `String`.

return titulo;

Retorna el valor del atributo titulo.

- **Sobrescritura del método toString**

@Override

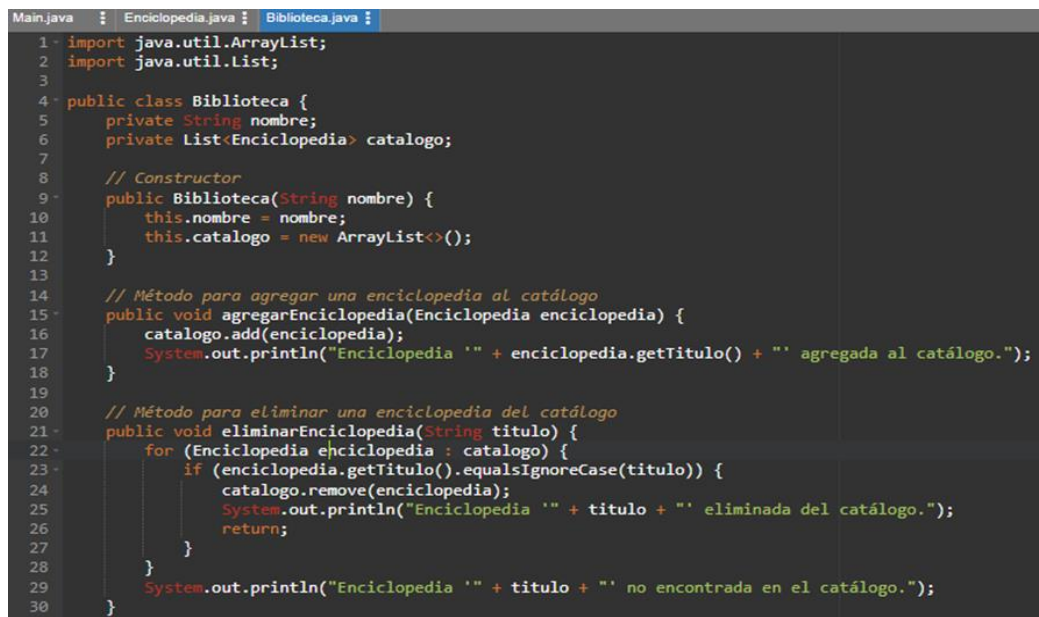
Anotación que indica que el método toString sobrescribe la implementación del método por defecto de la clase Object.

public String toString() {

Declara un método público llamado toString que retorna un valor de tipo String. Este método describe el objeto Enciclopedia en forma de texto.

return "Titulo: " + titulo + ", Autor: " + autor + ", Fecha de Publicación: " + fechaPublicacion + ", Valor: " + valor;

Construye y retorna una cadena que contiene el título, autor, fecha de publicación y valor de la enciclopedia, concatenando los valores de los atributos



```
1- import java.util.ArrayList;
2- import java.util.List;
3-
4- public class Biblioteca {
5-     private String nombre;
6-     private List<Enciclopedia> catalogo;
7-
8-     // Constructor
9-     public Biblioteca(String nombre) {
10-         this.nombre = nombre;
11-         this.catalogo = new ArrayList<>();
12-     }
13-
14-     // Método para agregar una enciclopedia al catálogo
15-     public void agregarEnciclopedia(Enciclopedia enciclopedia) {
16-         catalogo.add(enciclopedia);
17-         System.out.println("Enciclopedia '" + enciclopedia.getTitulo() + "' agregada al catálogo.");
18-     }
19-
20-     // Método para eliminar una enciclopedia del catálogo
21-     public void eliminarEnciclopedia(String titulo) {
22-         for (Enciclopedia enciclopedia : catalogo) {
23-             if (enciclopedia.getTitulo().equalsIgnoreCase(titulo)) {
24-                 catalogo.remove(enciclopedia);
25-                 System.out.println("Enciclopedia '" + titulo + "' eliminada del catálogo.");
26-                 return;
27-             }
28-         }
29-         System.out.println("Enciclopedia '" + titulo + "' no encontrada en el catálogo.");
30-     }
31- }
```

Importaciones

1. import java.util.ArrayList;

Importa la clase ArrayList de la biblioteca estándar de Java para manejar listas dinámicas.

2. import java.util.List;

Importa la interfaz List, que representa una lista genérica en Java.

Declaración de la clase y atributos

3. public class Biblioteca {

Declara una clase pública llamada **Biblioteca**.

4. private String nombre;

Declara un atributo privado de tipo **String** llamado **nombre** para almacenar el nombre de la biblioteca.

5. private List<Enciclopedia> catálogo;

Declara un atributo privado de tipo **List** (lista genérica) que almacenará objetos de tipo **Enciclopedia**. Esto representa el catálogo de la biblioteca.

Constructor

6. public Biblioteca(String nombre) {

Declara el constructor de la clase **Biblioteca**, que recibe como parámetro un nombre.

7. this.nombre = nombre;

Inicializa el atributo **nombre** con el valor del parámetro proporcionado.

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

Inicializa el atributo **catalogo** como una nueva instancia de **ArrayList**, creando un espacio para almacenar objetos **Enciclopedia**.

Método para agregar enciclopedias

9. public void agregarEnciclopedia(Enciclopedia enciclopedia) {

Declara un método público llamado **agregarEnciclopedia**, que recibe un objeto de tipo **Enciclopedia** como parámetro.

10. catalogo.add(enciclopedia);

Agrega el objeto **enciclopedia** al catálogo.

**11. System.out.println("Enciclopedia " + enciclopedia.getTitulo() + "
agregada al catálogo.");**

Imprime un mensaje indicando que la enciclopedia ha sido agregada al catálogo, utilizando el método `getTitulo` del objeto `Enciclopedia`.

Método para eliminar enciclopedias

12. `public void eliminarEnciclopedia(String titulo) {`

Declara un método público llamado `eliminarEnciclopedia`, que recibe un parámetro `titulo` de tipo `String`.

13. `for (Enciclopedia enciclopedia : catalogo) {`

Itera a través de cada objeto `Enciclopedia` en el catálogo utilizando un bucle `for-each`.

14. `if (enciclopedia.getTitulo().equalsIgnoreCase(titulo)) {`

Verifica si el título de la enciclopedia actual (obtenido con `getTitulo`) coincide con el título proporcionado, ignorando mayúsculas y minúsculas.

15. `catalogo.remove(enciclopedia);`

Si se encuentra una coincidencia, elimina la enciclopedia del catálogo.

16. `System.out.println("Enciclopedia " + titulo + " eliminada del catálogo.");`

Imprime un mensaje indicando que la enciclopedia fue eliminada.

17. `return;`

Termina la ejecución del método una vez que se ha eliminado la enciclopedia.

18. `}` (cierre del `if`)

19. `}` (cierre del bucle `for`)

20. `System.out.println("Enciclopedia " + titulo + " no encontrada en el catálogo.");`

```

31
32 // Método para mostrar el catálogo de enciclopedias
33 public void mostrarCatalogo() {
34     System.out.println("Catálogo de Enciclopedias en " + nombre + ":");
35     for (Enciclopedia enciclopedia : catalogo) {
36         System.out.println(enciclopedia.toString());
37     }
38 }
39
40 // Método para buscar una enciclopedia por título
41 public Enciclopedia buscarEnciclopedia(String titulo) {
42     for (Enciclopedia enciclopedia : catalogo) {
43         if (enciclopedia.getTitulo().equalsIgnoreCase(titulo)) {
44             return enciclopedia;
45         }
46     }
47     return null; // Si no se encuentra, se retorna null
48 }
49 }

```

Descripción

Método para mostrar el catálogo de enciclopedias

1. **public void mostrarCatalogo() {**

Declara un método público sin valor de retorno llamado `mostrarCatalogo`. Este método mostrará todas las enciclopedias en el catálogo.

