

Nuevo Ejercicio Propuesto 2: Biblioteca y Libros

1. Descripción general

Se debe desarrollar un sistema para gestionar una biblioteca, en la cual se registren los libros disponibles y sus autores. La relación central es de composición 1 a N: una Biblioteca contiene múltiples Libros, y cada Libro pertenece obligatoriamente a una Biblioteca. Si la Biblioteca se elimina, también se eliminan sus Libros.

2. Clases a implementar

Clase Autor

Atributos:

- id (String) → Identificador único del autor.
- nombre (String) → Nombre del autor.
- nacionalidad (String) → Nacionalidad del autor.

Métodos:

- mostrarInfo() → Muestra la información del autor en consola.

Clase Libro

Atributos:

- isbn (String) → Identificador único del libro.
- titulo (String) → Título del libro.
- anioPublicacion (int) → Año de publicación.
- autor (Autor) → Autor del libro.

Métodos:

- mostrarInfo() → Muestra título, ISBN, año y autor.

Clase Biblioteca

Atributo:

- String nombre
- List<Libro> libros → Colección de libros de la biblioteca.

Métodos requeridos:

- agregarLibro(String isbn, String titulo,int anioPublicacion, Autor autor)
- listarLibros()
- buscarLibroPorIsbn(String isbn)

- eliminarLibro(String isbn)
- obtenerCantidadLibros()
- filtrarLibrosPorAnio(int anio)
- mostrarAutoresDisponibles()

3. Tareas a realizar

1. Creamos una biblioteca.

```
import java.util.List;
public class Principal {
    public static void main(String[] args) {
        // Creamos una biblioteca
        Biblioteca biblioteca=new Biblioteca("Biblioteca Municipal");
```

2. Crear al menos tres autores

```
//Creamos Autores
Autor laura = new Autor("1","Laura Mendez","Argentina");
Autor diego = new Autor("2","Diego Salazar","Mexico");
Autor sofia = new Autor("3","Sofia Pereira","Uruguay");
```

3. Agregar 5 libros asociados a alguno de los Autores a la biblioteca.

```
//Agregamos los libros por composicion
biblioteca.agregarLibro("A234","Logica de Programacion Moderna", 2021, laura);
biblioteca.agregarLibro("H345", "Sistemas Distribuidos Esenciales", 2023, diego);
biblioteca.agregarLibro("S234", "Arquitectura de Software", 2021, laura );
biblioteca.agregarLibro("C234", "Patron de diseño con Java", 2020, sofia );
biblioteca.agregarLibro("F234", "Estructura de Datos en Profundidad", 2024, diego);
```

4. Listar todos los libros con su información y la del autor.

```
//Listamos los libros de la biblioteca
System.out.println("Los libros de la biblioteca "+biblioteca.getNombre()+"son: ");
biblioteca.listarLibros();
System.out.println("-----");
```

```
run:
Los libros de la biblioteca Biblioteca Municipalson:
La lista de libros de la bibliotecaBiblioteca Municipal es:
Libro {isbn: A234
      titulo: Logica de Programacion Moderna
      anio de Publicacion: 2021
      Autor{id: 1
            nombre: Laura Mendez
            nacionalidad: Argentina}}
Libro {isbn: H345
      titulo: Sistemas Distribuidos Esenciales
      anio de Publicacion: 2023
      Autor{id: 2
            nombre: Diego Salazar
            nacionalidad: Mexico}}
Libro {isbn: S234
      titulo: Arquitectura de Software
      anio de Publicacion: 2021
      Autor{id: 1
            nombre: Laura Mendez
            nacionalidad: Argentina}}
Libro {isbn: C234
      titulo: Patron de diseño con Java
      anio de Publicacion: 2020
      Autor{id: 3
            nombre: Sofia Pereira
            nacionalidad: Uruguay}}
Libro {isbn: F234
      titulo: Estructura de Datos en Profundidad
      anio de Publicacion: 2024
      Autor{id: 2
            nombre: Diego Salazar
            nacionalidad: Mexico})
```

