

PROGRAMACIÓN II

Trabajo Práctico N.º 6: Colecciones

Estudiante: Emilia Gómez Juárez

Objetivo:

Desarrollar estructuras de datos dinámicas en Java mediante el uso de colecciones (ArrayList) y enumeraciones (enum), implementando un sistema de stock con funcionalidades progresivas que refuerzan conceptos clave de la programación orientada a objetos.

Caso práctico 1: Sistema Stock

Clase Producto

```
11 public class Producto {
12     private String id;
13     private String nombre;
14     private double precio;
15     private int cantidad;
16     private CategoriaProducto categoria;
17
18     public Producto(String id, String nombre, double precio, int cantidad, CategoriaProducto categoria) {
19         this.id = id;
20         this.nombre = nombre;
21         this.cantidad = cantidad;
22         this.categoria = categoria;
23         this.precio = precio;
24     }
25
26     public String getId() {
27         return id;
28     }
29
30     public void setId(String id) {
31         this.id = id;
32     }
33
34     public String getNombre() {
35         return nombre;
36     }
37
38     public void setNombre(String nombre) {
39         this.nombre = nombre;
40     }
41 }
```

PROGRAMACIÓN II

```
42     public double getPrecio() {
43         return precio;
44     }
45
46     public void setPrecio(double precio) {
47         this.precio = precio;
48     }
49
50     public int getCantidad() {
51         return cantidad;
52     }
53
54     public void setCantidad(int cantidad) {
55         this.cantidad = cantidad;
56     }
57
58     public CategoriaProducto getCategoria() {
59         return categoria;
60     }
61
62     public void setCategoria(CategoriaProducto categoria) {
63         this.categoria = categoria;
64     }
65
66
67
68     public void mostrarInfo(){
69         System.out.println("ID: " + id + "\nNombre: " + nombre + "\nPrecio: " + precio
70         + "\nCantidad: " + cantidad + "\nCategoría: " + categoria);
71     }
72 }
```

Clase Inventario

```
5     package SistemaStock;
6
7     import java.util.ArrayList;
8     import java.util.List;
9
10    /**
11     *
12     * @author Usuario
13     */
14    public class Inventario {
15
16        private ArrayList<Producto> productos;
17
18        public Inventario() {
19            productos = new ArrayList<>();
20        }
21
22        public void agregarProducto(Producto p) {
23            if (buscarProductoPorId(p.getId()) != null) {
24                System.out.println("No se puede agregar. Ya existe un producto con ID: " + p.getId());
25                return;
26            }
27            productos.add(p);
28            System.out.println("Producto agregado: " + p.getNombre());
29        }
30    }
```

PROGRAMACIÓN II

```
31 public void listarProductos() {
32     if (productos.isEmpty()) {
33         System.out.println("El inventario está vacío.");
34         return;
35     }
36     System.out.println("Listado de productos:");
37     for (Producto p : productos) {
38         p.mostrarInfo();
39     }
40 }
41
42 public Producto buscarProductoPorId(String id) {
43     for (Producto p : productos) {
44         if (p.getId().equalsIgnoreCase(id)) {
45             return p;
46         }
47     }
48     return null;
49 }
50
51 public boolean eliminarProducto(String id) {
52     Producto p = buscarProductoPorId(id);
53     if (p != null) {
54         productos.remove(p);
55         System.out.println("Producto eliminado: " + p.getNombre());
56         return true;
57     }
58     System.out.println("No se encontró producto con ID: " + id);
59     return false;
60 }
```

```
62 public boolean actualizarStock(String id, int nuevaCantidad) {
63     Producto p = buscarProductoPorId(id);
64     if (p != null) {
65         p.setCantidad(nuevaCantidad);
66         System.out.println("Stock actualizado para " + p.getNombre() + ": " + nuevaCantidad);
67         return true;
68     }
69     System.out.println("No se encontró producto con ID: " + id);
70     return false;
71 }
72
73 public List<Producto> filtrarPorCategoria(CategoriaProducto categoria) {
74     List<Producto> resultado = new ArrayList<>();
75     for (Producto p : productos) {
76         if (p.getCategoria() == categoria) {
77             resultado.add(p);
78         }
79     }
80     return resultado;
81 }
82
83 public int obtenerTotalStock() {
84     int total = 0;
85     for (Producto p : productos) {
86         total += p.getCantidad();
87     }
88     return total;
89 }
```

PROGRAMACIÓN II

```
91 public Producto obtenerProductoConMayorStock() {
92     if (productos.isEmpty()) {
93         return null;
94     }
95     Producto mayor = productos.get(0);
96     for (Producto p : productos) {
97         if (p.getCantidad() > mayor.getCantidad()) {
98             mayor = p;
99         }
100     }
101     return mayor;
102 }
103
104 public List<Producto> filtrarProductosPorPrecio(double min, double max) {
105     List<Producto> resultado = new ArrayList<>();
106     for (Producto p : productos) {
107         if (p.getPrecio() >= min && p.getPrecio() <= max) {
108             resultado.add(p);
109         }
110     }
111     return resultado;
112 }
113
114 public void mostrarCategoriasDisponibles() {
115     System.out.println("Categorías disponibles:");
116     for (CategoriaProducto c : CategoriaProducto.values()) {
117         System.out.println(c + " - " + c.getDescripcion());
118     }
119 }
120 }
```

