

Trabajo práctico Nº 6

“Colecciones y Sistema de Stock”

Alumnos: Alemis, Juan Cruz – juancruzalemis@gmail.com

Materia: Programacion II

Caso Práctico 1

1. Descripción general Se debe desarrollar un sistema de stock que permita gestionar productos en una tienda, controlando su disponibilidad, precios y categorías. La información se modelará utilizando clases, colecciones dinámicas y enumeraciones en Java.

2. Clases a implementar Clase Producto

The screenshot shows a Java code editor with the following code:

```
1 package tp6;
2
3
4
5 public enum CategoríaProductos {
6     ALIMENTOS("Productos comestibles"),
7     ELECTRONICA("Dispositivos electrónicos"),
8     ROPA("Prendas de vestir"),
9     HOGAR("Articulos para el hogar");
10
11     // Atributoa
12     private final String descripcion;
13
14     // Constructor
15     CategoríaProductos(String descripcion) {
16         this.descripcion = descripcion;
17     }
18
19
20     public String getDescripcion() {
21         return descripcion;
22     }
23 }
```

The code defines an enum named `CategoríaProductos` with four entries: `ALIMENTOS`, `ELECTRONICA`, `ROPA`, and `HOGAR`. It includes a private attribute `descripcion` and a constructor that initializes it. There is also a `getDescripcion` method.

```
1  package tp6;
2
3  import tp6.CategoríaProductos;
4
5  public class Productos {
6      private String id;
7      private String nombre;
8      private double precio;
9      private int cantidad;
10     private CategoríaProductos categoria;
11
12    public Productos(String id, String nombre, double precio, int cantidad, CategoríaProductos categoria) {
13        this.id = id;
14        this.nombre = nombre;
15        this.precio = precio;
16        this.cantidad = cantidad;
17        this.categoria = categoria;
18    }
19
20    public String getId() {
21        return id;
22    }
23
24    public String getNombre() {
25        return nombre;
26    }
27
28    public double getPrecio() {
29        return precio;
30    }
31
32    public int getCantidad() {
33        return cantidad;
34    }
35
36    public CategoríaProductos getCategoría() {
37        return categoria;
38    }
39
40    public void setId(String id) {
41        this.id = id;
42    }
43
44    public void setNombre(String nombre) {
45        this.nombre = nombre;
46    }
47
48    public void setPrecio(double precio) {
49        this.precio = precio;
50    }
51
52    public void setCantidad(int cantidad) {
53        this.cantidad = cantidad;
54    }
55
56    public void setCategoría(CategoríaProductos categoria) {
57        this.categoria = categoria;
58    }
59
60    public void mostrarInfo() {
61        System.out.println("ID: " + id);
62        System.out.println("Nombre: " + nombre);
63        System.out.println("Precio: $" + precio);
64        System.out.println("Cantidad en stock: " + cantidad);
65        System.out.println("Categoría: " + categoria + " (" + categoria.getDescripción() + ")");
66        System.out.println("");
67        System.out.println(":::::::::::::::::::");
68        System.out.println("");
69    }
70 }
```

```
1 package tp6;
2
3 import java.util.ArrayList;
4
5 public class Inventario {
6     // Atributo
7     private ArrayList<Productos> productos;
8
9     // Constructor
10    public Inventario() {
11        this.productos = new ArrayList<>();
12    }
13
14
15    public void agregarProducto(Productos p) {
16        if (p == null) return;
17        productos.add(p);
18    }
19
20
21    public void agregarProductos(Productos... ps) {
22        if (ps == null) return;
23        for (Productos p : ps) {
24            agregarProducto(p);
25        }
26    }
27
28    // ===== Listado =====
29    public void listarProductos() {
30        if (productos.isEmpty()) {
31            System.out.println("No hay productos en el inventario.");
32        } else {
33            for (Productos p : productos) {
34                p.mostrarInfo();
35            }
36        }
37    }
38}
```

```
package tp6;

public class tp6 {

    public static void main(String[] args) {
        // Crear inventario
        Inventario inventario = new Inventario();

        Productos p1 = new Productos("P001", "Arroz", 1250, 45, CategoriaProductos.ALIMENTOS);
