

PROGRAMACIÓN II

Trabajo Práctico 6: Colecciones y sistema de stock

Alumno :

- González Federico

Profesor :

- Enferrel Ariel

Tutor :

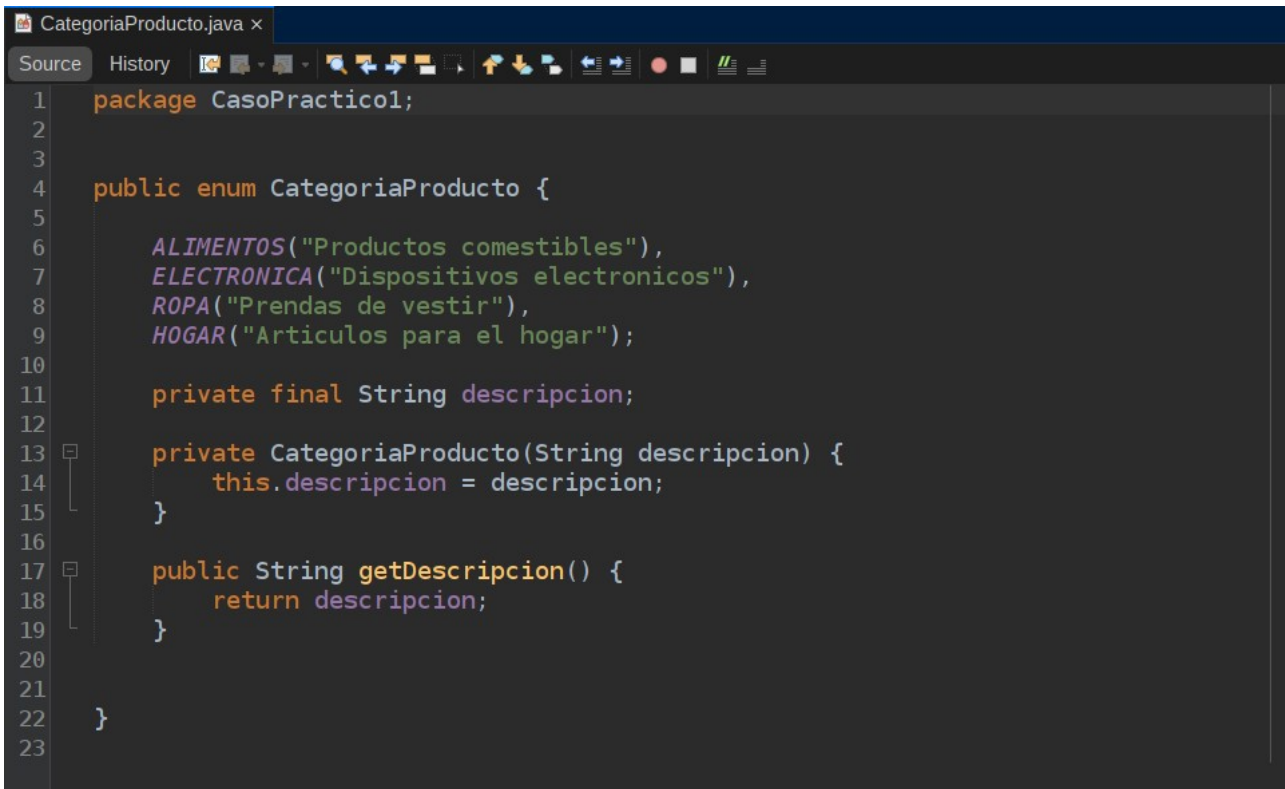
- Ferro Tomás

GitHub:

- <https://github.com/fedeglz/UTN-TUPaD-P2.git>

Caso practico 1:

1. Crear al menos cinco productos con diferentes categorías y agregarlos al inventario.
2. Listar todos los productos mostrando su información y categoría.
3. Buscar un producto por ID y mostrar su información.
4. Filtrar y mostrar productos que pertenezcan a una categoría específica.
5. Eliminar un producto por su ID y listar los productos restantes.
6. Actualizar el stock de un producto existente.
7. Mostrar el total de stock disponible.
8. Obtener y mostrar el producto con mayor stock.
9. Filtrar productos con precios entre \$1000 y \$3000.
10. Mostrar las categorías disponibles con sus descripciones.



```
1 package CasoPractico1;
2
3
4 public enum CategoriaProducto {
5
6     ALIMENTOS("Productos comestibles"),
7     ELECTRONICA("Dispositivos electronicos"),
8     ROPA("Prendas de vestir"),
9     HOGAR("Articulos para el hogar");
10
11     private final String descripcion;
12
13     private CategoriaProducto(String descripcion) {
14         this.descripcion = descripcion;
15     }
16
17     public String getDescripcion() {
18         return descripcion;
19     }
20
21
22 }
23
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```

CategoriaProducto.java x Inventario.java x
Source History
private List<Producto> productos;
11
12
13 public Inventario() {
14     this.productos = new ArrayList<>();
15 }
16
17 public void agregarProducto(Producto p){
18     if(buscarProductoPorId(p.getId()) != null){
19         System.out.println("Ya existe un producto con ID: " + p.getId());
20     }else {
21         productos.add(p);
22         System.out.println("Producto agregado: " + p.getNombre());
23     }
24 }
25
26 public void listarProductos(){
27     if(productos.isEmpty()){
28         System.out.println("No hay productos cargados.");
29         return;
30     }
31     System.out.println("Listado de productos en Inventario: ");
32     for (Producto p : productos) {
33         p.mostrarInfo();
34     }
35 }
36
37 public Producto buscarProductoPorId(String id){
38     for (Producto p : productos) {
39         if(p.getId().equalsIgnoreCase(id)){
40             return p;
41         }
42     }
43     return null;
44 }
45
46 public void eliminarProducto(String id) {
47     Producto encontrado = buscarProductoPorId(id);
48     if (encontrado != null) {
49         productos.remove(encontrado);
50         System.out.println("Producto eliminado: " + encontrado.getNombre());
51     } else {
52         System.out.println("No se encontró el producto con ID: " + id);
53     }
54 }
55
56 public void actualizarStock(String id, int nuevaCantidad) {
57     Producto p = buscarProductoPorId(id);
58     if (p != null) {
59         p.setCantidad(nuevaCantidad);
60         System.out.println("Stock actualizado para " + p.getNombre() + ": " + nuevaCantidad);
61     } else {
62         System.out.println("Producto no encontrado.");
63     }
64 }
65
66 public void filtrarPorCategoria(CategoriaProducto categoria) {
67     System.out.println("Productos de la categoría: " + categoria);
68     for (Producto p : productos) {
69         if (p.getCategoria() == categoria) {
70             p.mostrarInfo();
71         }
72     }
73 }
74
75 public int obtenerTotalStock() {
76     int total = 0;
77     for (Producto p : productos) {
78         total += p.getCantidad();
79     }
80     return total;
81 }
82
83 public Producto obtenerProductoConMayorStock() {
84     return productos.stream()
85         .max(Comparator.comparingInt(Producto::getCantidad))
86         .orElse(null);
87 }
88
89 public void filtrarProductosPorPrecio(double min, double max) {
90     System.out.println("Productos con precio entre $" + min + " y $" + max + ":");
91     for (Producto p : productos) {
92         if (p.getPrecio() >= min && p.getPrecio() <= max) {
93             p.mostrarInfo();
94         }
95     }
96 }
97
98 public void mostrarCategoriasDisponibles() {
99     System.out.println("Categorías disponibles:");
100     for (CategoriaProducto c : CategoriaProducto.values()) {
101         System.out.println(" " + c + ": " + c.getDescripcion());
102     }
103 }
104
105 }
106

```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```

CategoriaProducto.java x Inventario.java x Producto.java x
Source History
1 package CasoPractico1;
2
3
4 public class Producto {
5
6     private String id;
7     private String nombre;
8     private double precio;
9     private int cantidad;
10    private CategoriaProducto categoria;
11
12    public Producto(String id, String nombre, double precio, int cantidad, CategoriaProducto categoria) {
13        this.id = id;
14        this.nombre = nombre;
15        setPrecio(precio);
16        setCantidad(cantidad);
17        this.categoria = categoria;
18    }
19
20    public Producto(String id, String nombre, double precio, CategoriaProducto categoria) {
21        this(id, nombre, precio, 0, categoria);
22    }
23
24    public String getId() {
25        return id;
26    }
27
28    public void setId(String id) {
29        this.id = id;
30    }
31
32    public String getNombre() {
33        return nombre;
34    }
35
36    public void setNombre(String nombre) {
37        this.nombre = nombre;
38    }
39
40    public double getPrecio() {
41        return precio;
42    }
43
44    public void setPrecio(double precio) {
45        if(precio < 0){
46            System.out.println("El precio no puede ser negativo");
47        }
48        this.precio = precio;
49    }
50
51    public int getCantidad() {
52        return cantidad;
53    }
54
55    public void setCantidad(int cantidad) {
56        if(cantidad < 0){
57            System.out.println("La cantidad no puede ser negativa");
58        }
59        this.cantidad = cantidad;
60    }
61
62    public CategoriaProducto getCategoria() {
63        return categoria;
64    }
65
66    public void setCategoria(CategoriaProducto categoria) {
67        this.categoria = categoria;
68    }
69
70    @Override
71    public String toString() {
72        return "Producto{" + "id=" + id + ", nombre=" + nombre +
73            ", precio=" + precio + ", cantidad=" + cantidad +
74            ", categoria=" + categoria + '}';
75    }
76
77
78    public void mostrarInfo(){
79        System.out.println(this.toString());
80    }
81
82
83 }
84
  
