

TRABAJO PRÁCTICO N°6

Colecciones

Alumno: Ignacio Figueroa – 45.406.120

Tecnicatura Universitaria en Programación – UTN

Materia: Programación 2

Comisión: 7

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.

Ejercicio 1 – Sistema de Stock

Enum: CategoriaProducto

```
1 package Ejercicio1;
2
3 public enum CategoriaProducto {
4     ALIMENTOS("Productos comestibles"),
5     ELECTRONICA("Dispositivos electronicos"),
6     ROPA("Prendas de vestir"),
7     HOGAR("Articulos para el hogar");
8
9     private final String descripcion;
10
11    CategoriaProducto(String descripcion) {
12        this.descripcion = descripcion;
13    }
14
15    public String getDescripcion() {
16        return descripcion;
17    }
18}
19|
```

Clase Producto:

```
1 package Ejercicio1;
2
3 public class Producto {
4
5     private String id;
6     private String nombre;
7     private double precio;
8     private int cantidad;
9     private CategoríaProducto categoria;
10
11    public Producto(String id, String nombre, double precio, int cantidad, CategoríaProducto categoria) {
12        this.id = id;
13        this.nombre = nombre;
14        this.precio = precio;
15        this.cantidad = cantidad;
16        this.categoría = categoría;
17    }
18
19    public String getId() {
20        return id;
21    }
22
23    public String getNombre() {
24        return nombre;
25    }
26
27    public double getPrecio() {
28        return precio;
29    }
30
31    public CategoríaProducto getCategoría() {
32        return categoría;
33    }
34
35    public int getCantidad() {
36        return cantidad;
37    }
38
39    public void setCantidad(int cantidad) {
40        this.cantidad = cantidad;
41    }
42
43    public void mostrarInfo() {
44        System.out.println(this.toString());
45    }
46
47    @Override
48    public String toString() {
49        return "Producto{" + "id=" + id + ", nombre=" + nombre + ", precio=" + precio + ", cantidad=" + cantidad + ", categoría=" + categoría + '}';
50    }
51
52}
```

Clase Inventario:

```

1 package Ejercicio1;
2
3 import java.util.ArrayList;
4 import java.util.Comparator;
5 import java.util.List;
6 import java.util.stream.Collectors;
7
8 public class Inventario {
9
10    private ArrayList<Producto> productos;
11
12    public Inventario() {
13        this.productos = new ArrayList<>();
14    }
15
16    public void agregarProducto(Producto p) {
17        productos.add(p);
18    }
19
20    public void listarProductos() {
21        productos.forEach(Producto::mostrarInfo);
22    }
23
24    public Producto buscarProductoPorId(String id) {
25        return productos.stream()
26            .filter(p -> p.getId().equalsIgnoreCase(id))
27            .findFirst()
28            .orElse(null);
29    }
30
31    public boolean eliminarProducto(String id) {
32        Producto p = buscarProductoPorId(id);
33        if (p == null) {
34            return productos.remove(p);
35        }
36        return false;
37    }
38
39    public boolean actualizarStock(String id, int nuevaCantidad) {
40        Producto p = buscarProductoPorId(id);
41        if (p == null) {
42            p.setCantidad(nuevaCantidad);
43            return true;
44        }
45        return false;
46    }
47
48    public List<Producto> filtrarPorCategoria(CategoríaProducto categoría) {
49        return productos.stream()
50            .filter(p -> p.getCategoría() == categoría)
51            .collect(Collectors.toList());
52    }
53
54    public int obtenerTotalStock() {
55        return productos.stream().mapToInt(Producto::getCantidad).sum();
56    }
57
58    public Producto obtenerProductoConMayorStock() {
59        return productos.stream()
60            .max(Comparator.comparingInt(Producto::getCantidad))
61            .orElse(null);
62    }
63
64    public List<Producto> filtrarProductosPorPrecio(double min, double max) {
65        return productos.stream()
66            .filter(p -> p.getPrecio() >= min && p.getPrecio() <= max)
67            .collect(Collectors.toList());
68    }
69
70    public void mostrarCategoríasDisponibles() {
71        for (CategoríaProducto c : CategoríaProducto.values()) {
72            System.out.println(c + " - " + c.getDescripción());
73        }
74    }
75}