        Productos p2 = new Productos("P002", "Celular", 195000, 4, CategoriaProductos.ELECTRONICA);
        Productos p3 = new Productos("P003", "Remera", 3500, 15, CategoriaProductos.ROPA);
        Productos p4 = new Productos("P004", "Sillon", 200000, 10, CategoriaProductos.HOGAR);
        Productos p5 = new Productos("P005", "Pan", 500, 100, CategoriaProductos.ALIMENTOS);

        inventario.agregarProductos(p1);
        inventario.agregarProductos(p2);
        inventario.agregarProductos(p3);
        inventario.agregarProductos(p4);
        inventario.agregarProductos(p5);

        // 2. Listar todos los productos
        System.out.println(":::::::::: LISTA DE PRODUCTOS ::::::::::::");
        System.out.println("");
        inventario.listarProductos();

        // 3. Buscar un producto por ID
        System.out.println(":::::::::: BUSCAR PRODUCTO P003 ::::::::::::");
        System.out.println("");
        Productos buscado = inventario.buscarProductosPorId("P003");
        if (buscado != null) buscado.mostrarInfo();

        // 4. Filtrar y mostrar productos de una categoría
        System.out.println(":::::::::: FILTRAR POR CATEGORÍA: ALIMENTOS ::::::::::::");
        System.out.println("");
        inventario.filtrarPorCategoria(CategoriaProductos.ALIMENTOS);

        // 5. Eliminar un producto y listar los restantes
        System.out.println(":::::::::: ELIMINAR PRODUCTO P002 :::::::::::: ");
        System.out.println("");
        inventario.eliminarProductos("P002");
        inventario.listarProductos();

        // 6. Actualizar el stock de un producto
        System.out.println(":::::::::: ACTUALIZAR STOCK P001 ::::::::::::");
        System.out.println("");
        inventario.actualizarStock("P001", 70);
        inventario.buscarProductoPorId("P001").mostrarInfo();

        // 7. Mostrar total de stock disponible
        System.out.println(":::::::::: TOTAL DE STOCK DISPONIBLE:::::::::: ");
        System.out.println("");
        System.out.println("Total stock: " + inventario.obtenerTotalStock());

        // 8. Obtener y mostrar el producto con mayor stock
        System.out.println(":::::::::: PRODUCTO CON MAYOR STOCK ::::::::::::");
        Productos mayor = inventario.obtenerProductoConMayorStock();
        if (mayor != null) mayor.mostrarInfo();

        // 9. Filtrar productos por precio entre $1000 y $3000
        System.out.println(":::::::::: PRODUCTOS ENTRE $1000 y $3000:::::::::: ");
        inventario.filtrarProductosPorPrecio(1000, 3000);

        // 10. Mostrar categorías disponibles
        System.out.println(":::::::::: CATEGORÍAS DISPONIBLES::::::::::");
        inventario.mostrarCategoriasDisponibles();
    }
}
```

```
Output - Trabajo ejercico1 (run) ×
▶ run:
::::::: LISTA DE PRODUCTOS :::::::

ID: P001
Nombre: Arroz
Precio: $1250.0
Cantidad en stock: 45
Categoría: ALIMENTOS (Productos comestibles)

:::::::

ID: P002
Nombre: Celular
Precio: $195000.0
Cantidad en stock: 4
Categoría: ELECTRONICA (Dispositivos electrónicos)

:::::::

ID: P003
Nombre: Remera
Precio: $3500.0
Cantidad en stock: 15
Categoría: ROPA (Prendas de vestir)

:::::::

ID: P004
Nombre: Sillon
Precio: $200000.0
Cantidad en stock: 10
Categoría: HOGAR (Artículos para el hogar)

:::::::
```

```
Output - Tp6ejercicio1 (run) × < > □

::::::: BUSCAR PRODUCTO P003 :::::::

ID: P003
Nombre: Remera
Precio: $3500.0
Cantidad en stock: 15
Categoría: ROPA (Prendas de vestir)

:::::::

::::::: FILTRAR POR CATEGORÍA: ALIMENTOS :::::::

ID: P001
Nombre: Arroz
Precio: $1250.0
Cantidad en stock: 45
Categoría: ALIMENTOS (Productos comestibles)

:::::::

ID: P005
Nombre: Pan
Precio: $500.0
Cantidad en stock: 100
Categoría: ALIMENTOS (Productos comestibles)

:::::::
```

```
Output - Tp6ejercicio1 (run) ×

::::::::::::: ELIMINAR PRODUCTO P002 ::::::::::::
Producto con ID P002 eliminado.
ID: P001
Nombre: Arroz
Precio: $1250.0
Cantidad en stock: 45
Categoria: ALIMENTOS (Productos comestibles)

::::::::::::::::::
ID: P003
Nombre: Remera
Precio: $3500.0
Cantidad en stock: 15
Categoria: ROPA (Prendas de vestir)

::::::::::::::::::
ID: P004
Nombre: Sillon
Precio: $200000.0
Cantidad en stock: 10
Categoria: HOGAR (Articulos para el hogar)

::::::::::::::::::
ID: P005
Nombre: Pan
Precio: $500.0
Cantidad en stock: 100
Categoria: ALIMENTOS (Productos comestibles)
```