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```
1 package CasoPractico1;
2
3
4 public class main {
5
6     public static void main(String[] args) {
7
8         Inventario inventario = new Inventario();
9
10        Producto p1 = new Producto("01", "Yerba Mate", 1500, 20, CategoriaProducto.ALIMENTOS);
11        Producto p2 = new Producto("02", "Notebook Lenovo", 250000, 5, CategoriaProducto.ELECTRONICA);
12        Producto p3 = new Producto("03", "Remera", 8000, 15, CategoriaProducto.ROPA);
13        Producto p4 = new Producto("04", "Smart TV Samsung", 400000, 3, CategoriaProducto.ELECTRONICA);
14        Producto p5 = new Producto("05", "Silla de escritorio", 22000, 10, CategoriaProducto.HOGAR);
15
16
17        inventario.agregarProducto(p1);
18        inventario.agregarProducto(p2);
19        inventario.agregarProducto(p3);
20        inventario.agregarProducto(p4);
21        inventario.agregarProducto(p5);
22
23
24        System.out.println("\n-----");
25        inventario.listarProductos();
26
27        System.out.println("\n-----");
28        System.out.println(" Buscando producto con ID 03...");
29        Producto buscado = inventario.buscarProductoPorId("03");
30        if (buscado != null) buscado.mostrarInfo();
31
32        System.out.println("\n-----");
33        inventario.filtrarPorCategoria(CategoriaProducto.ELECTRONICA);
34
35        System.out.println("\n-----");
36        inventario.actualizarStock("01", 30);
37
38        System.out.println("\n-----");
39        inventario.eliminarProducto("05");
40
41        System.out.println("\n-----");
42        System.out.println("Total de unidades en stock: " + inventario.obtenerTotalStock());
43
44        System.out.println("\n-----");
45        Producto mayor = inventario.obtenerProductoConMayorStock();
46        System.out.println("Producto con mayor stock:");
47        if (mayor != null) mayor.mostrarInfo();
48
49        System.out.println("\n-----");
50        inventario.filtrarProductosPorPrecio(1000, 300000);
51
52        System.out.println("\n-----");
53        inventario.mostrarCategoriasDisponibles();
54
55    }
56
57 }
58
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```
Output - trabajoPractico06 (run)
run:
Producto agregado: Yerba Mate
Producto agregado: Notebook Lenovo
Producto agregado: Remera
Producto agregado: Smart TV Samsung
Producto agregado: Silla de escritorio

-----
Listado de productos en Inventario:
Producto{id=01, nombre=Yerba Mate, precio=1500.0, cantidad=20, categoria=ALIMENTOS}
Producto{id=02, nombre=Notebook Lenovo, precio=250000.0, cantidad=5, categoria=ELECTRONICA}
Producto{id=03, nombre=Remera, precio=8000.0, cantidad=15, categoria=ROPA}
Producto{id=04, nombre=Smart TV Samsung, precio=400000.0, cantidad=3, categoria=ELECTRONICA}
Producto{id=05, nombre=Silla de escritorio, precio=22000.0, cantidad=10, categoria=HOGAR}

-----
Buscando producto con ID 03...
Producto{id=03, nombre=Remera, precio=8000.0, cantidad=15, categoria=ROPA}

-----
Productos de la categoría: ELECTRONICA
Producto{id=02, nombre=Notebook Lenovo, precio=250000.0, cantidad=5, categoria=ELECTRONICA}
Producto{id=04, nombre=Smart TV Samsung, precio=400000.0, cantidad=3, categoria=ELECTRONICA}