Enum Categoría Producto

```
11 public enum CategoriaProducto {
12     ALIMENTOS("Productos comestibles"),
13     ELECTRONICA("Dispositivos electrónicos"),
14     ROPA("Prendas de vestir"), HOGAR("Artículos para el hogar");
15     private final String descripcion;
16
17     CategoriaProducto(String descripcion) {
18         this.descripcion = descripcion;
19     }
20
21     public String getDescripcion() {
22         return descripcion;
23     }
24 }
25 }
```

Main

PROGRAMACIÓN II

```
13 public class MainStock {
14
15     public static void main(String[] args) {
16
17         Inventario inventario = new Inventario();
18
19         Producto p1 = new Producto("P001", "Arroz 1kg", 850.0, 30, CategoriaProducto.ALIMENTOS);
20         Producto p2 = new Producto("P002", "Auriculares Bluetooth", 2999.0, 12, CategoriaProducto.ELECTRONICA);
21         Producto p3 = new Producto("P003", "Remera algodón", 2200.0, 25, CategoriaProducto.ROPA);
22         Producto p4 = new Producto("P004", "Set de sábanas", 4200.0, 8, CategoriaProducto.HOGAR);
23         Producto p5 = new Producto("P005", "Leche entera 1L", 450.0, 50, CategoriaProducto.ALIMENTOS);
24
25         inventario.agregarProducto(p1);
26         inventario.agregarProducto(p2);
27         inventario.agregarProducto(p3);
28         inventario.agregarProducto(p4);
29         inventario.agregarProducto(p5);
30
31         System.out.println();
32
33         System.out.println("Tarea 2: Listar todos los productos");
34         inventario.listarProductos();
35
36         System.out.println("Tarea 3: Buscar producto por ID 'P002'");
37         Producto buscado = inventario.buscarProductoPorId("P002");
38         if (buscado != null) {
39             buscado.mostrarInfo();
40         } else {
41             System.out.println("Producto no encontrado.");
42         }
43     }
44 }
```

```
44         System.out.println("Tarea 4: Filtrar por categoría ELECTRONICA");
45         List<Producto> electronica = inventario.filtrarPorCategoria(CategoriaProducto.ELECTRONICA);
46         for (Producto p : electronica) {
47             p.mostrarInfo();
48         }
49
50         System.out.println("Tarea 5: Eliminar producto 'P003' y listar restantes");
51         inventario.eliminarProducto("P003");
52         inventario.listarProductos();
53
54         System.out.println("Tarea 6: Actualizar stock de 'P001' a 20");
55         inventario.actualizarStock("P001", 20);
56         Producto pActualizado = inventario.buscarProductoPorId("P001");
57         if (pActualizado != null) {
58             pActualizado.mostrarInfo();
59         }
60
61         System.out.println("Tarea 7: Total de stock disponible: " + inventario.obtenerTotalStock());
62
63         System.out.println("Tarea 8: Producto con mayor stock:");
64         Producto mayor = inventario.obtenerProductoConMayorStock();
65         if (mayor != null) {
66             mayor.mostrarInfo();
67         }
68         System.out.println("Tarea 9: Productos con precio entre $1000 y $3000");
69         List<Producto> rango = inventario.filtrarProductosPorPrecio(1000.0, 3000.0);
70         for (Producto p : rango) {
71             p.mostrarInfo();
72         }
73     }
74 }
```

```
75         System.out.println("Tarea 10: Categorías disponibles:");
76         inventario.mostrarCategoriasDisponibles();
77     }
78 }
```

Resultados:

Tarea 1:

PROGRAMACIÓN II

```
Producto agregado: Arroz 1kg  
Producto agregado: Auriculares Bluetooth  
Producto agregado: Remera algodón  
Producto agregado: Set de sábanas  
Producto agregado: Leche entera 1L
```

Tarea 2:

```
Tarea 2: Listar todos los productos  
Listado de productos:  
ID: P001  
Nombre: Arroz 1kg  
Precio: 850.0  
Cantidad: 30  
Categoría: ALIMENTOS  
ID: P002  
Nombre: Auriculares Bluetooth  
Precio: 2999.0  
Cantidad: 12  
Categoría: ELECTRONICA  
ID: P003  
Nombre: Remera algodón  
Precio: 2200.0  
Cantidad: 25  
Categoría: ROPA  
ID: P004  
Nombre: Set de sábanas  
Precio: 4200.0  
Cantidad: 8  
Categoría: HOGAR  
ID: P005  
Nombre: Leche entera 1L  
Precio: 450.0  
Cantidad: 50  
Categoría: ALIMENTOS
```

Tarea 3 y 4:

PROGRAMACIÓN II

```
Tarea 3: Buscar producto por ID 'P002'
ID: P002
Nombre: Auriculares Bluetooth
Precio: 2999.0
Cantidad: 12
Categoría: ELECTRONICA
Tarea 4: Filtrar por categoría ELECTRONICA
ID: P002
Nombre: Auriculares Bluetooth
Precio: 2999.0
Cantidad: 12
Categoría: ELECTRONICA
```

Tarea 5:

```
Tarea 5: Eliminar producto 'P003' y listar restantes
Producto eliminado: Remera algodón
Listado de productos:
ID: P001
Nombre: Arroz 1kg
Precio: 850.0
Cantidad: 30
Categoría: ALIMENTOS
ID: P002
Nombre: Auriculares Bluetooth
Precio: 2999.0
Cantidad: 12
Categoría: ELECTRONICA
ID: P004
Nombre: Set de sábanas
Precio: 4200.0
Cantidad: 8
Categoría: HOGAR
ID: P005
Nombre: Leche entera 1L
Precio: 450.0
Cantidad: 50
Categoría: ALIMENTOS
```