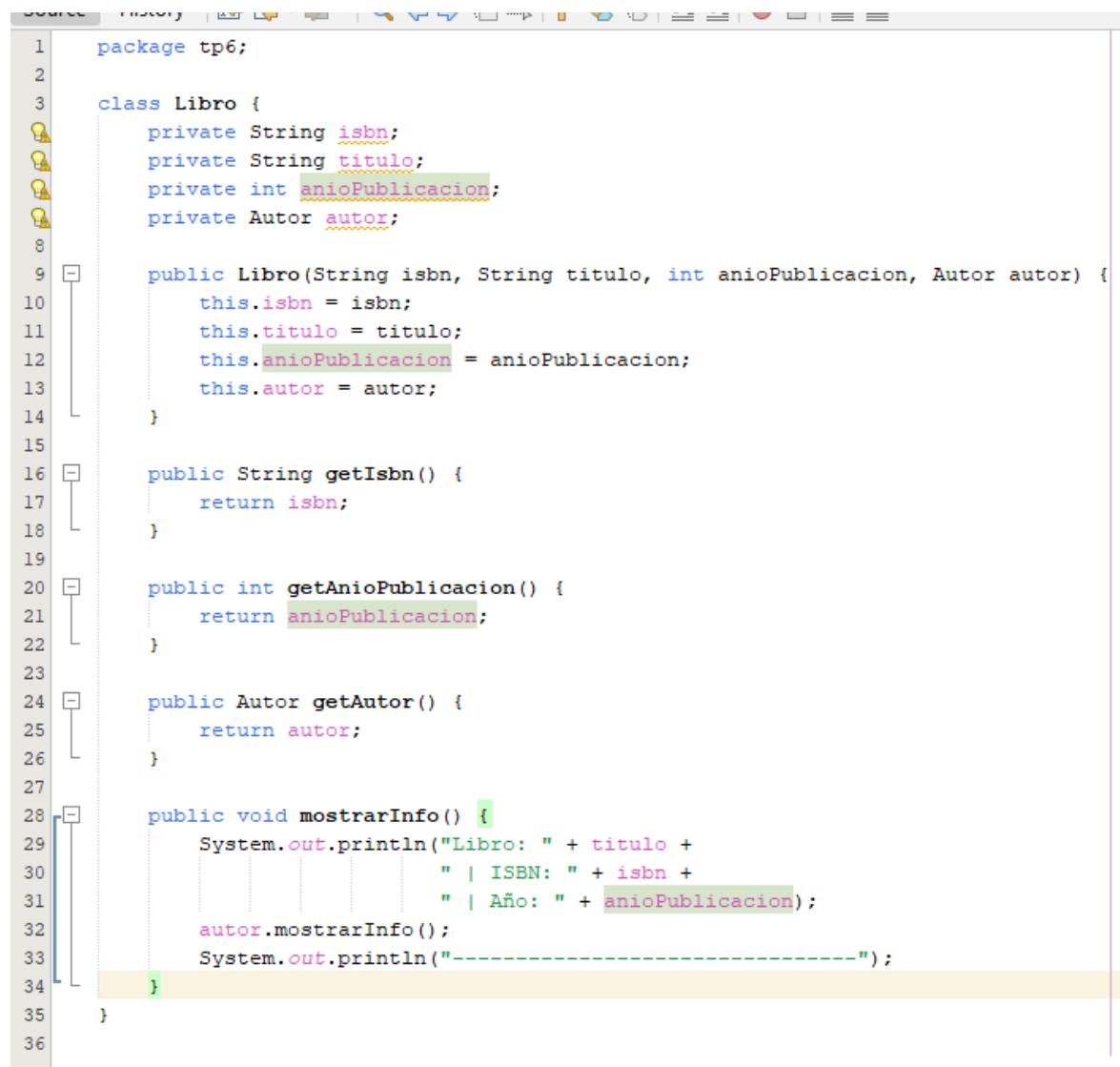
```
Output - Tp6ejercicio1 (run) × .....< > ▾ □  
::::::::: ACTUALIZAR STOCK P001 :::::::  
  
Stock actualizado para producto P001  
ID: P001  
Nombre: Arroz  
Precio: $1250.0  
Cantidad en stock: 70  
Categoría: ALIMENTOS (Productos comestibles)  
:::::::::::::::::::  
::::::::: TOTAL DE STOCK DISPONIBLE:::::::  
  
Total stock: 195  
::::::::: PRODUCTO CON MAYOR STOCK :::::::  
ID: P005  
Nombre: Pan  
Precio: $500.0  
Cantidad en stock: 100  
Categoría: ALIMENTOS (Productos comestibles)  
:::::::::::::::::::  
::::::::: PRODUCTOS ENTRE $1000 y $3000:::::::  
ID: P001  
Nombre: Arroz  
Precio: $1250.0  
Cantidad en stock: 70  
Categoría: ALIMENTOS (Productos comestibles)  
:::::::::::::::::::  
::::::::: CATEGORIAS DISPONIBLES:::::::  
ALIMENTOS ? Productos comestibles  
ELECTRONICA ? Dispositivos electrónicos  
ROPA ? Prendas de vestir  
HOGAR ? Artículos para el hogar  
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