2. **System.out.println("Catálogo de Enciclopedias en " + nombre + ":");**

Imprime un encabezado indicando el nombre de la biblioteca y que se listará su catálogo de enciclopedias.

3. **for (Enciclopedia enciclopedia : catalogo) {**

Itera a través de cada objeto `Enciclopedia` en el catálogo usando un bucle `for-each`.

4. **System.out.println(enciclopedia.toString());**

Para cada enciclopedia en el catálogo, llama al método `toString()` de la clase `Enciclopedia` y lo imprime en la consola, mostrando los detalles de la enciclopedia.

5. **}** (cierre del bucle for)

6. **}** (cierre del método `mostrarCatalogo`)

Método para buscar una enciclopedia por título

7. **public Enciclopedia buscarEnciclopedia(String titulo) {**

Declara un método público llamado `buscarEnciclopedia`, que recibe un parámetro `titulo` de tipo `String`. Este método retorna un objeto de tipo `Enciclopedia` si encuentra una coincidencia.

8. **for (Enciclopedia enciclopedia : catalogo) {**

Itera a través de cada objeto `Enciclopedia` en el catálogo usando un bucle `foreach`.

9. **if (enciclopedia.getTitulo().equalsIgnoreCase(titulo)) {**

Comprueba si el título de la enciclopedia actual coincide con el título dado, ignorando mayúsculas y minúsculas.

10. **return enciclopedia;**

Si se encuentra una coincidencia, retorna la instancia de la enciclopedia encontrada.

11. **}** (cierre del if)

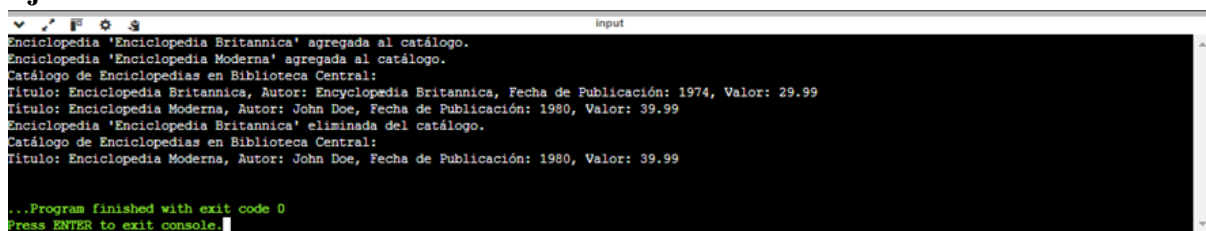
12. **}** (cierre del bucle for)

13. **return null; // Si no se encuentra, se retorna null**

Si el bucle termina sin encontrar una enciclopedia con el título dado, retorna `null`, indicando que no hubo coincidencia.

14. **}** (cierre del método `buscarEnciclopedia`)

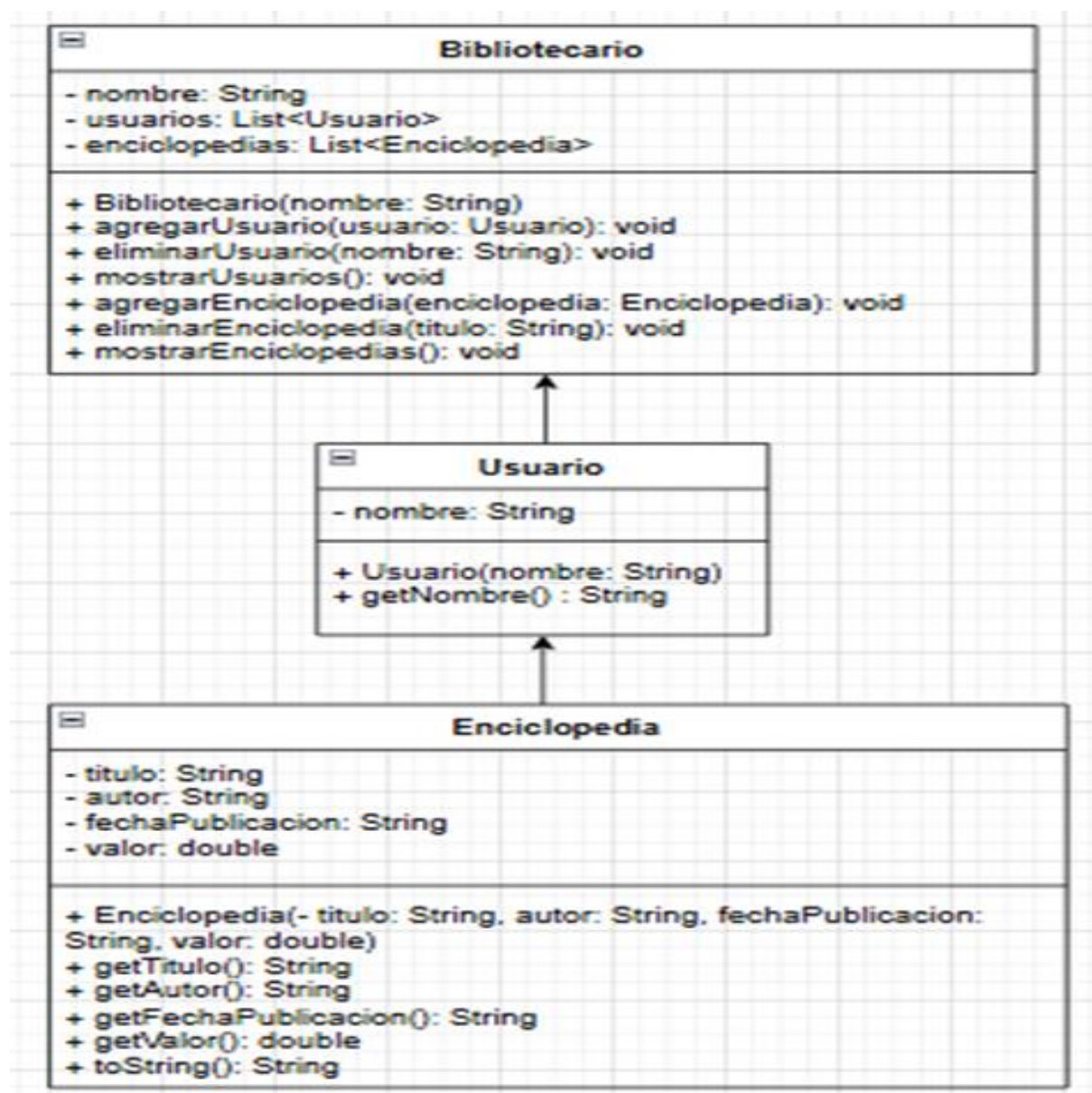
Ejecucion



```
Enciclopedia 'Enciclopedia Britannica' agregada al catálogo.
Enciclopedia 'Enciclopedia Moderna' agregada al catálogo.
Catálogo de Enciclopedias en Biblioteca Central:
Titulo: Enciclopedia Britannica, Autor: Encyclopædia Britannica, Fecha de Publicación: 1974, Valor: 29.99
Titulo: Enciclopedia Moderna, Autor: John Doe, Fecha de Publicación: 1980, Valor: 39.99
Enciclopedia 'Enciclopedia Britannica' eliminada del catálogo.
Catálogo de Enciclopedias en Biblioteca Central:
Titulo: Enciclopedia Moderna, Autor: John Doe, Fecha de Publicación: 1980, Valor: 39.99
...Program finished with exit code 0
Press ENTER to exit console.
```

- Bibliotecario: Gestiona usuarios y enciclopedias, y se relaciona con `Usuario` y `Enciclopedia` por asociación.