5. Buscar un libro por su ISBN y mostrar su información.

```
//buscamos un libro por ISBN
System.out.println("Buscamos un libro por isbn A234");
biblioteca.buscarLibroPorIsbn("A234").mostrarInfo();
System.out.println(".....");
```

```
Buscamos un libro por isbn A234
Libro {isbn: A234
      titulo: Logica de Programacion Moderna
      anio de Publicacion: 2021
      Autor{id: 1
            nombre: Laura Mendez
            nacionalidad: Argentina})
```

6. Filtrar y mostrar los libros publicados en un año específico.

```
//buscamos los libros del mismo año de publicación
int añoFiltro = 2021;
List<Libro> librosPorAño = biblioteca.filtrarLibrosPorAño(añoFiltro);

System.out.println("Los libros encontrados del año "+añoFiltro+" son: ");
for(Libro libro:librosPorAño){
    libro.mostrarInfo();
}
System.out.println("///////////////////////////////");
```

.....

```
Los libros encontrados del año 2021 son:
Libro {isbn: A234
titulo: Logica de Programacion Moderna
año de Publicacion: 2021
Autor{id: 1
nombre: Laura Mendez
nacionalidad: Argentina}}
Libro {isbn: S234
titulo: Arquitectura de Software
año de Publicacion: 2021
Autor{id: 1
nombre: Laura Mendez
nacionalidad: Argentina}}
```

7. Eliminar un libro por su ISBN y listar los libros restantes.

```
//eliminamos un libro por isbn
biblioteca.eliminarLibro("S234");

//Listamos los libros después de eliminar uno por isbn
System.out.println("Imprimimos los libros después de eliminar uno");
biblioteca.listarLibros();
```

```
Imprimimos los libros después de eliminar uno
La lista de libros de la bibliotecaBiblioteca Municipal es:
Libro {isbn: A234
titulo: Logica de Programacion Moderna
año de Publicacion: 2021
Autor{id: 1
nombre: Laura Mendez
nacionalidad: Argentina}}
Libro {isbn: H345
titulo: Sistemas Distribuidos Esenciales
año de Publicacion: 2023
Autor{id: 2
nombre: Diego Salazar
nacionalidad: Mexico}}
Libro {isbn: C234
titulo: Patron de diseño con Java
año de Publicacion: 2020
Autor{id: 3
nombre: Sofia Pereira
nacionalidad: Uruguay}}
Libro {isbn: F234
titulo: Estructura de Datos en Profundidad
año de Publicacion: 2024
Autor{id: 2
nombre: Diego Salazar
nacionalidad: Mexico}}
```

8. Mostrar la cantidad total de libros en la biblioteca.

```
//Imprimimos la cantidad total de libros en la biblioteca
System.out.println("La cantidad total de libros en la biblioteca " +
    biblioteca.getNombre()+"son: "+biblioteca.obtenerCantidadLibros());
```

```
La cantidad total de libros en la biblioteca Biblioteca Municipalson: 4
```

9. Listar todos los autores de los libros disponibles en la biblioteca.

```
//Listamos los autores de los libros de la biblioteca.
System.out.println("Los autores de los libros de la biblioteca son: ");
biblioteca.mostrarAutoresDisponibles();
}
```

```
Los autores de los libros de la biblioteca son:
Autor{id: 1
nombre: Laura Mendez
nacionalidad: Argentina}
Autor{id: 2
nombre: Diego Salazar
nacionalidad: Mexico}
Autor{id: 3
nombre: Sofia Pereira
nacionalidad: Uruguay}
Autor{id: 2
nombre: Diego Salazar
nacionalidad: Mexico}
```

Ejercicio: Universidad, Profesor y Curso (bidireccional 1 a N)

1. Descripción general

Se debe modelar un sistema académico donde un Profesor dicta muchos Cursos y cada Curso tiene exactamente un Profesor responsable. La relación Profesor Curso es bidireccional:

- Desde Curso se accede a su Profesor.
- Desde Profesor se accede a la lista de Cursos que dicta.