Tareas 6 – 10

PROGRAMACIÓN II

```
Tarea 6: Actualizar stock de 'P001' a 20
Stock actualizado para Arroz 1kg: 20
ID: P001
Nombre: Arroz 1kg
Precio: 850.0
Cantidad: 20
Categoría: ALIMENTOS
Tarea 7: Total de stock disponible: 90
Tarea 8: Producto con mayor stock:
ID: P005
Nombre: Leche entera 1L
Precio: 450.0
Cantidad: 50
Categoría: ALIMENTOS
Tarea 9: Productos con precio entre $1000 y $3000
ID: P002
Nombre: Auriculares Bluetooth
Precio: 2999.0
Cantidad: 12
Categoría: ELECTRONICA
Tarea 10: Categorías disponibles:
Categorías disponibles:
ALIMENTOS - Productos comestibles
ELECTRONICA - Dispositivos electrónicos
ROPA - Prendas de vestir
HOGAR - Artículos para el hogar
```

Caso práctico 2: Gestión Biblioteca

Clase Autor

PROGRAMACIÓN II

```
11 public class Autor {
12
13     private String id;
14     private String nombre;
15     private String nacionalidad;
16
17     public Autor(String id, String nombre, String nacionalidad) {
18         this.id = id;
19         this.nombre = nombre;
20         this.nacionalidad = nacionalidad;
21     }
22
23     public String getId() {
24         return id;
25     }
26
27     public String getNombre() {
28         return nombre;
29     }
30
31     public String getNacionalidad() {
32         return nacionalidad;
33     }
34
35     public void mostrarInfo() {
36         System.out.println("Autor: " + nombre + " (" + nacionalidad + ") - ID: " + id);
37     }
38
39     @Override
40     public String toString() {
41         return nombre + " (" + nacionalidad + ")";
42     }
43 }
```

Clase Libro

PROGRAMACIÓN II

```
11 public class Libro {  
12     private String isbn;  
13     private String titulo;  
14     private int anioPublicacion;  
15     private Autor autor;  
16  
17     public Libro(String isbn, String titulo, int anioPublicacion, Autor autor) {  
18         this.isbn = isbn;  
19         this.titulo = titulo;  
20         this.anioPublicacion = anioPublicacion;  
21         this.autor = autor;  
22     }  
23     public String getIsbn() {  
24         return isbn;  
25     }  
26     public String getTitulo() {  
27         return titulo;  
28     }  
29     public int getAnioPublicacion() {  
30         return anioPublicacion;  
31     }  
32     public Autor getAutor() {  
33         return autor;  
34     }  
35     public void mostrarInfo() {  
36         System.out.println("Titulo: " + titulo);  
37         System.out.println("ISBN: " + isbn);  
38         System.out.println("Año de publicación: " + anioPublicacion);  
39         System.out.println("Autor: " + autor.getNombre() + " - " + autor.getNacionalidad());  
40         System.out.println("-----");  
41     }  
42 }
```

Clase Biblioteca

PROGRAMACIÓN II

```
14 public class Biblioteca {
15
16     private String nombre;
17     private List<Libro> libros;
18
19     public Biblioteca(String nombre) {
20         this.nombre = nombre;
21         this.libros = new ArrayList<>();
22     }
23
24     public void agregarLibro(String isbn, String titulo, int anioPublicacion, Autor autor) {
25         if (buscarLibroPorIsbn(isbn) != null) {
26             System.out.println("Ya existe un libro con el ISBN: " + isbn);
27             return;
28         }
29         libros.add(new Libro(isbn, titulo, anioPublicacion, autor));
30         System.out.println("Libro agregado: " + titulo);
31     }
32
33     public void listarLibros() {
34         if (libros.isEmpty()) {
35             System.out.println("La biblioteca no tiene libros.");
36             return;
37         }
38
39         System.out.println("Listado de libros en la biblioteca '" + nombre + "':");
40         for (Libro l : libros) {
41             l.mostrarInfo();
42         }
43     }
44 }
```

```
45     public Libro buscarLibroPorIsbn(String isbn) {
46         for (Libro l : libros) {
47             if (l.getIsbn().equalsIgnoreCase(isbn)) {
48                 return l;
49             }
50         }
51         return null;
52     }
53
54     public boolean eliminarLibro(String isbn) {
55         Libro l = buscarLibroPorIsbn(isbn);
56         if (l != null) {
57             libros.remove(l);
58             System.out.println("Libro eliminado: " + l.getTitulo());
59             return true;
60         }
61         System.out.println("No se encontró un libro con ISBN: " + isbn);
62         return false;
63     }
64
65     public int obtenerCantidadLibros() {
66         return libros.size();
67     }
68 }
```

PROGRAMACIÓN II

```
69 public List<Libro> filtrarLibrosPorAnio(int anio) {
70     List<Libro> resultado = new ArrayList<>();
71     for (Libro l : libros) {
72         if (l.getAnioPublicacion() == anio) {
73             resultado.add(l);
74         }
75     }
76     return resultado;
77 }
78
79 public void mostrarAutoresDisponibles() {
80     System.out.println("Autores disponibles en la biblioteca:");
81     List<String> nombres = new ArrayList<>();
82     for (Libro l : libros) {
83         if (!nombres.contains(l.getAutor().getNombre())) {
84             nombres.add(l.getAutor().getNombre());
85             l.getAutor().mostrarInfo();
86         }
87     }
88 }
89 }
90
```