DIAGRAMA UML:



- **Clase Bibliotecario**

Esta clase parece ser el controlador principal del sistema, ya que maneja usuarios y enciclopedias.

- **Atributos:**

- **nombre: String**

- Representa el nombre del bibliotecario.

- **usuarios: List<Usuario>**

- Es una lista que almacena objetos de la clase Usuario, representando los usuarios bajo la gestión del bibliotecario.

- **enciclopedias: List<Enciclopedia>**

- Es una lista que almacena objetos de la clase Enciclopedia, representando el catálogo de enciclopedias.

- **Métodos:**

- + **Bibliotecario(nombre: String)**

- Constructor que inicializa la clase con el nombre del bibliotecario.

- + **agregarUsuario(usuario: Usuario): void**

- Método público para agregar un objeto Usuario a la lista de usuarios.

- + **eliminarUsuario(nombre: String): void**

- Método público para eliminar un usuario de la lista de usuarios según su nombre.

- + **mostrarUsuarios(): void**

- Método público para mostrar todos los usuarios gestionados.

- + **agregarEnciclopedia(enciclopedia: Enciclopedia): void**

- Método público para agregar una enciclopedia al catálogo.

- + **eliminarEnciclopedia(titulo: String): void**

- Método público para eliminar una enciclopedia del catálogo según su título.

- + **mostrarEnciclopedias(): void**

- Método público para mostrar todas las enciclopedias en el catálogo.

Clase Enciclopedia

Esta clase representa una enciclopedia en el sistema.

- **Atributos:**

- **titulo: String**

- Representa el título de la enciclopedia.

- **autor: String**

- Representa el autor de la enciclopedia.

- **fechaPublicacion: String**

- Representa la fecha de publicación de la enciclopedia.

- **valor: double**

- Representa el valor monetario de la enciclopedia.

- **Métodos:**

- + **Enciclopedia(titulo: String, autor: String, fechaPublicacion: String, valor: double)**

- Constructor que inicializa una enciclopedia con su título, autor, fecha de publicación y valor.

- + **getTitulo(): String**

- Método público para obtener el título de la enciclopedia.

- + **getAutor(): String**

- Método público para obtener el autor de la enciclopedia.

- + **getFechaPublicacion(): String**

- Método público para obtener la fecha de publicación de la enciclopedia.

- + **getValor(): double**

- Método público para obtener el valor de la enciclopedia.

+ toString(): String

Método público que retorna una representación textual de la enciclopedia.

- **Clase Usuario**

Esta clase representa a un usuario registrado.

- **Atributos:**

- nombre: String
Almacena el nombre del usuario.

- **Métodos:**

- + Usuario(nombre: String)
Constructor que inicializa la clase con el nombre del usuario.
- + getNombre(): String
Método público para obtener el nombre del usuario.

CÓDIGO:

```
Main.java : Bibliotecario.java : Usuario.java : Enciclopedia.java :
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Bibliotecario bibliotecario = new Bibliotecario("Luis Jaramillo");
6         Scanner scanner = new Scanner(System.in);
7         int opcion;
8
9         do {
10             System.out.println("\n--- Menú Bibliotecario ---");
11             System.out.println("1. Agregar Usuario");
12             System.out.println("2. Eliminar Usuario");
13             System.out.println("3. Mostrar Usuarios");
14             System.out.println("4. Agregar Enciclopedia");
15             System.out.println("5. Eliminar Enciclopedia");
16             System.out.println("6. Mostrar Enciclopedias");
17             System.out.println("0. Salir");
18             System.out.print("Selecciona una opción: ");
19             opcion = scanner.nextInt();
20             scanner.nextLine(); // Limpiar el buffer
```

Descripción

- **import java.util.Scanner;**
Importa la clase Scanner para leer datos del usuario.
- **public class Main {**
Declara la clase principal.
- **public static void main(String[] args) {**
Método principal donde comienza la ejecución.
- **Bibliotecario bibliotecario = new Bibliotecario("Luis Jaramillo");**
Crea un objeto Bibliotecario inicializado con "Luis Jaramillo".
- **Scanner scanner = new Scanner(System.in);**
Crea un objeto Scanner para leer entradas del usuario.
- **int opcion;**
Declara la variable opcion para guardar la opción del menú.
- **do {**
Inicia un ciclo para mostrar el menú repetidamente.
- **System.out.println(...)**
Imprime las opciones del menú.
- **opcion = scanner.nextInt();**
Lee la opción seleccionada por el usuario.
- **scanner.nextLine();**
Limpia el buffer después de leer un número.

```

21
22
23     switch (opcion) {
24         case 1:
25             System.out.print("Ingrese el nombre del usuario: ");
26             String nombreUsuario = scanner.nextLine();
27             bibliotecario.agregarUsuario(new Usuario(nombreUsuario));
28             System.out.println("Usuario agregado.");
29             break;
30         case 2:
31             System.out.print("Ingrese el nombre del usuario a eliminar: ");
32             String nombreEliminar = scanner.nextLine();
33             bibliotecario.eliminarUsuario(nombreEliminar);
34             System.out.println("Usuario eliminado.");
35             break;
36         case 3:
37             bibliotecario.mostrarUsuarios();
38             break;
39         case 4:
40             System.out.print("Ingrese el título de la enciclopedia: ");
41             String titulo = scanner.nextLine();
42             System.out.print("Ingrese el autor de la enciclopedia: ");
43             String autor = scanner.nextLine();
44             System.out.print("Ingrese la fecha de publicación: ");
45             String fechaPublicacion = scanner.nextLine();
46             System.out.print("Ingrese el valor de la enciclopedia $: ");
47             double valor = scanner.nextDouble();
48             scanner.nextLine(); // Limpiar el buffer
49             bibliotecario.agregarEnciclopedia(new Enciclopedia(titulo, autor, fechaPublicacion, valor));
50             System.out.println("Enciclopedia agregada.");
51             break;
52         case 5:
53             System.out.print("Ingrese el título de la enciclopedia a eliminar: ");
54             String tituloEliminar = scanner.nextLine();
55             bibliotecario.eliminarEnciclopedia(tituloEliminar);
56             System.out.println("Enciclopedia eliminada.");
57             break;

```

Explicación

Estructura switch

Se usa para ejecutar diferentes bloques de código según la opción seleccionada en el menú.

1. case 1:

Pide al usuario un nombre (nombreUsuario) y lo agrega a través de agregarUsuario().

Imprime "Usuario agregado."

2. case 2:

Pide un nombre (nombreEliminar) para eliminar un usuario con eliminarUsuario().

Imprime "Usuario eliminado."

3. case 3:

Llama a mostrarUsuarios() para mostrar todos los usuarios.

4. case 4:

Pide los datos de una enciclopedia: título, autor, fecha y valor.

Crea un objeto Enciclopedia con esos datos y lo agrega usando agregarEnciclopedia().

Imprime "Enciclopedia agregada."

5. case 5:

Pide el título de una enciclopedia (tituloEliminar) y la elimina con eliminarEnciclopedia().

Imprime "Enciclopedia eliminada."

6. break;

Finaliza cada caso para evitar que se ejecuten los demás.

```
57         case 6:
58             bibliotecario.mostrarEnciclopedias();
59             break;
60         case 0:
61             System.out.println("Saliendo...");
62             break;
63         default:
64             System.out.println("Opción no válida. Intente de nuevo.");
65     }
66     } while (opcion != 0);
67
68     scanner.close();
69 }
70 }
```

Explicación

Casos adicionales en el switch

1. case 6:

Llama al método mostrarEnciclopedias() para mostrar todas las enciclopedias.

break;: Finaliza este caso.

2. case 0:

Imprime "Saliendo..." para indicar que el programa finalizará.

break;: Finaliza este caso.

3. default:

Mensaje para opciones no válidas: "Opción no válida. Intente de nuevo."