Además, existe la clase Universidad que administra el alta/baja y consulta de profesores y cursos.

Invariante de asociación: cada vez que se asigne o cambie el profesor de un curso,

debe actualizarse en los dos lados (agregar/quitar en la lista del profesor correspondiente).

2. Clases a implementar

Clase Profesor

Atributos:

- id (String) → Identificador único.
- nombre (String) → Nombre completo.

- especialidad (String) → Área principal.
- List<Curso> cursos → Cursos que dicta.

Métodos sugeridos:

- agregarCurso(Curso c) → Agrega el curso a su lista si no está y sincroniza el lado del curso.
- eliminarCurso(Curso c) → Quita el curso y sincroniza el lado del curso (dejar profesor en null si corresponde).
- listarCursos() → Muestra códigos y nombres.
- mostrarInfo() → Imprime datos del profesor y cantidad de cursos.

Clase Curso

Atributos:

- codigo (String) → Código único.
- nombre (String) → Nombre del curso.
- profesor (Profesor) → Profesor responsable.

Métodos sugeridos:

- setProfesor(Profesor p) → Asigna/cambia el profesor sincronizando ambos lados:
 - Si tenía profesor previo, quitarse de su lista.
- mostrarInfo() → Muestra código, nombre y nombre del profesor (si tiene).

Clase Universidad

Atributos:

- String nombre
- List<Profesor> profesores
- List<Curso> cursos

Métodos requeridos:

- agregarProfesor(Profesor p)
- agregarCurso(Curso c)
- asignarProfesorACurso(String codigoCurso, String idProfesor) → Usa setProfesor del curso.
- listarProfesores() / listarCursos()
- buscarProfesorPorId(String id)
- buscarCursoPorCodigo(String codigo)
- eliminarCurso(String codigo) → Debe romper la relación con su profesor si la hubiera.
- eliminarProfesor(String id) → Antes de remover, dejar null los cursos que dictaba.

Tareas por realizar

1. Crear al menos 3 profesores y 5 cursos.

```

package Ejercicio_3;
public class Principal {

    public static void main(String[] args) {
        // Instanciamos 3 profesores
        Profesor profesor1 = new Profesor("1","Leandro De Angelis","ADO");
        Profesor profesor2 = new Profesor("2","Ariel Enferrel","Programacion");
        Profesor profesor3 = new Profesor("3","Carlos Frede","Financiera");

        //Instanciamos 5 cursos
        Curso programacion1= new Curso("1","Programacion1");
        Curso ado = new Curso("2","ADO");
        Curso financiera = new Curso("3","Financiera 1");
        Curso programacion2 = new Curso("4","Programacion 2");
        Curso baseDatos = new Curso("5","Base de Datos");
    }
}

```

2. Agregar profesores y cursos a la universidad.

```

//Agregamos los profesores a la universidad
uni.agregarProfesor(profesor3);
uni.agregarProfesor(profesor2);
uni.agregarProfesor(profesor1);

//Agregamos los cursos
uni.agregarCurso(ado);
uni.agregarCurso(programacion1);
uni.agregarCurso(financiera);
uni.agregarCurso(programacion2);
uni.agregarCurso(baseDatos);

```

3. Asignar profesores a cursos usando asignarProfesorACurso(...).

```

//Asignamos los profesores con los cursos
uni.asignarProfesorACurso("2", "1");
uni.asignarProfesorACurso("1", "2");
uni.asignarProfesorACurso("4", "2");
uni.asignarProfesorACurso("5", "2");
uni.asignarProfesorACurso("3", "3");

```

4. Listar cursos con su profesor y profesores con sus cursos.

```

//Listamos profesores con cursos
uni.listarProfesores();

//Listamos cursos
uni.listarCursos();