Main

```
14 public class MainGestionBiblioteca {
15
16     public static void main(String[] args) {
17         Scanner sc = new Scanner(System.in);
18         Biblioteca biblioteca = new Biblioteca("Biblioteca Central");
19
20         Autor a1 = new Autor("A001", "Gabriel García Márquez", "Colombiana");
21         Autor a2 = new Autor("A002", "J.K. Rowling", "Británica");
22         Autor a3 = new Autor("A003", "Ernest Hemingway", "Estadounidense");
23
24         boolean salir = false;
25
26         while (!salir) {
27             System.out.println("\n===== MENÚ BIBLIOTECA =====");
28             System.out.println("1. Agregar libro");
29             System.out.println("2. Listar libros");
30             System.out.println("3. Buscar libro por ISBN");
31             System.out.println("4. Filtrar libros por año");
32             System.out.println("5. Eliminar libro");
33             System.out.println("6. Mostrar cantidad total de libros");
34             System.out.println("7. Mostrar autores disponibles");
35             System.out.println("8. Salir");
36             System.out.print("Seleccione una opción: ");
37
38             int opcion = sc.nextInt();
39             sc.nextLine();

```

PROGRAMACIÓN II

```
42     switch (opcion) {
43     case 1:
44         System.out.print("Ingrese ISBN: ");
45         String isbn = sc.nextLine();
46         System.out.print("Ingrese título: ");
47         String titulo = sc.nextLine();
48         System.out.print("Ingrese año de publicación: ");
49         int anio = sc.nextInt();
50         sc.nextLine();
51         System.out.println("Seleccione autor (1: Márquez, 2: Rowling, 3: Hemingway): ");
52         int autorSel = sc.nextInt();
53         sc.nextLine();
54         Autor autor = switch (autorSel) {
55             case 1 ->
56                 a1;
57             case 2 ->
58                 a2;
59             case 3 ->
60                 a3;
61             default ->
62                 a1;
63         };
64         biblioteca.agregarLibro(isbn, titulo, anio, autor);
65         break;
66     case 2:
67         biblioteca.listarLibros();
68         break;
69 
```

```
70
71     case 3:
72         System.out.print("Ingrese ISBN a buscar: ");
73         String buscar = sc.nextLine();
74         Libro encontrado = biblioteca.buscarLibroPorIsbn(buscar);
75         if (encontrado != null) {
76             encontrado.mostrarInfo();
77         } else {
78             System.out.println("Libro no encontrado.");
79         }
80         break;
81     case 4:
82         System.out.print("Ingrese año de publicación a filtrar: ");
83         int anioFiltro = sc.nextInt();
84         sc.nextLine();
85         List<Libro> filtrados = biblioteca.filtrarLibrosPorAnio(anioFiltro);
86         if (filtrados.isEmpty()) {
87             System.out.println("No hay libros de ese año.");
88         } else {
89             for (Libro l : filtrados) {
90                 l.mostrarInfo();
91             }
92         }
93         break;
94     case 5:
95         System.out.print("Ingrese ISBN del libro a eliminar: ");
96         String eliminar = sc.nextLine();
97         biblioteca.eliminarLibro(eliminar);
98         break;
99 
```

PROGRAMACIÓN II

```
101         case 6:
102             System.out.println("Cantidad total de libros: " + biblioteca.obtenerCantidadLibros());
103             break;
104
105         case 7:
106             biblioteca.mostrarAutoresDisponibles();
107             break;
108
109         case 8:
110             salir = true;
111             System.out.println("Saliendo del sistema... ¡Hasta pronto!");
112             break;
113
114         default:
115             System.out.println("Opción no válida.");
116     }
117 }
118 }
119 }
```

Resultados

Tarea 1 y 2

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

Autor a1 = new Autor("A001", "Gabriel García Márquez", "Colombiana");
Autor a2 = new Autor("A002", "J.K. Rowling", "Británica");
Autor a3 = new Autor("A003", "Ernest Hemingway", "Estadounidense");
```

Tarea 3

PROGRAMACIÓN II

```
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 1
Ingrese ISBN: 9780307474728
Ingrese título: Cien años de soledad
Ingrese año de publicación: 1967
Seleccione autor (1: Márquez, 2: Rowling, 3: Hemingway):
1
Libro agregado: Cien años de soledad

===== MENÚ BIBLIOTECA =====
1. Agregar libro
2. Listar libros
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 1
Ingrese ISBN: 9780684801223
Ingrese título: El viejo y el mar
Ingrese año de publicación: 1952
Seleccione autor (1: Márquez, 2: Rowling, 3: Hemingway):
3
Libro agregado: El viejo y el mar
```

PROGRAMACIÓN II

```
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 1
Ingrese ISBN: 9780307389732
Ingrese título: Amor en tiempos de colera
Ingrese año de publicación: 1985
Seleccione autor (1: Márquez, 2: Rowling, 3: Hemingway):
1
Libro agregado: Amor en tiempos de colera

===== MENÚ BIBLIOTECA =====
1. Agregar libro
2. Listar libros
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 1
Ingrese ISBN: 9780747532699
Ingrese título: Harry Potter y la piedra filosofal
Ingrese año de publicación: 1997
Seleccione autor (1: Márquez, 2: Rowling, 3: Hemingway):
2
Libro agregado: Harry Potter y la piedra filosofal
```