Cierre del ciclo

```
} while (opcion != 0);
```

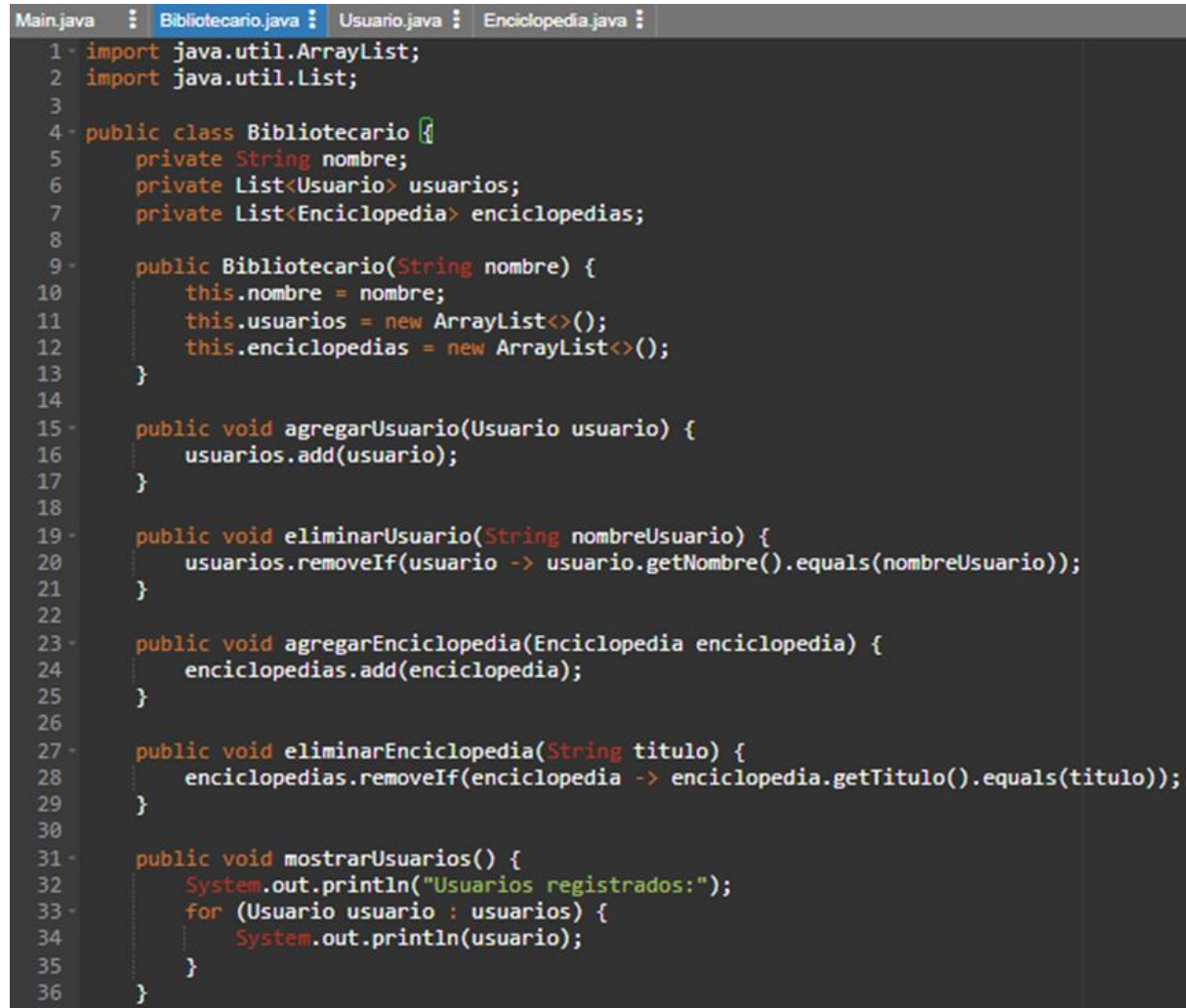
El ciclo se repite mientras la opción sea diferente de 0.

Si opcion es 0, el programa sale del ciclo.

Cerrar Scanner

```
scanner.close();
```

- Libera los recursos utilizados por el objeto Scanner.



```
Main.java : Bibliotecario.java : Usuario.java : Enciclopedia.java :
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class Bibliotecario {
5     private String nombre;
6     private List<Usuario> usuarios;
7     private List<Enciclopedia> enciclopedias;
8
9     public Bibliotecario(String nombre) {
10         this.nombre = nombre;
11         this.usuarios = new ArrayList<>();
12         this.enciclopedias = new ArrayList<>();
13     }
14
15     public void agregarUsuario(Usuario usuario) {
16         usuarios.add(usuario);
17     }
18
19     public void eliminarUsuario(String nombreUsuario) {
20         usuarios.removeIf(usuario -> usuario.getNombre().equals(nombreUsuario));
21     }
22
23     public void agregarEnciclopedia(Enciclopedia enciclopedia) {
24         enciclopedias.add(enciclopedia);
25     }
26
27     public void eliminarEnciclopedia(String titulo) {
28         enciclopedias.removeIf(enciclopedia -> enciclopedia.getTitulo().equals(titulo));
29     }
30
31     public void mostrarUsuarios() {
32         System.out.println("Usuarios registrados:");
33         for (Usuario usuario : usuarios) {
34             System.out.println(usuario);
35         }
36     }
37 }
```

Explicación

1. Atributos:

String nombre: Nombre del bibliotecario.

List<Usuario> usuarios: Lista de usuarios registrados.

List<Enciclopedia> enciclopedias: Lista de enciclopedias disponibles.

2. Constructor:

Inicializa nombre, y crea listas vacías para usuarios y enciclopedias.

3. Métodos:

agregarUsuario: Añade un usuario a la lista.

eliminarUsuario: Elimina un usuario por nombre.

agregarEnciclopedia: Añade una enciclopedia a la lista.

eliminarEnciclopedia: Elimina una enciclopedia por título.

mostrarUsuarios: Imprime en consola la lista de usuarios registrados.

```
37
38 -   public void mostrarEnciclopedias() {
39       System.out.println("Enciclopedias disponibles:");
40       for (Enciclopedia enciclopedia : enciclopedias) {
41           System.out.println(enciclopedia);
42       }
43   }
44 }
```

Explicacion

System.out.println("Enciclopedias disponibles:");

- Imprime un encabezado para indicar que se mostrarán las enciclopedias.

Bucle for:

- Itera sobre cada objeto Enciclopedia de la lista enciclopedias.
- System.out.println(enciclopedia);: Imprime cada enciclopedia.
Nota: Esto asume que la clase Enciclopedia tiene implementado un método toString() que define cómo se debe mostrar la información de la enciclopedia.

```

Main.java  Bibliotecario.java  Usuario.java  Enciclopedia.java
1  public class Usuario {
2      private String nombre;
3
4      public Usuario(String nombre) {
5          this.nombre = nombre;
6      }
7
8      public String getNombre() {
9          return nombre;
10     }
11
12     @Override
13     public String toString() {
14         return "Usuario: " + nombre;
15     }
16 }

```

Explicación

Atributo

Nombre: Almacena el nombre del usuario.