```

Main

```

import Ejercicio1.CategoríaProducto;
import Ejercicio1.Inventario;
import Ejercicio1.Producto;

/**
 *
 * @author Ignacio Figueroa - Comisión 7
 */
public class Tp6Colecciones {

    public static void main(String[] args) {
        Inventario inv = new Inventario();

        Producto p1 = new Producto("P001", "Arroz 1kg", 1500, 20, CategoríaProducto.ALIMENTOS);
        Producto p2 = new Producto("P002", "Auriculares", 2500, 10, CategoríaProducto.ELECTRÓNICA);
        Producto p3 = new Producto("P003", "Remera", 1200, 30, CategoríaProducto.ROPA);
        Producto p4 = new Producto("P004", "Plancha", 4500, 5, CategoríaProducto.HOGAR);
        Producto p5 = new Producto("P005", "Galletitas", 900, 50, CategoríaProducto.ALIMENTOS);

        inv.agregarProducto(p1);
        inv.agregarProducto(p2);
        inv.agregarProducto(p3);
        inv.agregarProducto(p4);
        inv.agregarProducto(p5);

        System.out.println("--- Listar productos ---");
        inv.listarProductos();

        System.out.println("\n--- Buscar P002 ---");
        Producto buscado = inv.buscarProductoPorId("P002");
        if (buscado != null) {
            buscado.mostrarInfo();
        }

        System.out.println("\n--- Filtrar ALIMENTOS ---");
        inv.filtrarPorCategoría(CategoríaProducto.ALIMENTOS).forEach(Producto::mostrarInfo);

        System.out.println("\n--- Eliminar P004 ---");
        inv.eliminarProducto("P004");
        inv.listarProductos();

        System.out.println("\n--- Actualizar stock P003 a 40 ---");
        inv.actualizarStock("P003", 40);
        inv.buscarProductoPorId("P003").mostrarInfo();

        System.out.println("\n--- Total stock ---");
        System.out.println(inv.obtenerTotalStock());

        System.out.println("\n--- Producto con más stock ---");
        Producto mayor = inv.obtenerProductoConMayorStock();
        if (mayor != null) {
            mayor.mostrarInfo();
        }

        System.out.println("\n--- Productos entre $1000 y $3000 ---");
        inv.filtrarProductosPorPrecio(1000, 3000).forEach(Producto::mostrarInfo);

        System.out.println("\n--- Categorías disponibles ---");
        inv.mostrarCategoríasDisponibles();
    }
}

```

Resultado:

```

run:
--- Listar productos ---
Producto{id=P001, nombre=Arroz 1kg, precio=1500.0, cantidad=20, categoria=ALIMENTOS}
Producto{id=P002, nombre=Auriculares, precio=2500.0, cantidad=10, categoria=ELECTRONICA}
Producto{id=P003, nombre=Remera, precio=1200.0, cantidad=30, categoria=ROPA}
Producto{id=P004, nombre=Plancha, precio=4500.0, cantidad=5, categoria=HOGAR}
Producto{id=P005, nombre=Galletitas, precio=900.0, cantidad=50, categoria=ALIMENTOS}

--- Buscar P002 ---
Producto{id=P002, nombre=Auriculares, precio=2500.0, cantidad=10, categoria=ELECTRONICA}

--- Filtrar ALIMENTOS ---
Producto{id=P001, nombre=Arroz 1kg, precio=1500.0, cantidad=20, categoria=ALIMENTOS}
Producto{id=P005, nombre=Galletitas, precio=900.0, cantidad=50, categoria=ALIMENTOS}

--- Eliminar P004 ---
Producto{id=P001, nombre=Arroz 1kg, precio=1500.0, cantidad=20, categoria=ALIMENTOS}
Producto{id=P002, nombre=Auriculares, precio=2500.0, cantidad=10, categoria=ELECTRONICA}
Producto{id=P003, nombre=Remera, precio=1200.0, cantidad=30, categoria=ROPA}
Producto{id=P005, nombre=Galletitas, precio=900.0, cantidad=50, categoria=ALIMENTOS}

--- Actualizar stock P003 a 40 ---
Producto{id=P003, nombre=Remera, precio=1200.0, cantidad=40, categoria=ROPA}

--- Total stock ---
120

--- Producto con más stock ---
Producto{id=P005, nombre=Galletitas, precio=900.0, cantidad=50, categoria=ALIMENTOS}

--- Productos entre $1000 y $3000 ---
Producto{id=P001, nombre=Arroz 1kg, precio=1500.0, cantidad=20, categoria=ALIMENTOS}
Producto{id=P002, nombre=Auriculares, precio=2500.0, cantidad=10, categoria=ELECTRONICA}
Producto{id=P003, nombre=Remera, precio=1200.0, cantidad=40, 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

Clase Autor:

```
package Ejercicio2;

public class Autor {

    private String id;
    private String nombre;
    private String nacionalidad;

    public Autor(String id, String nombre, String nacionalidad) {
        this.id = id;
        this.nombre = nombre;
        this.nacionalidad = nacionalidad;
    }

    public String getId() {
        return id;
    }

    public String getNombre() {
        return nombre;
    }

    public String getNacionalidad() {
        return nacionalidad;
    }

    public void mostrarInfo() {
        System.out.println(this.toString());
    }

    @Override
    public String toString() {
        return "Autor{" + "id=" + id + ", nombre=" + nombre + ", nacionalidad=" + nacionalidad + '}';
    }
}
```

Clase Libro:

```
package Ejercicio2;

public class Libro {

    private String isbn;
    private String titulo;
    private int anioPublicacion;
    private Autor autor;

    public Libro(String isbn, String titulo, int anioPublicacion, Autor autor) {
        this.isbn = isbn;
        this.titulo = titulo;
        this.anioPublicacion = anioPublicacion;
        this.autor = autor;
    }

    public String getIsbn() {
        return isbn;
    }

    public String getTitulo() {
        return titulo;
    }

    public int getAnioPublicacion() {
        return anioPublicacion;
    }

    public Autor getAutor() {
        return autor;
    }

    public void mostrarInfo() {
        System.out.println(this.toString());
    }

    @Override
    public String toString() {
        return "Libro{" + "isbn=" + isbn + ", titulo=" + titulo + ", anioPublicacion=" + anioPublicacion + ", autor=" + autor + '}';
    }
}
```

Clase Biblioteca:

```
package Ejercicio2;

import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public class Biblioteca {

    private String nombre;
    private List<Libro> libros;

    public Biblioteca(String nombre) {
        this.nombre = nombre;
        this.libros = new ArrayList<>();
    }

    public void agregarLibro(String isbn, String titulo, int anio, Autor autor) {
        libros.add(new Libro(isbn, titulo, anio, autor));
    }

    public void listarLibros() {
        libros.forEach(Libro::mostrarInfo);
    }

    public Libro buscarLibroPorIsbn(String isbn) {
        return libros.stream()
            .filter(l → l.getIsbn().equalsIgnoreCase(isbn))
            .findFirst()
            .orElse(null);
    }

    public boolean eliminarLibro(String isbn) {
        Libro l = buscarLibroPorIsbn(isbn);
        if (l != null) {
            return libros.remove(l);
        }
        return false;
    }

    public int obtenerCantidadLibros() {
        return libros.size();
    }

    public List<Libro> filtrarLibrosPorAnio(int anio) {
        return libros.stream()
            .filter(l → l.getAnioPublicacion() == anio)
            .collect(Collectors.toList());
    }

    public void mostrarAutoresDisponibles() {
        libros.stream()
            .map(Libro::getAutor)
            .distinct()
            .forEach(Autor::mostrarInfo);
    }
}
```

Demo:

```

public class DemoBiblioteca {

    public static void ejecutar() {
        Biblioteca b = 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", "Haruki Murakami", "Japonesa");

        b.agregarLibro("ISBN001", "Cien años de soledad", 1967, a1);
        b.agregarLibro("ISBN002", "Harry Potter 1", 1997, a2);
        b.agregarLibro("ISBN003", "Kafka en la orilla", 2002, a3);
        b.agregarLibro("ISBN004", "El amor en los tiempos del cólera", 1985, a1);

        System.out.println("--- Listado de libros ---");
        b.listarLibros();

        System.out.println("\n--- Buscar ISBN003 ---");
        Libro l = b.buscarLibroPorIsbn("ISBN003");
        if (l != null) {
            l.mostrarInfo();
        }

        System.out.println("\n--- Filtrar por año 1997 ---");
        b.filtrarLibrosPorAnio(1997).forEach(Libro::mostrarInfo);

        System.out.println("\n--- Eliminar ISBN002 ---");
        b.eliminarLibro("ISBN002");
        b.listarLibros();

        System.out.println("\n--- Total de libros ---");
        System.out.println(b.obtenerCantidadLibros());

        System.out.println("\n--- Autores disponibles ---");
        b.mostrarAutoresDisponibles();
    }
}

```

Resultados:

```

--- Listado de libros ---
Libro(isbn=ISBN001, titulo=Cien años de soledad, anioPublicacion=1967, autor=Autor(id=A001, nombre=Gabriel García Márquez, nacionalidad=Colombiana))
Libro(isbn=ISBN002, titulo=Harry Potter 1, anioPublicacion=1997, autor=Autor(id=A002, nombre=J.K. Rowling, nacionalidad=Británica))
Libro(isbn=ISBN003, titulo=Kafka en la orilla, anioPublicacion=2002, autor=Autor(id=A003, nombre=Haruki Murakami, nacionalidad=Japonesa))
Libro(isbn=ISBN004, titulo=El amor en los tiempos del cólera, anioPublicacion=1985, autor=Autor(id=A001, nombre=Gabriel García Márquez, nacionalidad=Colombiana))