```
===== MENÚ BIBLIOTECA =====
1. Agregar libro
2. Listar libros
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 1
Ingrese ISBN: 9780747546290
Ingrese título: Harry Potter y el prisionero de Azkaban
Ingrese año de publicación: 1999
Seleccione autor (1: Márquez, 2: Rowling, 3: Hemingway):
2
Libro agregado: Harry Potter y el prisionero de Azkaban
```


PROGRAMACIÓN II

```
Listado de libros en la biblioteca 'Biblioteca Central':
Título: Cien años de soledad
ISBN: 9780307474728
Año de publicación: 1967
Autor: Gabriel García Márquez - Colombiana
-----
Título: El viejo y el mar
ISBN: 9780684801223
Año de publicación: 1952
Autor: Ernest Hemingway - Estadounidense
-----
Título: Amor en tiempos de colera
ISBN: 9780307389732
Año de publicación: 1985
Autor: Gabriel García Márquez - Colombiana
-----
Título: Harry Potter y la piedra filosofal
ISBN: 9780747532699
Año de publicación: 1997
Autor: J.K. Rowling - Británica
-----
Título: Harry Potter y el prisionero de Azkaban
ISBN: 9780747546290
Año de publicación: 1999
Autor: J.K. Rowling - Británica
```

Tarea 5

```
===== MENÚ BIBLIOTECA =====
1. Agregar libro
2. Listar libros
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción:
3
Ingrese ISBN a buscar: 9780684801223
Título: El viejo y el mar
ISBN: 9780684801223
Año de publicación: 1952
Autor: Ernest Hemingway - Estadounidense
-----
```

Tarea 6

PROGRAMACIÓN II

```
===== MENÚ BIBLIOTECA =====
1. Agregar libro
2. Listar libros
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 4
Ingrese año de publicación a filtrar: 1999
Título: Harry Potter y el prisionero de Azkaban
ISBN: 9780747546290
Año de publicación: 1999
Autor: J.K. Rowling - Británica
```

Tarea 7

```
===== MENÚ BIBLIOTECA =====
1. Agregar libro
2. Listar libros
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 5
Ingrese ISBN del libro a eliminar: 9780307474728
Libro eliminado: Cien años de soledad
```

PROGRAMACIÓN II

```
Seleccione una opción: 2
Listado de libros en la biblioteca 'Biblioteca Central':
Título: El viejo y el mar
ISBN: 9780684801223
Año de publicación: 1952
Autor: Ernest Hemingway - Estadounidense
-----
Título: Amor en tiempos de colera
ISBN: 9780307389732
Año de publicación: 1985
Autor: Gabriel García Márquez - Colombiana
-----
Título: Harry Potter y la piedra filosofal
ISBN: 9780747532699
Año de publicación: 1997
Autor: J.K. Rowling - Británica
-----
Título: Harry Potter y el prisionero de Azkaban
ISBN: 9780747546290
Año de publicación: 1999
Autor: J.K. Rowling - Británica
-----
```

Tarea 8

```
===== MENÚ BIBLIOTECA =====
1. Agregar libro
2. Listar libros
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 6
Cantidad total de libros: 4
```

Tarea 9

PROGRAMACIÓN II

```
===== MENÚ BIBLIOTECA =====
1. Agregar libro
2. Listar libros
3. Buscar libro por ISBN
4. Filtrar libros por año
5. Eliminar libro
6. Mostrar cantidad total de libros
7. Mostrar autores disponibles
8. Salir
Seleccione una opción: 7
Autores disponibles en la biblioteca:
Autor: Ernest Hemingway (Estadounidense) - ID: A003
Autor: Gabriel García Márquez (Colombiana) - ID: A001
Autor: J.K. Rowling (Británica) - ID: A002
```

Caso práctico 3: Sistema Académico

Clase Curso

```
11 public class Curso {
12
13     private String codigo;
14     private String nombre;
15     private Profesor profesor;
16
17     public Curso(String codigo, String nombre) {
18         this.codigo = codigo;
19         this.nombre = nombre;
20         this.profesor = null;
21     }
22     public String getCodigo() {
23         return codigo;
24     }
25     public String getNombre() {
26         return nombre;
27     }
28     public Profesor getProfesor() {
29         return profesor;
30     }
}
```

PROGRAMACIÓN II

```
31 public void setProfesor(Profesor p) {
32     if (this.profesor == p) {
33         return;
34     }
35     if (this.profesor != null) {
36         Profesor previo = this.profesor;
37         this.profesor = null;
38         if (previo.getCursos().contains(this)) {
39             previo.getCursos().remove(this);
40         }
41     }
42     this.profesor = p;
43
44     if (p != null && !p.getCursos().contains(this)) {
45         p.getCursos().add(this);
46     }
47 }
48
49 public void mostrarInfo() {
50     System.out.println("Código: " + codigo);
51     System.out.println("Nombre: " + nombre);
52     System.out.println("Profesor: " + (profesor != null ? profesor.getNombre() : "(sin asignar)"));
53     System.out.println("-----");
54 }
55
56 @Override
57 public String toString() {
58     return codigo + " - " + nombre + " (Profesor: " + (profesor != null ? profesor.getNombre() : "-") + ")";
59 }
60 }
```

Clase Profesor

PROGRAMACIÓN II