Métodos

Usuario(String nombre): Constructor que inicializa el nombre.

getNombre(): Retorna el nombre del usuario.

toString(): Devuelve "Usuario: [nombre]" para mostrar al usuario como texto.


```

Main.java : Bibliotecario.java : Usuario.java : Enciclopedia.java :
1 public class Enciclopedia {
2     private String titulo;
3     private String autor;
4     private String fechaPublicacion;
5     private double valor;
6
7     public Enciclopedia(String titulo, String autor, String fechaPublicacion, double valor) {
8         this.titulo = titulo;
9         this.autor = autor;
10        this.fechaPublicacion = fechaPublicacion;
11        this.valor = valor;
12    }
13
14    public String getTitulo() {
15        return titulo;
16    }
17
18    public String getAutor() {
19        return autor;
20    }
21
22    public String getFechaPublicacion() {
23        return fechaPublicacion;
24    }
25
26    public double getValor() {
27        return valor;
28    }
29
30    @Override
31    public String toString() {
32        return "Enciclopedia: " + titulo + ", Autor: " + autor + ", Fecha de Publicación: " + fechaPublicacion + ", Valor: " + valor;
33    }
34 }

```

Explicación

Atributos privados:

- **titulo (String):** Almacena el título de la enciclopedia.
- **autor (String):** Almacena el autor de la enciclopedia.
- **fechaPublicacion (String):** Almacena la fecha de publicación.
- **valor (double):** Almacena el valor (precio).

Constructor: Permite inicializar una instancia de la clase con los valores para los atributos.

Métodos get: Proveen acceso a los valores de los atributos (getter para cada uno).

Método toString: Sobrescribe el método por defecto para retornar una representación en texto legible del objeto.

EJECUCIÓN

```
input
--- Menú Bibliotecario ---
1. Agregar Usuario
2. Eliminar Usuario
3. Mostrar Usuarios
4. Agregar Enciclopedia
5. Eliminar Enciclopedia
6. Mostrar Enciclopedias
0. Salir
Selecciona una opción: 1
Ingrese el nombre del usuario: Samir
Usuario agregado.

--- Menú Bibliotecario ---
1. Agregar Usuario
2. Eliminar Usuario
3. Mostrar Usuarios
4. Agregar Enciclopedia
5. Eliminar Enciclopedia
6. Mostrar Enciclopedias
0. Salir
Selecciona una opción: 2
Ingrese el nombre del usuario a eliminar: Samir
Usuario eliminado.
```

```
--- Menú Bibliotecario ---
1. Agregar Usuario
2. Eliminar Usuario
3. Mostrar Usuarios
4. Agregar Enciclopedia
5. Eliminar Enciclopedia
6. Mostrar Enciclopedias
0. Salir
Selecciona una opción: 3
Usuarios registrados:
Usuario: Samir
```

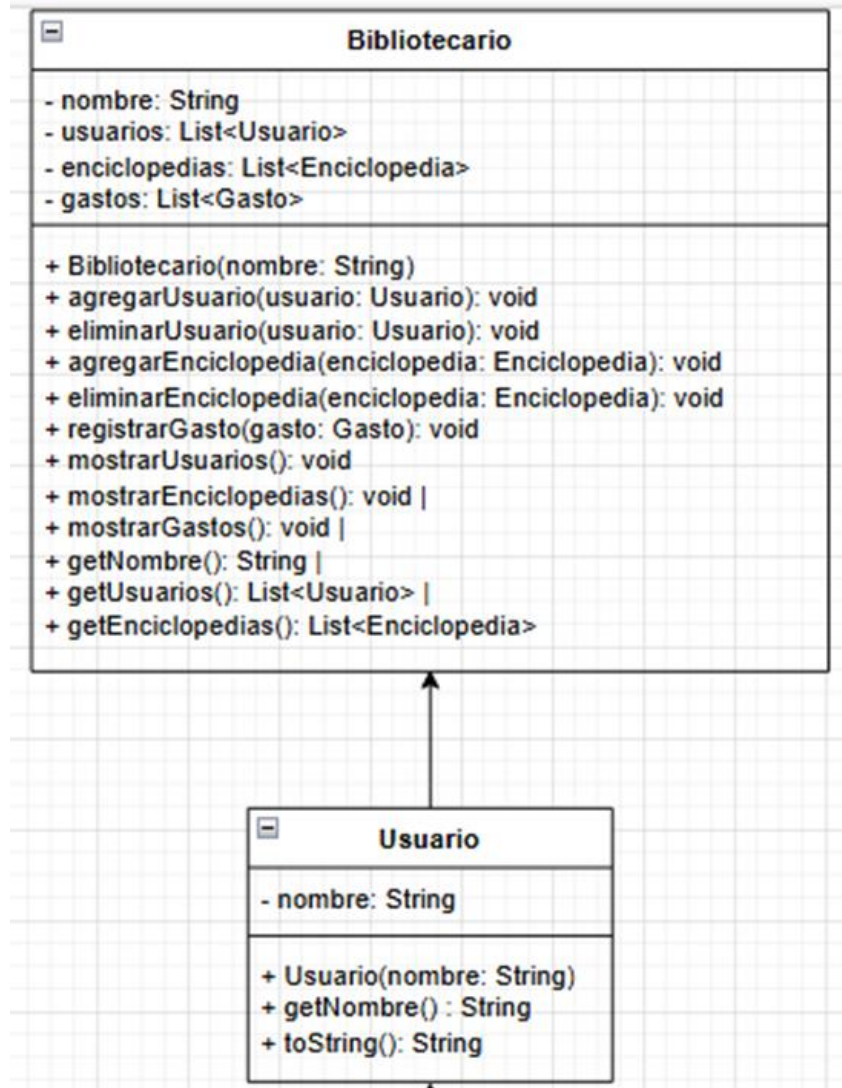
```
--- Menú Bibliotecario ---
1. Agregar Usuario
2. Eliminar Usuario
3. Mostrar Usuarios
4. Agregar Enciclopedia
5. Eliminar Enciclopedia
6. Mostrar Enciclopedias
0. Salir
Selecciona una opción: 4
Ingrese el título de la enciclopedia: Britannica
Ingrese el autor de la enciclopedia: Encyclopaedia Britannica
Ingrese la fecha de publicación: 1974
Ingrese el valor de la enciclopedia $: 39.99
Enciclopedia agregada.
```

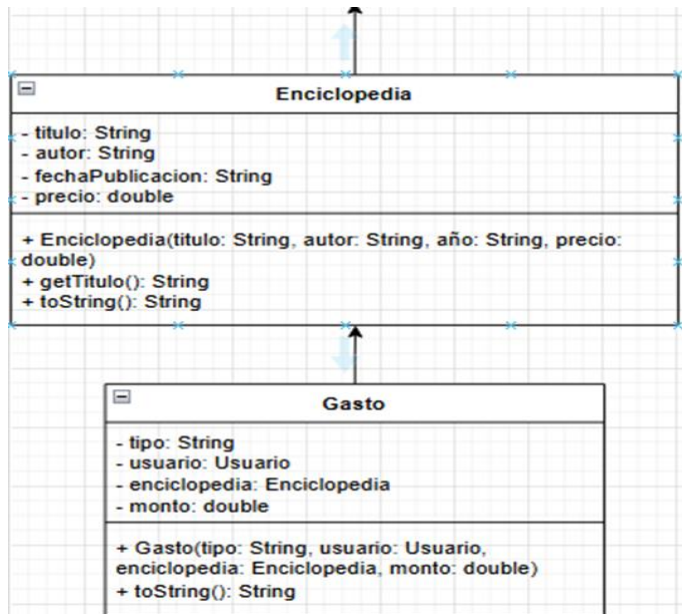
```
--- Menú Bibliotecario ---
1. Agregar Usuario
2. Eliminar Usuario
3. Mostrar Usuarios
4. Agregar Enciclopedia
5. Eliminar Enciclopedia
6. Mostrar Enciclopedias
0. Salir
Selecciona una opción: 6
Enciclopedias disponibles:
Enciclopedia: Britannica, Autor: Encyclopaedia Britannica, Fecha de Publicación: 1974, Valor: 39.99
```

```
--- Menú Bibliotecario ---
1. Agregar Usuario
2. Eliminar Usuario
3. Mostrar Usuarios
4. Agregar Enciclopedia
5. Eliminar Enciclopedia
6. Mostrar Enciclopedias
0. Salir
Selecciona una opción: 5
Ingrese el título de la enciclopedia a eliminar: Britannica
Enciclopedia eliminada.
```

- **Gastos:** Registra compras y regalos. Se relaciona con **Usuario** y **Enciclopedia** por agregación

DIAGRAMA UML:





Explicacion

Clase Bibliotecario:

- **Atributos:**

nombre: Nombre del bibliotecario.

usuarios: Lista de objetos Usuario.

enciclopedias: Lista de objetos Enciclopedia.

gastos: Lista de objetos Gasto.

