

PROGRAMACIÓN II

Trabajo Práctico 6: Colecciones y Sistema de Stock

repositorio Github: <https://github.com/AguP10/UTN-TUPaD-P2>

Caso Práctico 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.

Clase Producto

```
package tp6;

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 String getNombre() {
        return nombre;
    }

    public double getPrecio() {
        return precio;
    }

    public int getCantidad() {
        return cantidad;
    }
}
```

```

public CategoriaProducto getCategoria() {
    return categoria;
}

public void setPrecio(double precio) {
    this.precio = precio;
}

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("-----");
}
}

```

Enum CategoriaProducto

```

package tp6;

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;
    }
}

```

Inventario

```
package tp6;

import java.util.ArrayList;

public class Inventario {

    private ArrayList<Producto> productos;

    public Inventario() {
        productos = new ArrayList<>();
    }

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

    public void listarProductos() {
        if (productos.isEmpty()) {
            System.out.println("No hay productos.");
        } else {
            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) {
        Producto p = buscarProductoPorId(id);
        if (p != null) {
            productos.remove(p);
        }
    }

    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());
        }
    }
}

```

Main

```
package tp6;

public class Main {

    public static void main(String[] args) {

        Inventario inventario = new Inventario();

        // 1. Crear al menos cinco productos con diferentes categorías y agregarlos al inventario
        Producto p1 = new Producto("P001", "Laptop", 2500.0, 5,
CategoríaProducto.ELECTRONICA);
        Producto p2 = new Producto("P002", "Remera", 1500.0, 10, CategoríaProducto.ROPA);
        Producto p3 = new Producto("P003", "Cereal", 1200.0, 20, CategoríaProducto.ALIMENTOS);
        Producto p4 = new Producto("P004", "Sofá", 5000.0, 2, CategoríaProducto.HOGAR);
        Producto p5 = new Producto("P005", "Auriculares", 3000.0, 8,
CategoríaProducto.ELECTRONICA);

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

        System.out.println("\n--- 2. Listar todos los productos ---");
        inventario.listarProductos();

        System.out.println("\n--- 3. Buscar un producto por ID (P003) ---");
        Producto buscado = inventario.buscarProductoPorId("P003");
        if (buscado != null) {
            buscado.mostrarInfo();
        } else {
            System.out.println("Producto no encontrado");
        }

        System.out.println("\n--- 4. Filtrar y mostrar productos de categoría ELECTRONICA ---");
        inventario.filtrarPorCategoria(CategoríaProducto.ELECTRONICA);

        System.out.println("\n--- 5. Eliminar producto por ID (P002) ---");
        inventario.eliminarProducto("P002");
        System.out.println("Productos restantes:");
        inventario.listarProductos();

        System.out.println("\n--- 6. Actualizar el stock del producto P001 a 12 unidades ---");
        inventario.actualizarStock("P001", 12);
        inventario.listarProductos();

        System.out.println("\n--- 7. Mostrar el total de stock disponible ---");
        System.out.println("Total stock: " + inventario.obtenerTotalStock());

        System.out.println("\n--- 8. Obtener y mostrar el producto con mayor stock ---");
```

```

    Producto mayorStock = inventario.obtenerProductoConMayorStock();
    if (mayorStock != null) {
        mayorStock.mostrarInfo();
    }

    System.out.println("\n--- 9. Filtrar productos con precios entre $1000 y $3000 ---");
    inventario.filtrarProductosPorPrecio(1000, 3000);

    System.out.println("\n--- 10. Mostrar categorías disponibles con sus descripciones ---");
    inventario.mostrarCategoriasDisponibles();
}
}

```

Consola

run:

```

--- 2. Listar todos los productos ---
ID: P001
Nombre: Laptop
Precio: $2500.0
Cantidad: 5
Categoría: ELECTRONICA - Dispositivos electrónicos
-----
ID: P002
Nombre: Remera
Precio: $1500.0
Cantidad: 10
Categoría: ROPA - Prendas de vestir
-----
ID: P003
Nombre: Cereal
Precio: $1200.0
Cantidad: 20
Categoría: ALIMENTOS - Productos comestibles
-----
ID: P004
Nombre: Sofá
Precio: $5000.0
Cantidad: 2
Categoría: HOGAR - Artículos para el hogar
-----
ID: P005
Nombre: Auriculares
Precio: $3000.0
Cantidad: 8
Categoría: ELECTRONICA - Dispositivos electrónicos
-----

--- 3. Buscar un producto por ID (P003) ---
ID: P003

