

PROGRAMACION II

Trabajo Práctico 6: COLECCIONES Y SISTEMA DE STOCK

Estudiante: Fabián Alejandro Salas

Comisión: 17

Link de repositorio: <https://github.com/fabiansalas2025/PROGRAMACION-2-UNIDAD-6-COLECCIONES.git>

CASO PRACTICO N° 1 – SISTEMA DE STOCK

Enum CategoriaProducto

```
public enum CategoriaProducto {  
    ALIMENTOS("Productos comestibles"),  
    ELECTRONICA("Dispositivos electrónicos"),  
    ROPA("Prendas de vestir"),  
    HOGAR("Artículos para el hogar");  
  
    private final String descripcion;  
  
    CategoriaProducto(String descripcion) {  
        this.descripcion = descripcion;  
    }  
  
    public String getDescripcion() {  
        return descripcion;  
    }  
}
```

Clase Producto

```
public class Producto {  
    private String id;  
    private String nombre;  
    private double precio;  
    private int cantidad;  
    private CategoriaProducto categoria;  
  
    public Producto(String id, String nombre, double precio, int cantidad, CategoriaProducto categoria) {  
        this.id = id;  
        this.nombre = nombre;  
        this.precio = precio;  
    }  
}
```

```
        this.cantidad = cantidad;
        this.categoria = categoria;
    }

    public String getId() {
        return id;
    }

    public int getCantidad() {
        return cantidad;
    }

    public double getPrecio() {
        return precio;
    }

    public CategoriaProducto getCategoria() {
        return categoria;
    }

    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }

    public void mostrarInfo() {
        System.out.println("ID: " + id);
        System.out.println("Nombre: " + nombre);
        System.out.println("Precio: $" + precio);
        System.out.println("Cantidad: " + cantidad);
        System.out.println("Categoría: " + categoria + " - " + categoria.getDescripcion());
        System.out.println("-----");
    }
}
```

Clase Inventario

```
import java.util.ArrayList;
```

```
public class Inventario {
    private ArrayList<Producto> productos = new ArrayList<>();

    public void agregarProducto(Producto p) {
        productos.add(p);
    }

    public void listarProductos() {
        for (Producto p : productos) {
            p.mostrarInfo();
        }
    }
}
```

```
    }  
}  
  
public Producto buscarProductoPorId(String id) {  
    for (Producto p : productos) {  
        if (p.getId().equals(id)) {  
            return p;  
        }  
    }  
    return null;  
}  
  
public void eliminarProducto(String id) {  
    productos.removeIf(p -> p.getId().equals(id));  
}  
  
public void actualizarStock(String id, int nuevaCantidad) {  
    Producto p = buscarProductoPorId(id);  
    if (p != null) {  
        p.setCantidad(nuevaCantidad);  
    }  
}  
  
public void filtrarPorCategoria(CategoriaProducto categoria) {  
    for (Producto p : productos) {  
        if (p.getCategoria() == categoria) {  
            p.mostrarInfo();  
        }  
    }  
}  
  
public int obtenerTotalStock() {  
    int total = 0;  
    for (Producto p : productos) {  
        total += p.getCantidad();  
    }  
    return total;  
}  
  
public Producto obtenerProductoConMayorStock() {  
    if (productos.isEmpty()) return null;  
    Producto mayor = productos.get(0);  
    for (Producto p : productos) {  
        if (p.getCantidad() > mayor.getCantidad()) {  
            mayor = p;  
        }  
    }  
}
```

```
        return mayor;
    }

    public void filtrarProductosPorPrecio(double min, double max) {
        for (Producto p : productos) {
            if (p.getPrecio() >= min && p.getPrecio() <= max) {
                p.mostrarInfo();
            }
        }
    }

    public void mostrarCategoriasDisponibles() {
        for (CategoriaProducto c : CategoriaProducto.values()) {
            System.out.println(c + ": " + c.getDescripcion());
        }
    }
}
```

Clase Principal

```
public class Principal {

    public static void main(String[] args) {

        Inventario inventario = new Inventario();

        Producto p1 = new Producto("P001", "Arroz", 1200, 50,
        CategoriaProducto.ALIMENTOS);
        Producto p2 = new Producto("P002", "Celular", 25000, 20,
        CategoriaProducto.ELECTRONICA);
        Producto p3 = new Producto("P003", "Remera", 1500, 30, CategoriaProducto.ROPA);
        Producto p4 = new Producto("P004", "Silla", 3000, 15, CategoriaProducto.HOGAR);
        Producto p5 = new Producto("P005", "Leche", 800, 40,
        CategoriaProducto.ALIMENTOS);

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

        System.out.println("Listado de productos:");
        inventario.listarProductos();

        System.out.println("Buscar producto por ID 'P003':");
        Producto buscado = inventario.buscarProductoPorId("P003");
        if (buscado != null) buscado.mostrarInfo();
    }
}
```

```
System.out.println("Filtrar por categoría ALIMENTOS:");
inventario.filtrarPorCategoria(CategoriaProducto.ALIMENTOS);

System.out.println("Eliminar producto con ID 'P002':");
inventario.eliminarProducto("P002");
inventario.listarProductos();

System.out.println("Actualizar stock de 'P001' a 60:");
inventario.actualizarStock("P001", 60);
inventario.buscarProductoPorId("P001").mostrarInfo();

System.out.println("Total de stock disponible:");
System.out.println(inventario.obtenerTotalStock());

System.out.println("Producto con mayor stock:");
inventario.obtenerProductoConMayorStock().mostrarInfo();

System.out.println("Filtrar productos con precio entre $1000 y $3000:");
inventario.filtrarProductosPorPrecio(1000, 3000);

System.out.println("Categorías disponibles:");
inventario.mostrarCategoriasDisponibles();
}
}
```