-----
Stock actualizado para Yerba Mate: 30

-----
Producto eliminado: Silla de escritorio

-----
Total de unidades en stock: 53

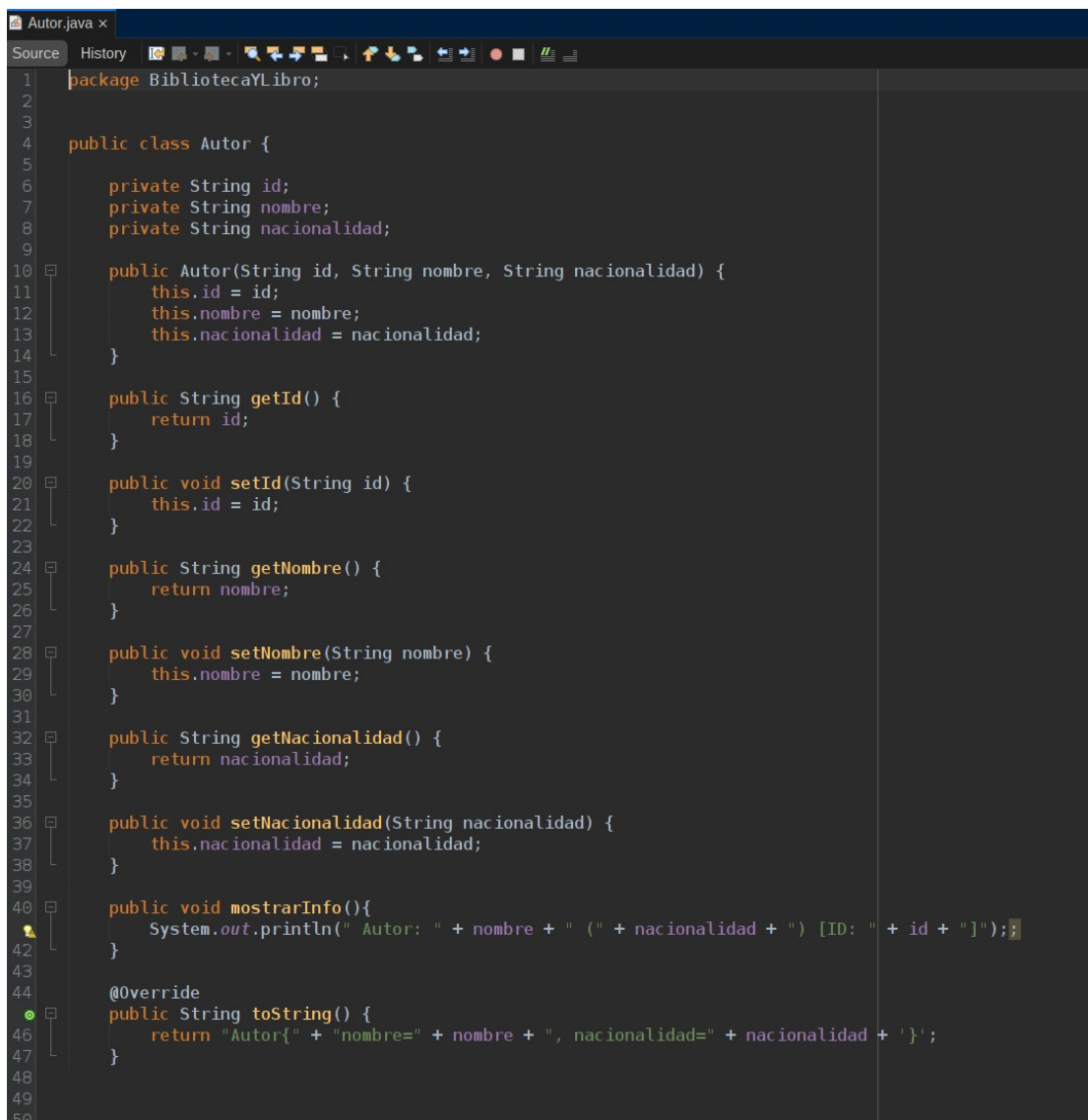
-----
Producto con mayor stock:
Producto{id=01, nombre=Yerba Mate, precio=1500.0, cantidad=30, categoria=ALIMENTOS}

-----
🔍 Productos con precio entre $1000.0 y $30000.0:
Producto{id=01, nombre=Yerba Mate, precio=1500.0, cantidad=30, categoria=ALIMENTOS}
Producto{id=03, nombre=Remera, precio=8000.0, cantidad=15, categoria=ROPA}

-----
Categorías disponibles:
- ALIMENTOS: Productos comestibles
- ELECTRONICA: Dispositivos electronicos
- ROPA: Prendas de vestir
- HOGAR: Articulos para el hogar
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