- **Métodos:**

Gestión de usuarios: agregar Usuario, eliminar Usuario.

Gestión de enciclopedias: agregar Enciclopedia, eliminar Enciclopedia.

Gestión de gastos: registrarGasto.

Mostrar datos: mostrarUsuarios, mostrarEnciclopedias, mostrarGastos.

Getters: Obtienen datos de atributos.

Clase Usuario:

- **Atributo:**

nombre: Nombre del usuario.

- **Métodos:**

Constructor: Inicializa el nombre.

Getters y toString: Acceso y representación de datos.

Clase Enciclopedia:

- **Atributos:**

título: Título de la enciclopedia.

autor: Autor de la enciclopedia.

fechaPublicacion: Fecha de publicación.

precio: Precio de la enciclopedia.

- **Métodos:**

Constructor: Inicializa los atributos.

getTitulo: Devuelve el título.

toString: Proporciona una representación en texto de los datos.

Clase Gasto:

- **Atributos:**

tipo: Tipo de gasto (compra, donación, etc.).

usuario: Relación con un objeto Usuario.

enciclopedia: Relación con un objeto Enciclopedia.

monto: Monto del gasto.

- **Métodos:**

Constructor: Inicializa el tipo, usuario, enciclopedia y monto.

toString: Representación textual del gasto.

CÓDIGO:

```
Main.java | Enciclopedia.java | Bibliotecario.java | Usuario.java | Gasto.java
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         Bibliotecario bibliotecario = new Bibliotecario("Luis Jaramillo");
7
8         // Agregar algunos usuarios y enciclopedias
9         Usuario usuario1 = new Usuario("Samir");
10        Usuario usuario2 = new Usuario("Steven");
11        bibliotecario.agregarUsuario(usuario1);
12        bibliotecario.agregarUsuario(usuario2);
13
14        Enciclopedia enciclopedia1 = new Enciclopedia("Enciclopedia Britannica", "Encyclopaedia Britannica", "1974", 39.99);
15        Enciclopedia enciclopedia2 = new Enciclopedia("Enciclopedia Moderna", "John Doe", "1988", 29.99);
16        bibliotecario.agregarEnciclopedia(enciclopedia1);
17        bibliotecario.agregarEnciclopedia(enciclopedia2);
18
19        // Menú de opciones
20        boolean continuar = true;
21        while (continuar) {
22            System.out.println("\nMenú:");
23            System.out.println("1. Mostrar usuarios");
24            System.out.println("2. Mostrar enciclopedias");
25            System.out.println("3. Registrar gasto");
26            System.out.println("4. Mostrar gastos");
27            System.out.println("5. Salir");
28            System.out.print("Seleccione una opción: ");
29            int opcion = scanner.nextInt();
30            scanner.nextLine(); // Limpiar el buffer
```

Explicación

Clase principal (Main): Contiene el método main que inicia el programa.

Objetos creados:

- Un objeto de tipo **Bibliotecario** llamado **bibliotecario**.
- Dos objetos de tipo **Usuario** (**usuario1** y **usuario2**).
- Dos objetos de tipo **Enciclopedia** (**enciclopedia1** y **enciclopedia2**).

Métodos:

- Los usuarios y enciclopedias se agregan al **bibliotecario** mediante métodos como **agregarUsuario** y **agregarEnciclopedia**.

Menú interactivo:

- Usa un bucle **while** para mostrar opciones al usuario, como mostrar usuarios, enciclopedias, registrar gastos, y salir.

```

31
32 -      switch (opcion) {
33 -          case 1:
34 -              bibliotecario.mostrarUsuarios();
35 -              break;
36 -          case 2:
37 -              bibliotecario.mostrarEnciclopedias();
38 -              break;
39 -          case 3:
40 -              System.out.print("Ingrese el tipo de gasto (compra/regalo): ");
41 -              String tipo = scanner.nextLine();
42 -              System.out.print("Ingrese el nombre del usuario: ");
43 -              String nombreUsuario = scanner.nextLine();
44 -              Usuario usuario = null;
45 -
46 -              // Buscar el usuario
47 -              for (Usuario u : bibliotecario.getUsuarios()) {
48 -                  if (u.getNombre().equalsIgnoreCase(nombreUsuario)) {
49 -                      usuario = u;
50 -                      break;
51 -                  }
52 -              }
53 -
54 -              if (usuario == null) {
55 -                  System.out.println("Usuario no encontrado.");
56 -                  break;
57 -              }
58 -
59 -              System.out.print("Ingrese el título de la enciclopedia: ");
60 -              String tituloEnciclopedia = scanner.nextLine();
61 -              Enciclopedia enciclopedia = null;
62 -
63 -              // Buscar la enciclopedia
64 -              for (Enciclopedia e : bibliotecario.getEnciclopedias()) {
65 -                  if (e.getTitulo().equalsIgnoreCase(tituloEnciclopedia)) {
66 -                      enciclopedia = e;
67 -                      break;
68 -                  }
69 -              }
70 -      }

```

Explicación

En esta sección del código, se utiliza un bloque **switch** para manejar las opciones del menú interactivo. Aquí está un resumen:

1. **case 1:** Llama al método `mostrarUsuarios()` del objeto `bibliotecario` para listar todos los usuarios registrados.
2. **case 2:** Llama al método `mostrarEnciclopedias()` del objeto `bibliotecario` para listar todas las enciclopedias disponibles.
3. **case 3:** Realiza las siguientes acciones:

- Solicita al usuario que ingrese el tipo de gasto (**compra o regalo**) y el nombre del usuario.
- Busca al usuario en la lista de usuarios usando un bucle **for**. Si no lo encuentra, muestra un mensaje indicando que no existe.
- Solicita el título de la enciclopedia y la busca en la lista de enciclopedias. Si no la encuentra, el valor de la variable correspondiente permanece en **null**.

```

71 -         if (enciclopedia == null) {
72             System.out.println("Enciclopedia no encontrada.");
73             break;
74         }
75
76         System.out.print("Ingrese el monto del gasto: ");
77         double monto = scanner.nextDouble();
78         Gasto gasto = new Gasto(tipo, usuario, enciclopedia, monto);
79         bibliotecario.registrarGasto(gasto);
80         System.out.println("Gasto registrado exitosamente.");
81         break;
82     case 4:
83         bibliotecario.mostrarGastos();
84         break;
85     case 5:
86         continuar = false;
87         System.out.println("Saliendo del sistema...");
88         break;
89     default:
90         System.out.println("Opción no válida. Intente de nuevo.");
91     }
92 }
93
94 scanner.close();
95 }
96 }

```

Explicación

Validación de la enciclopedia (case 3):

- Si no se encuentra la enciclopedia (**enciclopedia == null**), se muestra un mensaje: "Enciclopedia no encontrada".
- Si la enciclopedia existe:

Solicita el monto del gasto.

Crea un nuevo objeto de tipo **Gasto** con la información ingresada (tipo, usuario, enciclopedia, monto).

Llama al método **registrarGasto** del **bibliotecario** para almacenar el gasto.

Informa al usuario que el gasto fue registrado exitosamente.

Mostrar gastos (case 4):

- Llama al método **mostrarGastos()** del objeto **bibliotecario** para listar los gastos registrados.