```
14 public class Profesor {
15
16     private String id;
17     private String nombre;
18     private String especialidad;
19     private List<Curso> cursos;
20
21     public Profesor(String id, String nombre, String especialidad) {
22         this.id = id;
23         this.nombre = nombre;
24         this.especialidad = especialidad;
25         this.cursos = new ArrayList<>();
26     }
27
28     public String getId() {
29         return id;
30     }
31
32     public String getNombre() {
33         return nombre;
34     }
35
36     public String getEspecialidad() {
37         return especialidad;
38     }
39
40     public List<Curso> getCursos() {
41         return cursos;
42     }
}
```

PROGRAMACIÓN II

```
44 public void agregarCurso(Curso c) {
45     if (c == null) {
46         return;
47     }
48     if (!cursos.contains(c)) {
49         cursos.add(c);
50     }
51     if (c.getProfesor() != this) {
52         c.setProfesor(this);
53     }
54 }
55
56 public void eliminarCurso(Curso c) {
57     if (c == null) {
58         return;
59     }
60     if (cursos.contains(c)) {
61         cursos.remove(c);
62     }
63     if (c.getProfesor() == this) {
64         c.setProfesor(null);
65     }
66 }
```

```
68 public void listarCursos() {
69     if (cursos.isEmpty()) {
70         System.out.println("\tNo dicta cursos por el momento.");
71         return;
72     }
73     for (Curso c : cursos) {
74         System.out.println("\t" + c.getCodigo() + " - " + c.getNombre());
75     }
76 }
77
78 public void mostrarInfo() {
79     System.out.println("ID: " + id);
80     System.out.println("Nombre: " + nombre);
81     System.out.println("Especialidad: " + especialidad);
82     System.out.println("Cantidad de cursos: " + cursos.size());
83     System.out.println("Cursos:");
84     listarCursos();
85     System.out.println("-----");
86 }
87
88 @Override
89 public String toString() {
90     return nombre + " (" + especialidad + ")";
91 }
92 }
93 }
```

PROGRAMACIÓN II

Clase Universidad

```
14 public class Universidad {
15
16     private String nombre;
17     private List<Profesor> profesores;
18     private List<Curso> cursos;
19
20     public Universidad(String nombre) {
21         this.nombre = nombre;
22         this.profesores = new ArrayList<>();
23         this.cursos = new ArrayList<>();
24     }
25
26     public void agregarProfesor(Profesor p) {
27         if (buscarProfesorPorId(p.getId()) != null) {
28             System.out.println("Ya existe un profesor con ID: " + p.getId());
29             return;
30         }
31         profesores.add(p);
32         System.out.println("Profesor agregado: " + p.getNombre());
33     }
34
35     public void agregarCurso(Curso c) {
36         if (buscarCursoPorCodigo(c.getCodigo()) != null) {
37             System.out.println("Ya existe un curso con código: " + c.getCodigo());
38             return;
39         }
40         cursos.add(c);
41         System.out.println("Curso agregado: " + c.getNombre());
42     }

```

```
44     public Profesor buscarProfesorPorId(String id) {
45         for (Profesor p : profesores) {
46             if (p.getId().equalsIgnoreCase(id)) {
47                 return p;
48             }
49         }
50         return null;
51     }
52
53     public Curso buscarCursoPorCodigo(String codigo) {
54         for (Curso c : cursos) {
55             if (c.getCodigo().equalsIgnoreCase(codigo)) {
56                 return c;
57             }
58         }
59         return null;
60     }

```


PROGRAMACIÓN II

```
62 public void asignarProfesorACurso(String codigoCurso, String idProfesor) {
63     Curso curso = buscarCursoPorCodigo(codigoCurso);
64     Profesor profesor = buscarProfesorPorId(idProfesor);
65     if (curso == null) {
66         System.out.println("No se encontró el curso con código: " + codigoCurso);
67         return;
68     }
69     if (profesor == null) {
70         System.out.println("No se encontró el profesor con ID: " + idProfesor);
71         return;
72     }
73     curso.setProfesor(profesor);
74     System.out.println("Profesor " + profesor.getNombre() + " asignado al curso " + curso.getNombre());
75 }
76
77 public void listarProfesores() {
78     if (profesores.isEmpty()) {
79         System.out.println("No hay profesores registrados.");
80         return;
81     }
82     System.out.println("Lista de profesores:");
83     for (Profesor p : profesores) {
84         p.mostrarInfo();
85     }
86 }
87
```

```
88 public void listarCursos() {
89     if (cursos.isEmpty()) {
90         System.out.println("No hay cursos registrados.");
91         return;
92     }
93     System.out.println("Lista de cursos:");
94     for (Curso c : cursos) {
95         c.mostrarInfo();
96     }
97 }
98
99 public boolean eliminarCurso(String codigo) {
100     Curso c = buscarCursoPorCodigo(codigo);
101     if (c == null) {
102         System.out.println("No se encontró curso con código: " + codigo);
103         return false;
104     }
105
106     if (c.getProfesor() != null) {
107         Profesor prof = c.getProfesor();
108         prof.getCursos().remove(c);
109         c.setProfesor(null);
110     }
111     cursos.remove(c);
112     System.out.println("Curso eliminado: " + codigo);
113     return true;
114 }
```

PROGRAMACIÓN II

```
116 public boolean eliminarProfesor(String id) {
117     Profesor p = buscarProfesorPorId(id);
118     if (p == null) {
119         System.out.println("No se encontró profesor con ID: " + id);
120         return false;
121     }
122     for (Curso c : new ArrayList<>(p.getCursos())) {
123         c.setProfesor(null);
124     }
125     profesores.remove(p);
126     System.out.println("Profesor eliminado: " + p.getNombre());
127     return true;
128 }
129
130 public void reporteCursosPorProfesor() {
131     System.out.println("Reporte: cantidad de cursos por profesor");
132     for (Profesor p : profesores) {
133         System.out.println(p.getNombre() + " -> " + p.getCursos().size() + " curso(s)");
134     }
135 }
136
137 }
```

Main