Ejercicio 2: Biblioteca y libro

1. Creamos una biblioteca.
2. Crear al menos tres autores
3. Agregar 5 libros asociados a alguno de los Autores a la biblioteca.
4. Listar todos los libros con su información y la del autor.
5. Buscar un libro por su ISBN y mostrar su información.
6. Filtrar y mostrar los libros publicados en un año específico.
7. Eliminar un libro por su ISBN y listar los libros restantes.
8. Mostrar la cantidad total de libros en la biblioteca.
9. Listar todos los autores de los libros disponibles en la biblioteca.



```
1 package BibliotecaYLibro;
2
3
4 public class Autor {
5
6     private String id;
7     private String nombre;
8     private String nacionalidad;
9
10    public Autor(String id, String nombre, String nacionalidad) {
11        this.id = id;
12        this.nombre = nombre;
13        this.nacionalidad = nacionalidad;
14    }
15
16    public String getId() {
17        return id;
18    }
19
20    public void setId(String id) {
21        this.id = id;
22    }
23
24    public String getNombre() {
25        return nombre;
26    }
27
28    public void setNombre(String nombre) {
29        this.nombre = nombre;
30    }
31
32    public String getNacionalidad() {
33        return nacionalidad;
34    }
35
36    public void setNacionalidad(String nacionalidad) {
37        this.nacionalidad = nacionalidad;
38    }
39
40    public void mostrarInfo(){
41        System.out.println(" Autor: " + nombre + " (" + nacionalidad + ") [ID: " + id + " ]");
42    }
43
44    @Override
45    public String toString() {
46        return "Autor{" + "nombre=" + nombre + ", nacionalidad=" + nacionalidad + '}';
47    }
48
49
50 }
```


TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```

Autor.java x  Biblioteca.java x
Source  History  [Icons]

1  package BibliotecaYLibro;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6
7  public class Biblioteca {
8
9      private String nombre;
10     private List<Libro> libros;
11
12     public Biblioteca(String nombre) {
13         this.nombre = nombre;
14         this.libros = new ArrayList<>();
15     }
16
17     public void agregarLibro(String isbn, String titulo, int anioPublicacion, Autor autor){
18         Libro libro = new Libro(isbn, titulo, anioPublicacion, autor);
19         libros.add(libro);
20         System.out.println("Libro agregado: " + titulo);
21     }
22
23     public void listarLibros(){
24         if (libros.isEmpty()) {
25             System.out.println("No hay libros en la biblioteca.");
26             return;
27         }
28         System.out.println("Listado de libros en la biblioteca " + nombre + ":");
29         for (Libro l : libros) {
30             l.mostrarInfo();
31         }
32     }
33
34     public Libro buscarLibroPorIsbn(String isbn){
35         for (Libro l : libros) {
36             if(l.getIsbn().equalsIgnoreCase(isbn)){
37                 return l;
38             }
39         }
40         return null;
41     }
42
43     public void eliminarLibro(String isbn){
44         Libro encontrado = buscarLibroPorIsbn(isbn);
45         if(encontrado != null){
46             libros.remove(encontrado);
47             System.out.println("Libro eliminado: " + encontrado.getTitulo());
48         }else{
49             System.out.println("Libro no encontrado con ISBN: " + isbn);
50         }
51     }
52
53     public int obtenerCantidadLibros(){
54         return libros.size();
55     }
56
57     public void filtrarLibrosPorAnio(int anio){
58         System.out.println("Libros publicados en " + anio + ":");
59         for (Libro l : libros) {
60             if(l.getAnioPublicacion() == anio){
61                 l.mostrarInfo();
62             }
63         }
64     }
65
66     public void mostrarAutoresDisponibles(){
67         System.out.println("Autores disponibles en la biblioteca: ");
68         libros.stream().map(l -> l.getAutor().getNombre()).distinct().forEach(System.out::println);
69     }
70
71
  
```


TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```

1 package BibliotecaYLibro;
2
3
4 public class Libro {
5
6     private String isbn;
7     private String titulo;
8     private int anioPublicacion;
9     private Autor autor;
10
11     public Libro(String isbn, String titulo, int anioPublicacion, Autor autor) {
12         this.isbn = isbn;
13         this.titulo = titulo;
14         this.anioPublicacion = anioPublicacion;
15         this.autor = autor;
16     }
17
18     public String getIsbn() {
19         return isbn;
20     }
21
22     public String getTitulo() {
23         return titulo;
24     }
25
26     public int getAnioPublicacion() {
27         return anioPublicacion;
28     }
29
30     public Autor getAutor() {
31         return autor;
32     }
33
34     public void mostrarInfo(){
35         System.out.println("Libro: " + titulo +
36             "ISBN: " + isbn +
37             "Año: " + anioPublicacion +
38             "Autor: " + autor);
39     }
40
41     @Override
42     public String toString() {
43         return "Libro{" + "titulo=" + titulo + ", anioPublicacion=" + anioPublicacion + ", autor=" + autor + '}';
44     }
45
46
47 }
48
49
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```

1 package BibliotecaYLibro;
2
3
4 public class main {
5     public static void main(String[] args) {
6
7         Autor a1 = new Autor("01", "Julio Cortázar", "Argentina");
8         Autor a2 = new Autor("02", "Isabel Allende", "Chile");
9         Autor a3 = new Autor("03", "Gabriel García Márquez", "Colombia");
10
11         Biblioteca biblio = new Biblioteca("Biblioteca Central UTN");
12
13         biblio.agregarLibro("01", "Rayuela", 1963, a1);
14         biblio.agregarLibro("02", "La casa de los espíritus", 1982, a2);
15         biblio.agregarLibro("03", "Cien años de soledad", 1967, a3);
16         biblio.agregarLibro("04", "Bestiario", 1951, a1);
17         biblio.agregarLibro("05", "Eva Luna", 1987, a2);
18
19         System.out.println("\n-----");
20         biblio.listarLibros();
21
22         System.out.println("\n-----");
23         System.out.println("Buscando libro con ISBN 03...");
24         Libro encontrado = biblio.buscarLibroPorIsbn("03");
25         if (encontrado != null) encontrado.mostrarInfo();
26
27         System.out.println("\n-----");
28         biblio.filtrarLibrosPorAño(1982);
29
30         System.out.println("\n-----");
31         biblio.eliminarLibro("04");
32
33         System.out.println("\n-----");
34         System.out.println("Cantidad total de libros: " + biblio.obtenerCantidadLibros());
35
36         System.out.println("\n-----");
37         biblio.mostrarAutoresDisponibles();
38
39     }
40 }
41
42 }
43