CASO PRACTICO N° 2 – BIBLIOTECA Y LIBROS

CLASE BIBLIOTECA

```
import java.util.ArrayList;
public class Biblioteca {
    private String nombre;
    private ArrayList<Libro> libros = new ArrayList<>();

    public Biblioteca(String nombre) {
        this.nombre = nombre;
    }

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

    public void listarLibros() {
        for (Libro libro : libros) {
            libro.mostrarInfo();
        }
    }
}
```

```
public Libro buscarLibroPorIsbn(String isbn) {
    for (Libro libro : libros) {
        if (libro.getIsbn().equals(isbn)) {
            return libro;
        }
    }
    return null;
}

public void eliminarLibro(String isbn) {
    libros.removeIf(libro -> libro.getIsbn().equals(isbn));
}

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

public void filtrarLibrosPorAnio(int anio) {
    for (Libro libro : libros) {
        if (libro.getAnioPublicacion() == anio) {
            libro.mostrarInfo();
        }
    }
}

public void mostrarAutoresDisponibles() {
    ArrayList<String> autoresMostrados = new ArrayList<>();
    for (Libro libro : libros) {
        String idAutor = libro.getAutor().getId();
        if (!autoresMostrados.contains(idAutor)) {
            libro.getAutor().mostrarInfo();
            autoresMostrados.add(idAutor);
        }
    }
}
```

CLASE LIBRO

```
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 int getAnioPublicacion() {
    return anioPublicacion;
}

public Autor getAutor() {
    return autor;
}

public void mostrarInfo() {
    System.out.println("Título: " + titulo);
    System.out.println("ISBN: " + isbn);
    System.out.println("Año de publicación: " + anioPublicacion);
    autor.mostrarInfo();
}
}
```

CLASE AUTOR

```
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 void mostrarInfo() {
        System.out.println("Autor: " + nombre);
        System.out.println("ID: " + id);
        System.out.println("Nacionalidad: " + nacionalidad);
    }
}
```

```
        System.out.println("-----");
    }
}
```

CLASE PRINCIPAL

```
public class Principal {
    public static void main(String[] args) {

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

        Autor a1 = new Autor("A001", "Gabriel García Márquez", "Colombiana");
        Autor a2 = new Autor("A002", "Jorge Luis Borges", "Argentina");
        Autor a3 = new Autor("A003", "Isabel Allende", "Chilena");

        biblioteca.agregarLibro("L001", "Cien años de soledad", 1967, a1);
        biblioteca.agregarLibro("L002", "El Aleph", 1949, a2);
        biblioteca.agregarLibro("L003", "La casa de los espíritus", 1982, a3);
        biblioteca.agregarLibro("L004", "Crónica de una muerte anunciada", 1981, a1);
        biblioteca.agregarLibro("L005", "Ficciones", 1944, a2);

        System.out.println("Listado de libros:");
        biblioteca.listarLibros();

        System.out.println("Buscar libro por ISBN 'L003':");
        Libro buscado = biblioteca.buscarLibroPorIsbn("L003");
        if (buscado != null) buscado.mostrarInfo();

        System.out.println("Filtrar libros por año 1949:");
        biblioteca.filtrarLibrosPorAnio(1949);

        System.out.println("Eliminar libro 'L002':");
        biblioteca.eliminarLibro("L002");
        biblioteca.listarLibros();

        System.out.println("Cantidad total de libros:");
        System.out.println(biblioteca.obtenerCantidadLibros());

        System.out.println("Autores disponibles:");
        biblioteca.mostrarAutoresDisponibles();
    }
}
```