```
13 public class Main {
14
15     public static void main(String[] args) {
16         Scanner sc = new Scanner(System.in);
17         Universidad uni = new Universidad("Universidad Nacional");
18
19         // 1. Crear al menos 3 profesores y 5 cursos precargados
20         Profesor pr1 = new Profesor("P001", "Ana López", "Matemáticas");
21         Profesor pr2 = new Profesor("P002", "Carlos Pérez", "Programación");
22         Profesor pr3 = new Profesor("P003", "Lucía Gómez", "Física");
23
24         Curso c1 = new Curso("C101", "Álgebra I");
25         Curso c2 = new Curso("C102", "Programación I");
26         Curso c3 = new Curso("C103", "Física I");
27         Curso c4 = new Curso("C104", "Estructuras de Datos");
28         Curso c5 = new Curso("C105", "Cálculo Diferencial");
29
30         uni.agregarProfesor(pr1);
31         uni.agregarProfesor(pr2);
32         uni.agregarProfesor(pr3);
33
34         uni.agregarCurso(c1);
35         uni.agregarCurso(c2);
36         uni.agregarCurso(c3);
37         uni.agregarCurso(c4);
38         uni.agregarCurso(c5);
39
40         boolean salir = false;
41     }
42 }
```

PROGRAMACIÓN II

```
42 do {
43     System.out.println("\n===== MENÚ UNIVERSIDAD =====");
44     System.out.println("1. Agregar profesor");
45     System.out.println("2. Agregar curso");
46     System.out.println("3. Asignar profesor a curso");
47     System.out.println("4. Listar profesores");
48     System.out.println("5. Listar cursos");
49     System.out.println("6. Buscar profesor por ID");
50     System.out.println("7. Buscar curso por código");
51     System.out.println("8. Eliminar curso");
52     System.out.println("9. Eliminar profesor");
53     System.out.println("10. Salir");
54     System.out.print("Seleccione una opción: ");
55
56     int opcion = sc.nextInt();
57     sc.nextLine();
```

```
60 switch (opcion) {
61     case 1:
62         System.out.print("Ingrese ID del profesor: ");
63         String idP = sc.nextLine();
64         System.out.print("Ingrese nombre: ");
65         String nombreP = sc.nextLine();
66         System.out.print("Ingrese especialidad: ");
67         String esp = sc.nextLine();
68         uni.agregarProfesor(new Profesor(idP, nombreP, esp));
69         break;
70     case 2:
71         System.out.print("Ingrese código del curso: ");
72         String cod = sc.nextLine();
73         System.out.print("Ingrese nombre del curso: ");
74         String nomC = sc.nextLine();
75         uni.agregarCurso(new Curso(cod, nomC));
76         break;
77     case 3:
78
79         System.out.print("Ingrese código del curso: ");
80         String codCurso = sc.nextLine();
81         System.out.print("Ingrese ID del profesor: ");
82         String idProfesor = sc.nextLine();
83         uni.asignarProfesorACurso(codCurso, idProfesor);
84         break;
85     case 4:
86         uni.listarProfesores();
87         break;
```

PROGRAMACIÓN II

```
91         case 5:
92             uni.listarCursos();
93             break;
94
95         case 6:
96             System.out.print("Ingrese ID del profesor a buscar: ");
97             String busqP = sc.nextLine();
98             Profesor encontradoP = uni.buscarProfesorPorId(busqP);
99             if (encontradoP != null) {
100                 encontradoP.mostrarInfo();
101             } else {
102                 System.out.println("Profesor no encontrado.");
103             }
104             break;
105
106         case 7:
107             System.out.print("Ingrese código del curso a buscar: ");
108             String busqC = sc.nextLine();
109             Curso encontradoC = uni.buscarCursoPorCodigo(busqC);
110             if (encontradoC != null) {
111                 encontradoC.mostrarInfo();
112             } else {
113                 System.out.println("Curso no encontrado.");
114             }
115             break;
116
117         case 8:
118             System.out.print("Ingrese código del curso a eliminar: ");
119             String elimC = sc.nextLine();
120             uni.eliminarCurso(elimC);
121             break;
122
123         case 9:
124             System.out.print("Ingrese ID del profesor a eliminar: ");
125             String elimP = sc.nextLine();
126             uni.eliminarProfesor(elimP);
127             break;
128
129         case 10:
130             salir = true;
131             System.out.println("Saliendo... ¡hasta pronto!");
132             break;
133
134         default:
135             System.out.println("Opción inválida.");
136     }
137
138     } while (!salir);
139 }
140
141
```

Resultados

Tarea 1 y 2

PROGRAMACIÓN II

```
Profesor pr1 = new Profesor("P001", "Ana López", "Matemáticas");
Profesor pr2 = new Profesor("P002", "Carlos Pérez", "Programación");
Profesor pr3 = new Profesor("P003", "Lucía Gómez", "Física");

Curso c1 = new Curso("C101", "Álgebra I");
Curso c2 = new Curso("C102", "Programación I");
Curso c3 = new Curso("C103", "Física I");
Curso c4 = new Curso("C104", "Estructuras de Datos");
Curso c5 = new Curso("C105", "Cálculo Diferencial");

uni.agregarProfesor(pr1);
uni.agregarProfesor(pr2);
uni.agregarProfesor(pr3);

uni.agregarCurso(c1);
uni.agregarCurso(c2);
uni.agregarCurso(c3);
uni.agregarCurso(c4);
uni.agregarCurso(c5);
```