```

```

Output - trabajoPractico06 (run)

run:
Libro agregado: Rayuela
Libro agregado: La casa de los espíritus
Libro agregado: Cien años de soledad
Libro agregado: Bestiario
Libro agregado: Eva Luna

-----
Listado de libros en la biblioteca Biblioteca Central UTN:
Libro: RayuelaISBN: 01Año: 1963Autor: Autor{nombre=Julio Cortázar, nacionalidad=Argentina}
Libro: La casa de los espíritusISBN: 02Año: 1982Autor: Autor{nombre=Isabel Allende, nacionalidad=Chile}
Libro: Cien años de soledadISBN: 03Año: 1967Autor: Autor{nombre=Gabriel García Márquez, nacionalidad=Colombia}
Libro: BestiarioISBN: 04Año: 1951Autor: Autor{nombre=Julio Cortázar, nacionalidad=Argentina}
Libro: Eva LunaISBN: 05Año: 1987Autor: Autor{nombre=Isabel Allende, nacionalidad=Chile}

-----
Buscando libro con ISBN 03...
Libro: Cien años de soledadISBN: 03Año: 1967Autor: Autor{nombre=Gabriel García Márquez, nacionalidad=Colombia}

-----
Libros publicados en 1982 :
Libro: La casa de los espíritusISBN: 02Año: 1982Autor: Autor{nombre=Isabel Allende, nacionalidad=Chile}

-----
Libro eliminado: Bestiario

-----
Cantidad total de libros: 4

-----
Autores disponibles en la biblioteca:
Julio Cortázar
Isabel Allende
Gabriel García Márquez
BUILD SUCCESSFUL (total time: 0 seconds)