--- Buscar ISBN003 ---
Libro(isbn=ISBN003, titulo=Kafka en la orilla, anioPublicacion=2002, autor=Autor(id=A003, nombre=Haruki Murakami, nacionalidad=Japonesa))

--- Filtrar por año 1997 ---
Libro(isbn=ISBN002, titulo=Harry Potter 1, anioPublicacion=1997, autor=Autor(id=A002, nombre=J.K. Rowling, nacionalidad=Británica))

--- Eliminar ISBN002 ---
Libro(isbn=ISBN001, titulo=Cien años de soledad, anioPublicacion=1967, autor=Autor(id=A001, nombre=Gabriel García Márquez, nacionalidad=Colombiana))
Libro(isbn=ISBN003, titulo=Kafka en la orilla, anioPublicacion=2002, autor=Autor(id=A003, nombre=Haruki Murakami, nacionalidad=Japonesa))
Libro(isbn=ISBN004, titulo=El amor en los tiempos del cólera, anioPublicacion=1985, autor=Autor(id=A001, nombre=Gabriel García Márquez, nacionalidad=Colombiana))

--- Total de libros ---
3

--- Autores disponibles ---
Autor(id=A001, nombre=Gabriel García Márquez, nacionalidad=Colombiana)
Autor(id=A003, nombre=Haruki Murakami, nacionalidad=Japonesa)

BUILD SUCCESSFUL (total time: 0 seconds)

```

Ejercicio 3 - Universidad (Relacion Bidireccional)

Clase Profesor:

```
 1 import java.util.List;
 2
 3 public class Profesor {
 4
 5     private String id;
 6     private String nombre;
 7     private String especialidad;
 8     private List<Curso> cursos;
 9
10
11     public Profesor(String id, String nombre, String especialidad) {
12         this.id = id;
13         this.nombre = nombre;
14         this.especialidad = especialidad;
15         this.cursos = new ArrayList<>();
16     }
17
18
19     public String getId() {
20         return id;
21     }
22
23     public String getNombre() {
24         return nombre;
25     }
26
27     public List<Curso> getCursos() {
28         return cursos;
29     }
30
31
32     public void agregarCurso(Curso c) {
33         if (!cursos.contains(c)) {
34             cursos.add(c);
35             if (c.getProfesor() != this) {
36                 c.setProfesor(this);
37             }
38         }
39     }
40
41     public void eliminarCurso(Curso c) {
42         if (cursos.remove(c)) {
43             if (c.getProfesor() == this) {
44                 c.setProfesor(null);
45             }
46         }
47     }
48
49     public void mostrarInfo() {
50         System.out.println("Profesor: " + nombre + " | Especialidad: " + especialidad
51                             + " | Cursos: " + cursos.size());
52     }
53
54 }
```

Clase Curso:

```
package Ejercicio3;

public class Curso {

    private String codigo;
    private String nombre;
    private Profesor profesor;

    public Curso(String codigo, String nombre) {
        this.codigo = codigo;
        this.nombre = nombre;
    }

    public String getCodigo() {
        return codigo;
    }

    public String getNombre() {
        return nombre;
    }

    public Profesor getProfesor() {
        return profesor;
    }

    public void setProfesor(Profesor p) {
        if (this.profesor == p) {
            return;
        }

        if (this.profesor != null) {
            Profesor old = this.profesor;
            this.profesor = null;
            old.eliminarCurso(this);
        }

        this.profesor = p;

        if (p != null && !p.getCursoros().contains(this)) {
            p.getCursoros().add(this);
        }
    }

    public void mostrarInfo() {
        System.out.println("Curso: " + codigo + " - " + nombre
                           + " | Profesor: " + (profesor != null ? profesor.getNombre() : "Sin asignar"));
    }
}
```

Clase Universidad:

```

1  package Ejercicio3;
2
3  import java.util.List;
4  import java.util.ArrayList;
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     }
21
22     public void agregarCurso(Curso c) {
23         cursos.add(c);
24     }
25
26     public Profesor buscarProfesor(String id) {
27         return profesores.stream()
28             .filter(p → p.getId().equalsIgnoreCase(id))
29             .findFirst()
30             .orElse(null);
31     }
32
33     public Curso buscarCurso(String codigo) {
34         return cursos.stream()
35             .filter(c → c.getCodigo().equalsIgnoreCase(codigo))
36             .findFirst()
37             .orElse(null);
38     }
39
40     public void asignarProfesor(String codigoCurso, String idProfesor) {
41         Curso c = buscarCurso(codigoCurso);
42         Profesor p = buscarProfesor(idProfesor);
43         if (c != null) {
44             c.setProfesor(p);
45         }
46     }
47
48     public void listarCursos() {
49         cursos.forEach(Curso::mostrarInfo);
50     }
51
52     public void listarProfesores() {
53         profesores.forEach(Profesor::mostrarInfo);
54     }
55
56     public void eliminarCurso(String codigo) {
57         Curso c = buscarCurso(codigo);
58         if (c != null) {
59             if (c.getProfesor() != null) {
60                 c.getProfesor().eliminarCurso(c);
61             }
62             cursos.remove(c);
63         }
64     }
65
66     public void eliminarProfesor(String id) {
67         Profesor p = buscarProfesor(id);
68         if (p != null) {
69             for (Curso c : new ArrayList<>(p.getCursos())) {
70                 c.setProfesor(null);
71             }
72             profesores.remove(p);
73         }
74     }
75
76 }

```

Demo:

```
package Ejercicio3;

public class DemoUniversidad {

    public static void ejecutar() {
        Universidad u = new Universidad("Universidad Nacional");

        Profesor p1 = new Profesor("P001", "Ana Gómez", "Matemáticas");
        Profesor p2 = new Profesor("P002", "Luis Pérez", "Informática");
        Profesor p3 = new Profesor("P003", "Maria Ruiz", "Física");

        Curso c1 = new Curso("C101", "Cálculo I");
        Curso c2 = new Curso("C102", "Programación I");
        Curso c3 = new Curso("C103", "Física General");
        Curso c4 = new Curso("C104", "Algoritmos");

        u.agregarProfesor(p1);
        u.agregarProfesor(p2);
        u.agregarProfesor(p3);

        u.agregarCurso(c1);
        u.agregarCurso(c2);
        u.agregarCurso(c3);
        u.agregarCurso(c4);

        u.asignarProfesor("C101", "P001");
        u.asignarProfesor("C102", "P002");
        u.asignarProfesor("C103", "P003");
        u.asignarProfesor("C104", "P002");

        System.out.println("--- Cursos ---");
        u.listarCursos();

        System.out.println("\n--- Profesores ---");
        u.listarProfesores();

        System.out.println("\n--- Cambiar profesor de C104 a P003 ---");
        u.asignarProfesor("C104", "P003");
        u.listarCursos();

        System.out.println("\n--- Eliminar curso C103 ---");
        u.eliminarCurso("C103");
        u.listarCursos();

        System.out.println("\n--- Eliminar profesor P001 ---");
        u.eliminarProfesor("P001");
        u.listarProfesores();
        u.listarCursos();
    }
}
```

Resultados:

```
--- Cursos ---
Curso: C101 - Cálculo I | Profesor: Ana Gómez
Curso: C102 - Programación I | Profesor: Luis Pérez
Curso: C103 - Física General | Profesor: María Ruiz
Curso: C104 - Algoritmos | Profesor: Luis Pérez

--- Profesores ---
Profesor: Ana Gómez | Especialidad: Matemáticas | Cursos: 1
Profesor: Luis Pérez | Especialidad: Informática | Cursos: 2
Profesor: María Ruiz | Especialidad: Física | Cursos: 1

--- Cambiar profesor de C104 a P003 ---
Curso: C101 - Cálculo I | Profesor: Ana Gómez
Curso: C102 - Programación I | Profesor: Luis Pérez
Curso: C103 - Física General | Profesor: María Ruiz
Curso: C104 - Algoritmos | Profesor: María Ruiz

--- Eliminar curso C103 ---
Curso: C101 - Cálculo I | Profesor: Ana Gómez
Curso: C102 - Programación I | Profesor: Luis Pérez
Curso: C104 - Algoritmos | Profesor: María Ruiz

--- Eliminar profesor P001 ---
Profesor: Luis Pérez | Especialidad: Informática | Cursos: 1
Profesor: María Ruiz | Especialidad: Física | Cursos: 1
Curso: C101 - Cálculo I | Profesor: Sin asignar
Curso: C102 - Programación I | Profesor: Luis Pérez
Curso: C104 - Algoritmos | Profesor: María Ruiz
BUILD SUCCESSFUL (total time: 0 seconds)
```