```
1 package tp6;
2
3 class Autor {
4     private String id;
5     private String nombre;
6     private String nacionalidad;
7
8     public Autor(String id, String nombre, String nacionalidad) {
9         this.id = id;
10        this.nombre = nombre;
11        this.nacionalidad = nacionalidad;
12    }
13
14     public String getId() {
15         return id;
16     }
17
18     public String getNombre() {
19         return nombre;
20     }
21
22     public String getNacionalidad() {
23         return nacionalidad;
24     }
25
26     public void mostrarInfo() {
27         System.out.println("Autor: " + nombre + " (ID: " + id + ", Nacionalidad: " + nacionalidad + ")");
28     }
29 }
30 }
```



```
1 package tp6;
2
3 class Libro {
4     private String isbn;
5     private String titulo;
6     private int anioPublicacion;
7     private Autor autor;
8
9     public Libro(String isbn, String titulo, int anioPublicacion, Autor autor) {
10         this.isbn = isbn;
11         this.titulo = titulo;
12         this.anioPublicacion = anioPublicacion;
13         this.autor = autor;
14     }
15
16     public String getIsbn() {
17         return isbn;
18     }
19
20     public int getAnioPublicacion() {
21         return anioPublicacion;
22     }
23
24     public Autor getAutor() {
25         return autor;
26     }
27
28     public void mostrarInfo() {
29         System.out.println("Libro: " + titulo +
30                             " | ISBN: " + isbn +
31                             " | Año: " + anioPublicacion);
32         autor.mostrarInfo();
33         System.out.println("-----");
34     }
35 }
36 }
```

```
1 package tp6;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 class Biblioteca {
7     private String nombre;
8     private List<Libro> libros;
9
10    public Biblioteca(String nombre) {
11        this.nombre = nombre;
12        this.libros = new ArrayList<>();
13    }
14
15    public void agregarLibro(String isbn, String titulo, int anioPublicacion, Autor autor) {
16        Libro libro = new Libro(isbn, titulo, anioPublicacion, autor);
17        libros.add(libro);
18    }
19
20    public void listarLibros() {
21        if (libros.isEmpty()) {
22            System.out.println("No hay libros en la biblioteca.");
23            return;
24        }
25        System.out.println("Listado de libros en " + nombre + ":" );
26        for (Libro libro : libros) {
27            libro.mostrarInfo();
28        }
29    }
30}
```