```

```

La unidversidad UTN cuenta con 5 cursos y son:
El codigo del curso es: 2
el nombre del curso es: ADO
el profesor asignado al curso es: Leandro De Angelis
El codigo del curso es: 1
el nombre del curso es: Programacion1
el profesor asignado al curso es: Ariel Enferrel
El codigo del curso es: 3
el nombre del curso es: Financiera 1
el profesor asignado al curso es: Carlos Frede
El codigo del curso es: 4
el nombre del curso es: Programacion 2
el profesor asignado al curso es: Ariel Enferrel
El codigo del curso es: 5
el nombre del curso es: Base de Datos
el profesor asignado al curso es: Ariel Enferrel

```

5. Cambiar el profesor de un curso y verificar que ambos lados quedan sincronizados.

```
//Cambiemos un profesor de curso
System.out.println("||||||||||||||||||||||||||||");
System.out.println("Reasignamos el curso de Programacion 1 al docente"
+ "Leandro De Angelis");
uni.asignarProfesorACurso("1", "1");
uni.listarCursos();
uni.listarProfesor();

Reasignamos el curso de Programacion 1 al docenteLeandro De Angelis
La unidversidad UTN cuenta con 5 cursos y son:
El codigo del curso es: 2
el nombre del curso es: ADO
el profesor asignado al curso es: Leandro De Angelis
El codigo del curso es: 1
el nombre del curso es: Programacion1
el profesor asignado al curso es: Leandro De Angelis
El codigo del curso es: 3
el nombre del curso es: Financiera 1
el profesor asignado al curso es: Carlos Frede
El codigo del curso es: 4
el nombre del curso es: Programacion 2
el profesor asignado al curso es: Ariel Enferrel
El codigo del curso es: 5
el nombre del curso es: Base de Datos
el profesor asignado al curso es: Ariel Enferrel
Los profsores de la universidad UTN cuenta con 3 docentes:
El profesor con id: 3
nombre: Carlos Frede
especialidad: Financiera
ejerse en 1 cursos.
```

6. Remover un curso y confirmar que ya no aparece en la lista del profesor.

```
//Removemos un curso y validamos que no lo tiene al profesor
System.out.println("*****");
System.out.println(" Removemos el curso de id=2 y vemos los profesores");
uni.eliminarCurso(financiera);
uni.listarProfesor();
//Vemos que ahora hay 1 curso menos en la universidad
uni.listarCursos();
```

```

Removemos el curso de id=2 y vemos los profesores
Los profsores de la universidad UTN cuenta con 3 docentes:
El profesor con id: 3
nombre: Carlos Frede
especialidad: Financiera
ejerce en 0 cursos.
-----
El profesor con id: 2
nombre: Ariel Enferrel
especialidad: Programacion
ejerce en 2 cursos.
-----
El profesor con id: 1
nombre: Leandro De Angelis
especialidad: ADO
ejerce en 2 cursos.
-----
La unidversidad UTN cuenta con 4 cursos y son:
El codigo del curso es: 2
el nombre del curso es: ADO
el profesor asignado al curso es: Leandro De Angelis
El codigo del curso es: 1
el nombre del curso es: Programacionl
el profesor asignado al curso es: Leandro De Angelis
El codigo del curso es: 4
el nombre del curso es: Programacion 2
el profesor asignado al curso es: Ariel Enferrel
El codigo del curso es: 5
el nombre del curso es: Base de Datos
el profesor asignado al curso es: Ariel Enferrel

```

7. Remover un profesor y dejar profesor = null,

```

//Removemos un profesor
System.out.println("||||||||||||||||||||||||");
System.out.println(" Removemos al profesor de id=1");
uni.eliminarProfesor(profesor1);

uni.listarProfesor();
}

```

8. Mostrar un reporte: cantidad de cursos por profesor.

```

uni.listarProfesor();
}

Removemos al profesor de id=1
Los profsores de la universidad UTN cuenta con 2 docentes
El profesor con id: 3
nombre: Carlos Frede
especialidad: Financiera
ejerce en 0 cursos.
-----
El profesor con id: 2
nombre: Ariel Enferrel
especialidad: Programacion
ejerce en 2 cursos.
-----
```