```

Ejercicio 3: Universidad, Profesor y Curso

1. Crear al menos 3 profesores y 5 cursos.
2. Agregar profesores y cursos a la universidad.
3. Asignar profesores a cursos usando `asignarProfesorACurso(...)`.
4. Listar cursos con su profesor y profesores con sus cursos.
5. Cambiar el profesor de un curso y verificar que ambos lados quedan sincronizados.
6. Remover un curso y confirmar que ya no aparece en la lista del profesor.
7. Remover un profesor y dejar profesor = null,
8. Mostrar un reporte: cantidad de cursos por profesor.

```
Universidad.java x
Source History
1 package UniversidadProfesorCurso;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Universidad {
7
8     private String nombre;
9     private List<Profesor> profesores;
10    private List<Curso> cursos;
11
12    public Universidad(String nombre) {
13        this.nombre = nombre;
14        this.profesores = new ArrayList<>();
15        this.cursos = new ArrayList<>();
16    }
17
18    public void agregarProfesor(Profesor p) {
19        profesores.add(p);
20        System.out.println("Profesor agregado: " + p.getNombre());
21    }
22
23    public void agregarCurso(Curso c) {
24        cursos.add(c);
25        System.out.println("Curso agregado: " + c.getNombre());
26    }
27
28    public void asignarProfesorACurso(String codigoCurso, String idProfesor) {
29        Curso curso = buscarCursoPorCodigo(codigoCurso);
30        Profesor profesor = buscarProfesorPorId(idProfesor);
31
32        if (curso == null || profesor == null) {
33            System.out.println("Curso o profesor no encontrado.");
34            return;
35        }
36
37        curso.setProfesor(profesor);
38        System.out.println("Asignado profesor " + profesor.getNombre()
39            + " al curso " + curso.getNombre());
40    }
41
42    public void listarProfesores() {
43        System.out.println("? Profesores registrados: ");
44        for (Profesor p : profesores) {
45            p.mostrarInfo();
46        }
47    }
48
49    public void listarCursos() {
50        System.out.println("Cursos registrados:");
51        for (Curso c : cursos) {
52            c.mostrarInfo();
53        }
54    }
55
56    public Profesor buscarProfesorPorId(String id) {
57        for (Profesor p : profesores) {
58            if (p.getId().equalsIgnoreCase(id)) {
59                return p;
60            }
61        }
62        return null;
63    }
64 }
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

```
public Curso buscarCursoPorCodigo(String codigo) {
    for (Curso c : cursos) {
        if (c.getCodigo().equalsIgnoreCase(codigo)) {
            return c;
        }
    }
    return null;
}

public void eliminarCurso(String codigo) {
    Curso c = buscarCursoPorCodigo(codigo);
    if (c != null) {
        if (c.getProfesor() != null) {
            c.getProfesor().eliminarCurso(c);
        }
        cursos.remove(c);
        System.out.println("Curso eliminado: " + c.getNombre());
    }
}

public void eliminarProfesor(String id) {
    Profesor p = buscarProfesorPorId(id);
    if (p != null) {
        for (Curso c : new ArrayList<>(p.getCursos())) {
            c.setProfesor(null);
        }
        profesores.remove(p);
        System.out.println("Profesor eliminado: " + p.getNombre());
    }
}

public void reporteCursosPorProfesor() {
    System.out.println("Cantidad de cursos por profesor:");
    for (Profesor p : profesores) {
        System.out.println("- " + p.getNombre() + ": " + p.getCursos().size() + " cursos");
    }
}
```

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

```
Universidad.java x  Profesor.java x
Source  History  [Icons]

7  public class Profesor {
8
9      private String id;
10     private String nombre;
11     private String especialidad;
12     private List<Curso> cursos;
13
14     public Profesor(String id, String nombre, String especialidad) {
15         this.id = id;
16         this.nombre = nombre;
17         this.especialidad = especialidad;
18         this.cursos = new ArrayList<>();
19     }
20
21     public String getId() {
22         return id;
23     }
24
25     public String getNombre() {
26         return nombre;
27     }
28
29     public String getEspecialidad() {
30         return especialidad;
31     }
32
33     public List<Curso> getCursos() {
34         return cursos;
35     }
36
37     public void agregarCurso(Curso c) {
38         if(!cursos.contains(c)){
39             cursos.add(c);
40             if(c.getProfesor() != this){
41                 c.setProfesor(this);
42             }
43         }
44     }
45
46     public void eliminarCurso(Curso c) {
47         if (cursos.contains(c)) {
48             cursos.remove(c);
49             if (c.getProfesor() == this) {
50                 c.setProfesor(null);
51             }
52         }
53     }
54
55     public void listarCursos() {
56         System.out.println("Cursos dictados por " + nombre + ":");
57         for (Curso c : cursos) {
58             System.out.println("- " + c.getCodigo() + " | " + c.getNombre());
59         }
60     }
61
62     public void mostrarInfo() {
63         System.out.println(" Profesor: " + nombre +
64             " (" + especialidad + ") - Cursos: " + cursos.size());
65     }
66
67     @Override
68     public String toString() {
69         return nombre + " (" + especialidad + ")";
70     }
71 }
72
```


TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