```
31     public Libro buscarLibroPorIsbn(String isbn) {
32         for (Libro libro : libros) {
33             if (libro.getIsbn().equals(isbn)) {
34                 return libro;
35             }
36         }
37         return null;
38     }
39
40     public void eliminarLibro(String isbn) {
41         Libro libroAEliminar = null;
42         for (Libro libro : libros) {
43             if (libro.getIsbn().equals(isbn)) {
44                 libroAEliminar = libro;
45                 break;
46             }
47         }
48         if (libroAEliminar != null) {
49             libros.remove(libroAEliminar);
50         }
51     }
52
53     public int obtenerCantidadLibros() {
54         return libros.size();
55     }
56
57     public List<Libro> filtrarLibrosPorAnio(int anio) {
58         List<Libro> filtrados = new ArrayList<>();
59         for (Libro libro : libros) {
60             if (libro.getAnioPublicacion() == anio) {
61                 filtrados.add(libro);
62             }
63         }
64         return filtrados;
65     }
```

```
66
67     public void mostrarAutoresDisponibles() {
68         System.out.println("Autores disponibles en la biblioteca:");
69         List<String> autoresMostrados = new ArrayList<>();
70         for (Libro libro : libros) {
71             Autor autor = libro.getAutor();
72             if (!autoresMostrados.contains(autor.getId())) {
73                 autor.mostrarInfo();
74                 autoresMostrados.add(autor.getId());
75             }
76         }
77     }
78 }
```

```
1 package tp6;
2
3
4
5 import java.util.List;
6 public class Tp6 {
7
8
9     public static void main(String[] args) {
10         // 1. Crear biblioteca
11         Biblioteca biblioteca = new Biblioteca("Biblioteca Central");
12
13         // 2. Crear autores
14         Autor autor1 = new Autor("A1", "Gabriel García Márquez", "Colombiana");
15         Autor autor2 = new Autor("A2", "Julio Cortázar", "Argentina");
16         Autor autor3 = new Autor("A3", "Isabel Allende", "Chilena");
17
18         // 3. Agregar libros
19         biblioteca.agregarLibro("ISBN-001", "Cien años de soledad", 1967, autor1);
20         biblioteca.agregarLibro("ISBN-002", "El amor en los tiempos del cólera", 1985, autor1);
21         biblioteca.agregarLibro("ISBN-003", "Rayuela", 1963, autor2);
22         biblioteca.agregarLibro("ISBN-004", "La casa de los espíritus", 1982, autor3);
23         biblioteca.agregarLibro("ISBN-005", "Paula", 1994, autor3);
24
25         // 4. Listar libros
26         biblioteca.listarLibros();
27
28         // 5. Buscar libro por ISBN
29         System.out.println("Buscando libro con ISBN 'ISBN-003':");
30         Libro buscado = biblioteca.buscarLibroPorIsbn("ISBN-003");
31         if (buscado != null) {
32             buscado.mostrarInfo();
33         } else {
34             System.out.println("No se encontró el libro.");
35         }
36
37         // 6. Filtrar libros por año
38
39         // 6. Filtrar libros por año
40         System.out.println("Libros publicados en 1982:");
41         List<Libro> filtrados = biblioteca.filtrarLibrosPorAnio(1982);
42         for (Libro l : filtrados) {
43             l.mostrarInfo();
44         }
45
46         // 7. Eliminar libro por ISBN
47         System.out.println("Eliminando libro con ISBN 'ISBN-002'...");
48         biblioteca.eliminarLibro("ISBN-002");
49         biblioteca.listarLibros();
50
51         // 8. Mostrar cantidad de libros
52         System.out.println("Cantidad de libros en la biblioteca: " + biblioteca.obtenerCantidadLibros());
53
54     }
55 }
```