CASO PRACTICO N° 3 – UNIVERSIDAD-PROFESOR-CURSO

CLASE UNIVERSIDAD

```
import java.util.ArrayList;
import java.util.List;

public class Universidad {
    private String nombre;
    private List<Profesor> profesores;
    private List<Curso> cursos;

    public Universidad(String nombre) {
        this.nombre = nombre;
        this.profesores = new ArrayList<>();
        this.cursos = new ArrayList<>();
    }

    public void agregarProfesor(Profesor p) {
        profesores.add(p);
    }

    public void agregarCurso(Curso c) {
        cursos.add(c);
    }

    public void asignarProfesorACurso(String codigoCurso, String idProfesor) {
        Curso curso = buscarCursoPorCodigo(codigoCurso);
        Profesor profesor = buscarProfesorPorId(idProfesor);
        if (curso != null && profesor != null) {
            curso.setProfesor(profesor);
        }
    }

    public void listarProfesores() {
        for (Profesor p : profesores) {
            p.mostrarInfo();
        }
    }

    public void listarCursos() {
        for (Curso c : cursos) {
            c.mostrarInfo();
        }
    }

    public Profesor buscarProfesorPorId(String id) {
        for (Profesor p : profesores) {
            if (p.getId().equals(id)) return p;
        }
    }
}
```

```
        return null;
    }

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

    public void eliminarCurso(String codigo) {
        Curso curso = buscarCursoPorCodigo(codigo);
        if (curso != null) {
            if (curso.getProfesor() != null) {
                curso.getProfesor().eliminarCurso(curso);
            }
            cursos.remove(curso);
        }
    }

    public void eliminarProfesor(String id) {
        Profesor profesor = buscarProfesorPorId(id);
        if (profesor != null) {
            for (Curso c : new ArrayList<>(profesor.getCursos())) {
                c.setProfesor(null);
            }
            profesores.remove(profesor);
        }
    }

    public void reporteCantidadCursosPorProfesor() {
        for (Profesor p : profesores) {
            System.out.println("Profesor " + p.getId() + " dicta " + p.getCursos().size() + "
cursos.");
        }
    }
}
```

CLASE PROFESOR

```
import java.util.ArrayList;
import java.util.List;

public class Profesor {
    private String id;
    private String nombre;
    private String especialidad;
    private List<Curso> cursos;
```

```
public Profesor(String id, String nombre, String especialidad) {
    this.id = id;
    this.nombre = nombre;
    this.especialidad = especialidad;
    this.cursos = new ArrayList<>();
}

public void agregarCurso(Curso c) {
    if (!cursos.contains(c)) {
        cursos.add(c);
        if (c.getProfesor() != this) {
            c.setProfesor(this);
        }
    }
}

public void eliminarCurso(Curso c) {
    if (cursos.remove(c)) {
        c.setProfesor(null);
    }
}

public void listarCursos() {
    for (Curso c : cursos) {
        System.out.println("Curso: " + c.getCodigo() + " - " + c.getNombre());
    }
}

public void mostrarInfo() {
    System.out.println("Profesor: " + nombre + " | Especialidad: " + especialidad + " |
Cursos: " + cursos.size());
}

public String getId() {
    return id;
}

public List<Curso> getCursos() {
    return cursos;
}
}
```

CLASE CURSO

```
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 void setProfesor(Profesor p) {
        if (this.profesor == p) return; // Evita recursión infinita

        if (this.profesor != null) {
            this.profesor.getCursos().remove(this);
        }

        this.profesor = p;

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

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

    public String getCodigo() {
        return codigo;
    }

    public String getNombre() {
        return nombre;
    }

    public Profesor getProfesor() {
        return profesor;
    }
}

CLASE PRINCIPAL
public class Principal {
    public static void main(String[] args) {
        Universidad uni = new Universidad("UTN");
    }
}
```

```
Profesor p1 = new Profesor("P001", "Ana Gómez", "Matemática");  
Profesor p2 = new Profesor("P002", "Luis Pérez", "Programación");  
Profesor p3 = new Profesor("P003", "María López", "Física");
```

```
Curso c1 = new Curso("C101", "Álgebra");  
Curso c2 = new Curso("C102", "Programación I");  
Curso c3 = new Curso("C103", "Física I");  
Curso c4 = new Curso("C104", "Estadística");  
Curso c5 = new Curso("C105", "Programación II");
```

```
uni.agregarProfesor(p1);  
uni.agregarProfesor(p2);  
uni.agregarProfesor(p3);
```

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

```
uni.asignarProfesorACurso("C101", "P001");  
uni.asignarProfesorACurso("C102", "P002");  
uni.asignarProfesorACurso("C103", "P003");  
uni.asignarProfesorACurso("C104", "P001");  
uni.asignarProfesorACurso("C105", "P002");
```

```
System.out.println("--- Cursos ---");  
uni.listarCursos();
```

```
System.out.println("--- Profesores ---");  
uni.listarProfesores();
```

```
System.out.println("--- Cambio de profesor en curso C105 ---");  
uni.asignarProfesorACurso("C105", "P003");  
uni.listarCursos();  
uni.listarProfesores();
```

```
System.out.println("--- Eliminación del curso C104 ---");  
uni.eliminarCurso("C104");  
uni.listarCursos();  
uni.listarProfesores();
```

```
System.out.println("--- Eliminación del profesor P002 ---");  
uni.eliminarProfesor("P002");  
uni.listarCursos();  
uni.listarProfesores();
```

```
        System.out.println("--- Reporte de cursos por profesor ---");  
        uni.reporteCantidadCursosPorProfesor();  
    }  
}
```