Salir del sistema (case 5):

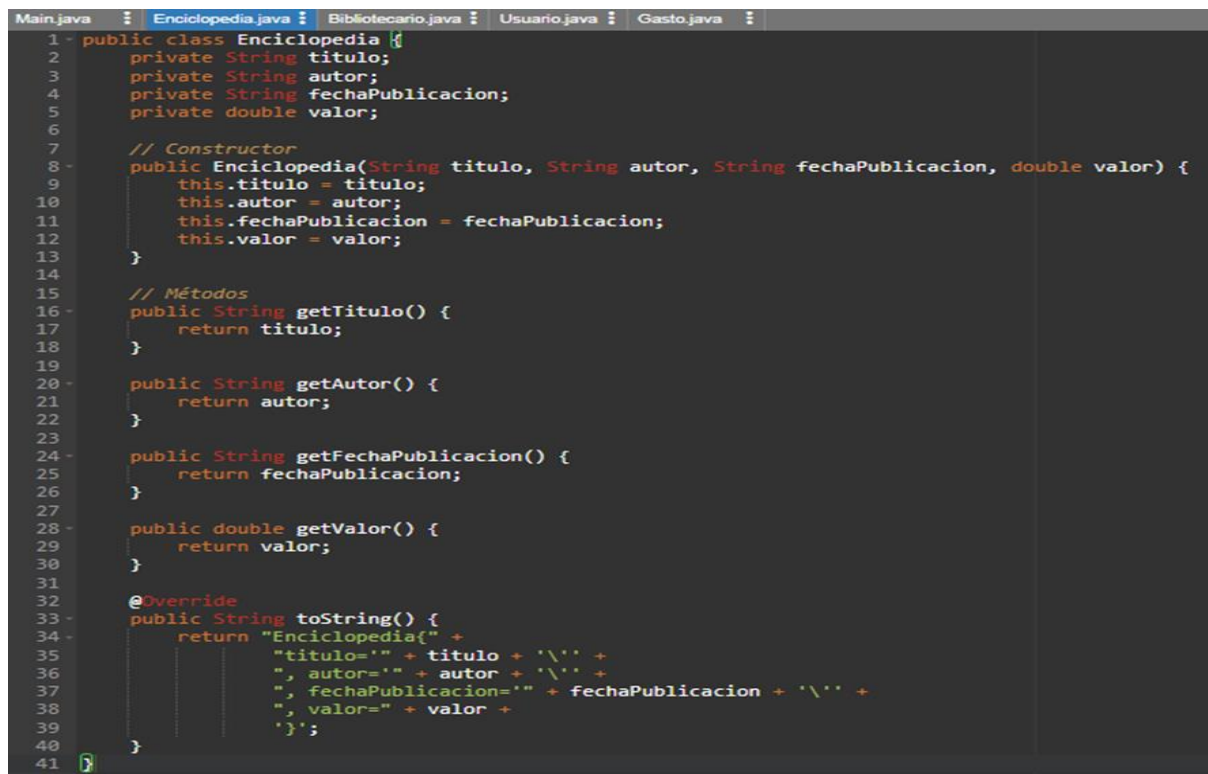
- Cambia la variable **continuar** a **false** para salir del bucle del menú.
- Muestra el mensaje: "Saliendo del sistema...".

Opción por defecto (default):

- Si el usuario ingresa una opción no válida, muestra el mensaje: "Opción no válida. Intente de nuevo."

Cierre del recurso **Scanner**:

- Al final del programa, el recurso **scanner** se cierra con **scanner.close()** para liberar memoria.



```
1 public class Enciclopedia {
2     private String titulo;
3     private String autor;
4     private String fechaPublicacion;
5     private double valor;
6
7     // Constructor
8     public Enciclopedia(String titulo, String autor, String fechaPublicacion, double valor) {
9         this.titulo = titulo;
10        this.autor = autor;
11        this.fechaPublicacion = fechaPublicacion;
12        this.valor = valor;
13    }
14
15    // Métodos
16    public String getTitulo() {
17        return titulo;
18    }
19
20    public String getAutor() {
21        return autor;
22    }
23
24    public String getFechaPublicacion() {
25        return fechaPublicacion;
26    }
27
28    public double getValor() {
29        return valor;
30    }
31
32    @Override
33    public String toString() {
34        return "Enciclopedia{" +
35            "titulo='" + titulo + '\'' +
36            ", autor='" + autor + '\'' +
37            ", fechaPublicacion='" + fechaPublicacion + '\'' +
38            ", valor=" + valor +
39            '}';
40    }
41 }
```

Explicación

Atributos

- **titulo**: Almacena el título de la enciclopedia.
- **autor**: Representa el nombre del autor.
- **fechaPublicacion**: Fecha en que se publicó la enciclopedia.
- **valor**: Valor monetario asociado con la enciclopedia.

2. Constructor

- Permite inicializar una instancia de **Enciclopedia** con los datos del título, autor, fecha de publicación y valor.
- Usa **this** para asignar los valores de los parámetros a los atributos de la clase.

3. Métodos **getter**

- Cada atributo tiene un método **getter** que devuelve su valor:
- **getTitulo()**: Devuelve el título.
- **getAutor()**: Devuelve el autor.
- **getFechaPublicacion()**: Devuelve la fecha de publicación.
- **getValor()**: Devuelve el valor monetario.

4. Sobrecarga de **toString**

- Se sobrescribe el método **toString** para devolver una representación en texto del objeto **Enciclopedia**.
- Incluye todos los atributos formateados en una estructura legible.

```

Main.java : Enciclopedia.java : Bibliotecario.java : Usuario.java : Gasto.java :
1  import java.util.ArrayList;
2  import java.util.List;
3
4  public class Bibliotecario {
5      private String nombre;
6      private List<Usuario> usuarios;
7      private List<Enciclopedia> enciclopedias;
8      private List<Gasto> gastos;
9
10     // Constructor
11     public Bibliotecario(String nombre) {
12         this.nombre = nombre;
13         this.usuarios = new ArrayList<>();
14         this.enciclopedias = new ArrayList<>();
15         this.gastos = new ArrayList<>();
16     }
17
18     // Métodos para agregar usuarios y enciclopedias
19     public void agregarUsuario(Usuario usuario) {
20         usuarios.add(usuario);
21     }
22
23     public void eliminarUsuario(Usuario usuario) {
24         usuarios.remove(usuario);
25     }
26
27     public void agregarEnciclopedia(Enciclopedia enciclopedia) {
28         enciclopedias.add(enciclopedia);
29     }
30
31     public void eliminarEnciclopedia(Enciclopedia enciclopedia) {
32         enciclopedias.remove(enciclopedia);
33     }
34
35     public void registrarGasto(Gasto gasto) {
36         gastos.add(gasto);
37     }

```

Explicación

1. Atributos

- **nombre:** Nombre del bibliotecario.
- **usuarios:** Lista de usuarios registrados (tipo List<Usuario>).
- **enciclopedias:** Lista de enciclopedias disponibles (tipo List<Enciclopedia>).
- **gastos:** Lista de gastos registrados (tipo List<Gasto>).

2. Constructor

- Inicializa el nombre del bibliotecario y crea instancias vacías de las listas para usuarios, enciclopedias y gastos usando ArrayList.

3. Métodos

Para manejar usuarios:

- **agregarUsuario(Usuario usuario):**

Agrega un usuario a la lista de usuarios usando el método add.

- **eliminarUsuario(Usuario usuario):**

Elimina un usuario de la lista de usuarios usando el método remove.

Para manejar enciclopedias:

- **agregarEnciclopedia(Enciclopedia enciclopedia):**

Agrega una enciclopedia a la lista de enciclopedias.