```

1 package UniversidadProfesorCurso;
2
3 public class Curso {
4
5     private String codigo;
6     private String nombre;
7     private Profesor profesor;
8
9     public Curso(String codigo, String nombre) {
10         this.codigo = codigo;
11         this.nombre = nombre;
12     }
13
14     public String getCodigo() {
15         return codigo;
16     }
17
18     public String getNombre() {
19         return nombre;
20     }
21
22     public Profesor getProfesor() {
23         return profesor;
24     }
25
26     public void setProfesor(Profesor nuevoProfesor) {
27         if (this.profesor == nuevoProfesor) {
28             return;
29         }
30         if (this.profesor != null) {
31             this.profesor.getCursos().remove(this);
32         }
33         this.profesor = nuevoProfesor;
34         if (nuevoProfesor != null && !nuevoProfesor.getCursos().contains(this)) {
35             nuevoProfesor.getCursos().add(this);
36         }
37     }
38
39     public void mostrarInfo() {
40         String profe = (profesor != null) ? profesor.getNombre() : "Sin asignar";
41         System.out.println("Curso: " + nombre + " (" + codigo + ") - Profesor: " + profe);
42     }
43
44
45     @Override
46     public String toString() {
47         return codigo + " - " + nombre;
48     }
49

```


TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA

```
1 package UniversidadProfesorCurso;
2
3
4 public class main {
5
6     public static void main(String[] args) {
7
8         Universidad utn = new Universidad("UTN");
9
10
11         Profesor p1 = new Profesor("P01", "Ana Pérez", "Programación");
12         Profesor p2 = new Profesor("P02", "Carlos Gómez", "Bases de Datos");
13         Profesor p3 = new Profesor("P03", "Laura Díaz", "Redes");
14
15         utn.agregarProfesor(p1);
16         utn.agregarProfesor(p2);
17         utn.agregarProfesor(p3);
18
19
20         Curso c1 = new Curso("C01", "Programación II");
21         Curso c2 = new Curso("C02", "Base de Datos I");
22         Curso c3 = new Curso("C03", "Redes I");
23         Curso c4 = new Curso("C04", "Sistemas Operativos");
24         Curso c5 = new Curso("C05", "Algoritmos");
25
26         utn.agregarCurso(c1);
27         utn.agregarCurso(c2);
28         utn.agregarCurso(c3);
29         utn.agregarCurso(c4);
30         utn.agregarCurso(c5);
31
32         System.out.println("\n-----");
33         utn.asignarProfesorACurso("C01", "P001");
34         utn.asignarProfesorACurso("C02", "P002");
35         utn.asignarProfesorACurso("C03", "P003");
36         utn.asignarProfesorACurso("C04", "P001");
37         utn.asignarProfesorACurso("C05", "P001");
38
39         System.out.println("\n-----");
40         utn.listarProfesores();
41         System.out.println();
42         utn.listarCursos();
43
44         System.out.println("\n-----");
45         utn.asignarProfesorACurso("C04", "P02");
46
47         System.out.println("\n-----");
48         utn.eliminarCurso("C05");
49
50         System.out.println("\n-----");
51         utn.eliminarProfesor("P03");
52
53         System.out.println("\n-----");
54         utn.reporteCursosPorProfesor();
55     }
56 }
57
58
```

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Output - trabajoPractico06 (run)

```
run:
Profesor agregado: Ana Pérez
Profesor agregado: Carlos Gómez
Profesor agregado: Laura Díaz
Curso agregado: Programación II
Curso agregado: Base de Datos I
Curso agregado: Redes I
Curso agregado: Sistemas Operativos
Curso agregado: Algoritmos

-----
Curso o profesor no encontrado.
Curso o profesor no encontrado.
Curso o profesor no encontrado.
Curso o profesor no encontrado.
Curso o profesor no encontrado.

-----
? Profesores registrados:
Profesor: Ana Pérez (Programación) - Cursos: 0
Profesor: Carlos Gómez (Bases de Datos) - Cursos: 0
Profesor: Laura Díaz (Redes) - Cursos: 0

Cursos registrados:
Curso: Programación II (C01) - Profesor: Sin asignar
Curso: Base de Datos I (C02) - Profesor: Sin asignar
Curso: Redes I (C03) - Profesor: Sin asignar
Curso: Sistemas Operativos (C04) - Profesor: Sin asignar
Curso: Algoritmos (C05) - Profesor: Sin asignar

-----
Asignado profesor Carlos Gómez al curso Sistemas Operativos

-----
Curso eliminado: Algoritmos

-----
Profesor eliminado: Laura Díaz

-----
Cantidad de cursos por profesor:
- Ana Pérez: 0 cursos
- Carlos Gómez: 1 cursos
BUILD SUCCESSFUL (total time: 0 seconds)
```