```
Output - Tp6Ejercicio2 (run) ×

run:
Listado de libros en Biblioteca Central:
Libro: Cien años de soledad | ISBN: ISBN-001 | Año: 1967
Autor: Gabriel García Márquez (ID: A1, Nacionalidad: Colombiana)
-----
Libro: El amor en los tiempos del cólera | ISBN: ISBN-002 | Año: 1985
Autor: Gabriel García Márquez (ID: A1, Nacionalidad: Colombiana)
-----
Libro: Rayuela | ISBN: ISBN-003 | Año: 1963
Autor: Julio Cortázar (ID: A2, Nacionalidad: Argentina)
-----
Libro: La casa de los espíritus | ISBN: ISBN-004 | Año: 1982
Autor: Isabel Allende (ID: A3, Nacionalidad: Chilena)
-----
Libro: Paula | ISBN: ISBN-005 | Año: 1994
Autor: Isabel Allende (ID: A3, Nacionalidad: Chilena)
-----
Buscando libro con ISBN 'ISBN-003':
Libro: Rayuela | ISBN: ISBN-003 | Año: 1963
Autor: Julio Cortázar (ID: A2, Nacionalidad: Argentina)
-----
Libros publicados en 1982:
Libro: La casa de los espíritus | ISBN: ISBN-004 | Año: 1982
Autor: Isabel Allende (ID: A3, Nacionalidad: Chilena)
-----

Eliminando libro con ISBN 'ISBN-002'...
Listado de libros en Biblioteca Central:
Libro: Cien años de soledad | ISBN: ISBN-001 | Año: 1967
Autor: Gabriel García Márquez (ID: A1, Nacionalidad: Colombiana)
-----
Libro: Rayuela | ISBN: ISBN-003 | Año: 1963
Autor: Julio Cortázar (ID: A2, Nacionalidad: Argentina)
-----
Libro: La casa de los espíritus | ISBN: ISBN-004 | Año: 1982
Autor: Isabel Allende (ID: A3, Nacionalidad: Chilena)
-----
Libro: Paula | ISBN: ISBN-005 | Año: 1994
Autor: Isabel Allende (ID: A3, Nacionalidad: Chilena)
-----
Cantidad de libros en la biblioteca: 4
Autores disponibles en la biblioteca:
Autor: Gabriel García Márquez (ID: A1, Nacionalidad: Colombiana)
Autor: Julio Cortázar (ID: A2, Nacionalidad: Argentina)
Autor: Isabel Allende (ID: A3, Nacionalidad: Chilena)
BUILD SUCCESSFUL (total time: 8 seconds)
```

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.

2. Clases a implementar Clase Profesor

```
1  package tp6;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  class Profesor {
7      private String id;
8      private String nombre;
9      private String especialidad;
10     private List<Curso> cursos;
11
12    public Profesor() {
13        this.cursos = new ArrayList<>();
14    }
15
16    public Profesor(String id, String nombre, String especialidad) {
17        this.id = id;
18        this.nombre = nombre;
19        this.especialidad = especialidad;
20        this.cursos = new ArrayList<>();
21    }
22
23    public String getId() { return id; }
24    public String getNombre() { return nombre; }
25
26    public void agregarCurso(Curso c) {
27        if (!cursos.contains(c)) {
28            cursos.add(c);
29            if (c.getProfesor() != this) c.setProfesor(this);
30        }
31    }
32}
```

```
33     public void eliminarCurso(Curso c) {
34         if (cursos.remove(c) && c.getProfesor() == this) c.setProfesor(null);
35     }
36
37     public void listarCursos() {
38         System.out.println("Cursos dictados por " + nombre + ":");
39         for (Curso c : cursos) System.out.println(" - " + c.getCodigo() + " " + c.getNombre());
40     }
41
42     public void mostrarInfo() {
43         System.out.println("Profesor: " + nombre + " | ID: " + id +
44                               " | Especialidad: " + especialidad +
45                               " | Cursos: " + cursos.size());
46     }
47
48     public int cantidadCursos() { return cursos.size(); }
49 }
```