- **eliminarEnciclopedia(Enciclopedia enciclopedia):**

Elimina una enciclopedia de la lista enciclopedias.

Para manejar gastos:

- **registrarGasto(Gasto gasto):**

Agrega un gasto a la lista gastos.

```

38
39 - public void mostrarUsuarios() {
40     System.out.println("Usuarios:");
41     for (Usuario usuario : usuarios) {
42         System.out.println(usuario);
43     }
44 }
45
46 - public void mostrarEnciclopedias() {
47     System.out.println("Enciclopedias:");
48     for (Enciclopedia enciclopedia : enciclopedias) {
49         System.out.println(enciclopedia);
50     }
51 }
52
53 - public void mostrarGastos() {
54     System.out.println("Gastos:");
55     for (Gasto gasto : gastos) {
56         System.out.println(gasto);
57     }
58 }
59
60 - public String getNombre() {
61     return nombre;
62 }
63
64 // Métodos para acceder a las listas de usuarios y enciclopedias
65 - public List<Usuario> getUsuarios() {
66     return usuarios;
67 }
68
69 - public List<Enciclopedia> getEnciclopedias() {
70     return enciclopedias;
71 }
72

```

Explicacion

Métodos para mostrar datos:

- **mostrarUsuarios():** Itera y muestra todos los usuarios almacenados en una lista llamada usuarios.
- **mostrarEnciclopedias():** Itera y muestra todas las enciclopedias almacenadas en una lista llamada enciclopedias.
- **mostrarGastos():** Itera y muestra todos los gastos almacenados en una lista llamada gastos.

Métodos de acceso:

- **getNombre():** Devuelve el nombre asociado a esta clase como un String.
- **getUsuarios():** Devuelve la lista de usuarios (List<Usuario>).
- **get Enciclopedias():** Devuelve la lista de enciclopedias (List<Enciclopedia>).


```

Main.java  Enciclopedia.java  Bibliotecario.java  Usuario.java  Gasto.java
1  public class Usuario {
2      private String nombre;
3
4      // Constructor
5      public Usuario(String nombre) {
6          this.nombre = nombre;
7      }
8
9      // Método para obtener el nombre
10     public String getNombre() {
11         return nombre;
12     }
13
14     @Override
15     public String toString() {
16         return "Usuario{" +
17             "nombre='" + nombre + '\'' +
18             '}';
19     }
20 }

```

Explicación

Atributo privado:

- **private String nombre;** Almacena el nombre del usuario.

Constructor:

- **public Usuario(String nombre):** Permite crear un objeto de la clase Usuario inicializando el atributo nombre con el valor proporcionado.

Método getter:

- **public String getNombre():** Devuelve el valor del atributo nombre.

Sobrescritura del método toString():

- Proporciona una representación en texto del objeto Usuario que incluye el atributo nombre en el formato:

Usuario{nombre='valor_del_nombre'}

```

1 public class Gasto {
2     private String tipo;
3     private Usuario usuario;
4     private Enciclopedia enciclopedia;
5     private double monto;
6
7     // Constructor
8     public Gasto(String tipo, Usuario usuario, Enciclopedia enciclopedia, double monto) {
9         this.tipo = tipo;
10        this.usuario = usuario;
11        this.enciclopedia = enciclopedia;
12        this.monto = monto;
13    }
14
15    @Override
16    public String toString() {
17        return "Gasto{" +
18            "tipo='" + tipo + '\'' +
19            ", usuario=" + usuario.getNombre() +
20            ", enciclopedia=" + enciclopedia.getTitulo() +
21            ", monto=" + monto +
22            '}';
23    }
24 }

```

Explicación

Atributos:

- **tipo:** Tipo de gasto (ej. "compra", "préstamo").
- **usuario:** El usuario relacionado con el gasto (objeto Usuario).
- **enciclopedia:** La enciclopedia asociada al gasto (objeto Enciclopedia).
- **monto:** El monto del gasto (tipo double).

Constructor:

- Inicializa todos los atributos (tipo, usuario, enciclopedia, monto) al crear un objeto.

Método toString():

- Sobrescribe el método para mostrar la información del gasto en un formato legible, incluyendo:

El nombre del usuario (usuario.getNombre()).

El título de la enciclopedia (enciclopedia.getTitulo()).

El monto del gasto.

Ejecucion


```
Menú:
1. Mostrar usuarios
2. Mostrar enciclopedias
3. Registrar gasto
4. Mostrar gastos
5. Salir
Seleccione una opción: 1
Usuarios:
Usuario(nombre='Samir')
Usuario(nombre='Steven')

Menú:
1. Mostrar usuarios
2. Mostrar enciclopedias
3. Registrar gasto
4. Mostrar gastos
5. Salir
Seleccione una opción: 2
Enciclopedias:
Enciclopedia[titulo='Enciclopedia Britannica', autor='Encyclopaedia Britannica', fechaPublicacion='1974', valor=39.99]
Enciclopedia[titulo='Enciclopedia Moderna', autor='John Doe', fechaPublicacion='1980', valor=29.99]
```

```
Menú:
1. Mostrar usuarios
2. Mostrar enciclopedias
3. Registrar gasto
4. Mostrar gastos
5. Salir
Seleccione una opción: 3
Ingrese el tipo de gasto (compra/regalo): 39.99
Ingrese el nombre del usuario: Samir
Ingrese el título de la enciclopedia: Enciclopedia Britannica
Ingrese el monto del gasto: 39.99
Gasto registrado exitosamente.
```

```
Menú:
1. Mostrar usuarios
2. Mostrar enciclopedias
3. Registrar gasto
4. Mostrar gastos
5. Salir
Seleccione una opción: 4
Gastos:
Gasto[tipo='39.99', usuario=Samir, enciclopedia=Enciclopedia Britannica, monto=39.99]
```

```
Menú:
1. Mostrar usuarios
2. Mostrar enciclopedias
3. Registrar gasto
4. Mostrar gastos
5. Salir
Seleccione una opción: 5
Saliendo del sistema...
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

4. CONCLUSIONES

1. El uso de diagramas UML promueve una representación clara y precisa de los objetos y sus relaciones, lo que permite una mejor comprensión del sistema en su conjunto, lo cual es importante en entornos académicos y profesionales.
2. La correcta definición de asociaciones, composiciones y agregados en UML asegura una interacción efectiva entre objetos, optimización de la funcionalidad y escalabilidad del sistema diseñado.
3. La estructuración adecuada de los objetos y sus interacciones mejora la capacidad de mantenimiento, la reutilización del código y las mejoras futuras, lo que le ayuda a desarrollar sistemas fiables y flexibles.

5. RECOMENDACIONES

1. Al diseñar objetos, es importante seguir principios como encapsulación, herencia y polimorfismo, asegurando que el sistema sea modular y flexible, permitiendo una mayor extensión o modificación.
2. Recomendamos utilizar herramientas como Draw.io para crear diagramas claros y profesionales que puedan integrarse fácilmente con documentos técnicos e informes académicos.
3. Antes de implementar un sistema, es de extrema importancia revisar el diagrama para asegurarse de que las relaciones entre los objetos estén correctamente definidas y que cumplan con los requisitos funcionales y no funcionales del proyecto.

6. BIBLIOGRAFÍA

- Doe, J. (2020). Investigador en ciencias sociales con especialización en estudios culturales y antropología. Universidad de Cambridge.
- Smith, A. B. (2018). Experta en neurociencia enfocada en plasticidad cerebral. Universidad de Harvard.
- González, P. (2022). Historiador especializado en historia latinoamericana. Instituto Nacional de Historia de México.
- Lee, C. H. (2019). Ingeniera informática especializada en inteligencia artificial.
- Brown, R. T. (2021). Investigador en sostenibilidad ambiental y cambio climático. Universidad de Stanford.