```
Profesor agregado: Ana López
Profesor agregado: Carlos Pérez
Profesor agregado: Lucía Gómez
Curso agregado: Álgebra I
Curso agregado: Programación I
Curso agregado: Física I
Curso agregado: Estructuras de Datos
Curso agregado: Cálculo Diferencial
```

Tarea 3

```
Seleccione una opción: 3
Ingrese código del curso: C101
Ingrese ID del profesor: P001
Profesor Ana López asignado al curso Álgebra I
```

```
Seleccione una opción: 3
Ingrese código del curso: C102
Ingrese ID del profesor: P003
Profesor Lucía Gómez asignado al curso Programación I
```

```
Seleccione una opción: 3
Ingrese código del curso: C103
Ingrese ID del profesor: P002
Profesor Carlos Pérez asignado al curso Física I
```

PROGRAMACIÓN II

```
Seleccione una opción: 3
Ingrese código del curso: C104
Ingrese ID del profesor: P001
Profesor Ana López asignado al curso Estructuras de Datos
```

```
Seleccione una opción: 3
Ingrese código del curso: C105
Ingrese ID del profesor: P003
Profesor Lucía Gómez asignado al curso Cálculo Diferencial
```

Tarea 4

```
Seleccione una opción: 4
Lista de profesores:
ID: P001
Nombre: Ana López
Especialidad: Matemáticas
Cantidad de cursos: 2
Cursos:
    C101 - Álgebra I
    C104 - Estructuras de Datos
-----
ID: P002
Nombre: Carlos Pérez
Especialidad: Programación
Cantidad de cursos: 1
Cursos:
    C103 - Física I
-----
ID: P003
Nombre: Lucía Gómez
Especialidad: Física
Cantidad de cursos: 2
Cursos:
    C105 - Cálculo Diferencial
    C102 - Programación I
-----
```

```
Lista de cursos:
Código: C101
Nombre: Álgebra I
Profesor: Ana López
-----
Código: C102
Nombre: Programación I
Profesor: Lucía Gómez
-----
Código: C103
Nombre: Física I
Profesor: Carlos Pérez
-----
Código: C104
Nombre: Estructuras de Datos
Profesor: Ana López
-----
Código: C105
Nombre: Cálculo Diferencial
Profesor: Lucía Gómez
-----
```

Tarea 5

```
Seleccione una opción: 3
Ingrese código del curso: C102
Ingrese ID del profesor: P002
Profesor Carlos Pérez asignado al curso Programación I
```

PROGRAMACIÓN II

```
Lista de profesores:
ID: P001
Nombre: Ana López
Especialidad: Matemáticas
Cantidad de cursos: 1
Cursos:
    C104 - Estructuras de Datos
-----
ID: P002
Nombre: Carlos Pérez
Especialidad: Programación
Cantidad de cursos: 2
Cursos:
    C103 - Física I
    C102 - Programación I
-----
ID: P003
Nombre: Lucía Gómez
Especialidad: Física
Cantidad de cursos: 1
Cursos:
    C105 - Cálculo Diferencial
-----
```

```
Lista de cursos:
Código: C102
Nombre: Programación I
Profesor: Carlos Pérez
-----
Código: C103
Nombre: Física I
Profesor: Carlos Pérez
-----
Código: C104
Nombre: Estructuras de Datos
Profesor: Ana López
-----
Código: C105
Nombre: Cálculo Diferencial
Profesor: Lucía Gómez
-----
```

Tarea 6

PROGRAMACIÓN II

```
===== MENÚ UNIVERSIDAD =====  
1. Agregar profesor  
2. Agregar curso  
3. Asignar profesor a curso  
4. Listar profesores  
5. Listar cursos  
6. Buscar profesor por ID  
7. Buscar curso por código  
8. Eliminar curso  
9. Eliminar profesor  
10. Salir  
Seleccione una opción: 8  
Ingrese código del curso a eliminar: C101  
Curso eliminado: C101
```

```
6. Buscar profesor por ID  
7. Buscar curso por código  
8. Eliminar curso  
9. Eliminar profesor  
10. Salir  
Seleccione una opción: 6  
Ingrese ID del profesor a buscar: P001  
ID: P001  
Nombre: Ana López  
Especialidad: Matemáticas  
Cantidad de cursos: 1  
Cursos:  
    C104 - Estructuras de Datos  
-----
```

Tarea 7

PROGRAMACIÓN II

```
9. Eliminar profesor
10. Salir
Seleccione una opción: 9
Ingrese ID del profesor a eliminar: P001
Profesor eliminado: Ana López

===== MENÚ UNIVERSIDAD =====
1. Agregar profesor
2. Agregar curso
3. Asignar profesor a curso
4. Listar profesores
5. Listar cursos
6. Buscar profesor por ID
7. Buscar curso por código
8. Eliminar curso
9. Eliminar profesor
10. Salir
Seleccione una opción: 7
Ingrese código del curso a buscar: C104
Código: C104
Nombre: Estructuras de Datos
Profesor: (sin asignar)
-----
```

REPOSITORIO REMOTO: <https://github.com/GomezJEmilia/UTN-Programacion2-TPs-EmiliaGJ/tree/a09c6fecf7e72a7ba17ed8c9399c371fdc7f89e6/05%20UML>