```
1 package tp6;
2 class Curso {
3     private String codigo;
4     private String nombre;
5     private Profesor profesor;
6
7     public Curso(String codigo, String nombre) {
8         this.codigo = codigo;
9         this.nombre = nombre;
10        this.profesor = null;
11    }
12
13    public String getCodigo() {
14        return codigo;
15    }
16
17    public String getNombre() {
18        return nombre;
19    }
20
21    public Profesor getProfesor() {
22        return profesor;
23    }
24 }
```

```
25  public void setProfesor(Profesor p) {
26      if (this.profesor == p) return;
27
28
29      if (this.profesor != null) {
30          Profesor anterior = this.profesor;
31          this.profesor = null;
32          anterior.eliminarCurso(this);
33      }
34
35      this.profesor = p;
36
37
38      if (p != null) {
39          p.agregarCurso(this);
40      }
41  }
42
43  public void mostrarInfo() {
44      System.out.print("Curso: " + codigo + " - " + nombre);
45      if (profesor != null) {
46          System.out.println(" | Profesor: " + profesor.getNombre());
47      } else {
48          System.out.println(" | Profesor: [Sin asignar]");
49      }
50  }
51
52
53 }
```

```
1 package tp6;
2
3
4 import java.util.ArrayList;
5 import java.util.List;
6
7 class Universidad {
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    }
21
22    public void agregarCurso(Curso c) {
23        cursos.add(c);
24    }
25
26    public Profesor buscarProfesorPorId(String id) {
27        for (Profesor p : profesores) {
28            if (p.getId().equals(id)) return p;
29        }
30        return null;
31    }
32
33
34
35
36
37
38
39
40    public void asignarProfesorACurso(String codigoCurso, String idProfesor) {
41        Curso curso = buscarCursoPorCodigo(codigoCurso);
42        Profesor profesor = buscarProfesorPorId(idProfesor);
43        if (curso != null &amp; profesor != null) {
44            curso.setProfesor(profesor);
45        }
46    }
47
48    public void listarProfesores() {
49        System.out.println("Profesores en " + nombre + ":");
50        for (Profesor p : profesores) {
51            p.mostrarInfo();
52        }
53    }
54
55    public void listarCursos() {
56        System.out.println("Cursos en " + nombre + ":");
57        for (Curso c : cursos) {
58            c.mostrarInfo();
59        }
60    }
61
62    public void eliminarCurso(String codigo) {
63        Curso curso = buscarCursoPorCodigo(codigo);
64        if (curso != null) {
65            curso.setProfesor(null); // Romper la relación
66            cursos.remove(curso);
67        }
68    }
69
```

```
69
70     public void eliminarProfesor(String id) {
71         Profesor profesor = buscarProfesorPorId(id);
72         if (profesor != null) {
73             // Los cursos que dictaba quedan con profesor = null
74             List<Curso> cursosParaActualizar = new ArrayList<>(cursos);
75             for (Curso c : cursosParaActualizar) {
76                 if (c.getProfesor() == profesor) {
77                     c.setProfesor(null);
78                 }
79             }
80             profesores.remove(profesor);
81         }
82     }
83
84     public void reporteCursosPorProfesor() {
85         System.out.println("Reporte de cantidad de cursos por profesor:");
86         for (Profesor p : profesores) {
87             System.out.println(p.getNombre() + ": " + p.cantidadCursos() + " cursos");
88         }
89     }
90 }
```

```
1 package tp6;
2
3
4 import java.util.List;
5 public class Tp6 {
6
7
8     public static void main(String[] args) {
9         // 1. Crear profesores y cursos
10        Profesor prof1 = new Profesor("P1", "Ana Torres", "Matemática");
11        Profesor prof2 = new Profesor("P2", "Luis Pérez", "Historia");
12        Profesor prof3 = new Profesor("P3", "Maria Gómez", "Programación");
13
14        Curso c1 = new Curso("C101", "Álgebra");
15        Curso c2 = new Curso("C102", "Historia Antigua");
16        Curso c3 = new Curso("C103", "Estructuras de Datos");
17        Curso c4 = new Curso("C104", "Literatura");
18        Curso c5 = new Curso("C105", "Bases de Datos");
19
20        // 2. Crear universidad y agregar
21        Universidad uni = new Universidad("Universidad Nacional");
22        uni.agregarProfesor(prof1);
23        uni.agregarProfesor(prof2);
24        uni.agregarProfesor(prof3);
25
26        uni.agregarCurso(c1);
27        uni.agregarCurso(c2);
28        uni.agregarCurso(c3);
29        uni.agregarCurso(c4);
30        uni.agregarCurso(c5);
31
32        // 3. Asignar profesores a cursos
33        uni.asignarProfesorACurso("C101", "P1");
34        uni.asignarProfesorACurso("C102", "P2");
35        uni.asignarProfesorACurso("C103", "P3");
36        uni.asignarProfesorACurso("C105", "P3");
37
```

```
38     // 4. Listar
39     uni.listarCursos();
40     uni.listarProfesores();
41
42     // 5. Cambiar profesor de un curso
43     System.out.println("\n\n\t Cambiando profesor del curso C103...");
44     uni.asignarProfesorACurso("C103", "P1");
45     uni.listarCursos();
46     uni.listarProfesores();
47
48     // 6. Remover un curso
49     System.out.println("\n\t Eliminando curso C102...");
50     uni.eliminarCurso("C102");
51     uni.listarCursos();
52     prof2.listarCursos();
53
54     // 7. Remover un profesor
55     System.out.println("\n\t Eliminando profesor P3...");
56     uni.eliminarProfesor("P3");
57     uni.listarCursos();
58     uni.listarProfesores();
59
60     // 8. Reporte
61     System.out.println();
62     uni.reporteCursosPorProfesor();
63 }
64 }
```

```
Output - Tp6Ejercicio3 (run) ×

▶ run:
Cursos en Universidad Nacional:
Curso: C101 - Álgebra | Profesor: Ana Torres
Curso: C102 - Historia Antigua | Profesor: Luis Pérez
Curso: C103 - Estructuras de Datos | Profesor: Marisa Gómez
Curso: C104 - Literatura | Profesor: [Sin asignar]
Curso: C105 - Bases de Datos | Profesor: Marisa Gómez
Profesores en Universidad Nacional:
Profesor: Ana Torres | ID: P1 | Especialidad: Matemática | Cursos: 1
Profesor: Luis Pérez | ID: P2 | Especialidad: Historia | Cursos: 1
Profesor: Marisa Gómez | ID: P3 | Especialidad: Programación | Cursos: 2

? Cambiando profesor del curso C103...
Cursos en Universidad Nacional:
Curso: C101 - Álgebra | Profesor: Ana Torres
Curso: C102 - Historia Antigua | Profesor: Luis Pérez
Curso: C103 - Estructuras de Datos | Profesor: Ana Torres
Curso: C104 - Literatura | Profesor: [Sin asignar]
Curso: C105 - Bases de Datos | Profesor: Marisa Gómez
Profesores en Universidad Nacional:
Profesor: Ana Torres | ID: P1 | Especialidad: Matemática | Cursos: 2
Profesor: Luis Pérez | ID: P2 | Especialidad: Historia | Cursos: 1
Profesor: Marisa Gómez | ID: P3 | Especialidad: Programación | Cursos: 1

Eliminando curso C102...
Cursos en Universidad Nacional:
Curso: C101 - Álgebra | Profesor: Ana Torres
Curso: C103 - Estructuras de Datos | Profesor: Ana Torres
Curso: C104 - Literatura | Profesor: [Sin asignar]
Curso: C105 - Bases de Datos | Profesor: Marisa Gómez
Cursos dictados por Luis Pérez:

Eliminando profesor P3...

Eliminando profesor P3...
Cursos en Universidad Nacional:
Curso: C101 - Álgebra | Profesor: Ana Torres
Curso: C103 - Estructuras de Datos | Profesor: Ana Torres
Curso: C104 - Literatura | Profesor: [Sin asignar]
Curso: C105 - Bases de Datos | Profesor: [Sin asignar]
Profesores en Universidad Nacional:
Profesor: Ana Torres | ID: P1 | Especialidad: Matemática | Cursos: 2
Profesor: Luis Pérez | ID: P2 | Especialidad: Historia | Cursos: 0

Reporte de cantidad de cursos por profesor:
Ana Torres: 2 cursos
Luis Pérez: 0 cursos
BUILD SUCCESSFUL (total time: 0 seconds)
```