```

Nombre: Cereal
Precio: \$1200.0
Cantidad: 20
Categoría: ALIMENTOS - Productos comestibles

--- 4. Filtrar y mostrar productos de categoría ELECTRONICA ---

ID: P001
Nombre: Laptop
Precio: \$2500.0
Cantidad: 5
Categoría: ELECTRONICA - Dispositivos electrónicos

ID: P005
Nombre: Auriculares
Precio: \$3000.0
Cantidad: 8
Categoría: ELECTRONICA - Dispositivos electrónicos

--- 5. Eliminar producto por ID (P002) ---

Productos restantes:
ID: P001
Nombre: Laptop
Precio: \$2500.0
Cantidad: 5
Categoría: ELECTRONICA - Dispositivos electrónicos

ID: P003
Nombre: Cereal
Precio: \$1200.0
Cantidad: 20
Categoría: ALIMENTOS - Productos comestibles

ID: P004
Nombre: Sofá
Precio: \$5000.0
Cantidad: 2
Categoría: HOGAR - Artículos para el hogar

ID: P005
Nombre: Auriculares
Precio: \$3000.0
Cantidad: 8
Categoría: ELECTRONICA - Dispositivos electrónicos

--- 6. Actualizar el stock del producto P001 a 12 unidades ---

ID: P001
Nombre: Laptop
Precio: \$2500.0
Cantidad: 12

Categoría: ELECTRONICA - Dispositivos electrónicos

ID: P003

Nombre: Cereal

Precio: \$1200.0

Cantidad: 20

Categoría: ALIMENTOS - Productos comestibles

ID: P004

Nombre: Sofá

Precio: \$5000.0

Cantidad: 2

Categoría: HOGAR - Artículos para el hogar

ID: P005

Nombre: Auriculares

Precio: \$3000.0

Cantidad: 8

Categoría: ELECTRONICA - Dispositivos electrónicos

--- 7. Mostrar el total de stock disponible ---

Total stock: 42

--- 8. Obtener y mostrar el producto con mayor stock ---

ID: P003

Nombre: Cereal

Precio: \$1200.0

Cantidad: 20

Categoría: ALIMENTOS - Productos comestibles

--- 9. Filtrar productos con precios entre \$1000 y \$3000 ---

ID: P001

Nombre: Laptop

Precio: \$2500.0

Cantidad: 12

Categoría: ELECTRONICA - Dispositivos electrónicos

ID: P003

Nombre: Cereal

Precio: \$1200.0

Cantidad: 20

Categoría: ALIMENTOS - Productos comestibles

ID: P005

Nombre: Auriculares

Precio: \$3000.0

Cantidad: 8

Categoría: ELECTRONICA - Dispositivos electrónicos

--- 10. Mostrar categorías disponibles con sus descripciones ---

ALIMENTOS - Productos comestibles

ELECTRONICA - Dispositivos electrónicos

ROPA - Prendas de vestir

HOGAR - Artículos para el hogar

BUILD SUCCESSFUL (total time: 0 seconds)

Caso Práctico 2

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.

Clase Autor

```
package tp6;

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("ID: " + id);
        System.out.println("Nombre: " + nombre);
        System.out.println("Nacionalidad: " + nacionalidad);
        System.out.println("-----");
    }
}
```

Clase Biblioteca

```
package tp6;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

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 anioPublicacion, Autor autor) {
        Libro libro = new Libro(isbn, titulo, anioPublicacion, autor);
        libros.add(libro);
        System.out.println("Libro agregado: " + titulo);
    }

    public void listarLibros() {
        if (libros.isEmpty()) {
            System.out.println("No hay libros en la biblioteca.");
        } else {
            for (Libro l : libros) {
                l.mostrarInfo();
            }
        }
    }

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

    public void eliminarLibro(String isbn) {
        Libro l = buscarLibroPorIsbn(isbn);
        if (l != null) {
            libros.remove(l);
            System.out.println("Libro eliminado: " + l.getTitulo());
        } else {
            System.out.println("Libro no encontrado con ISBN: " + isbn);
        }
    }
}
```

```

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

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

public void mostrarAutoresDisponibles() {
    Set<Autor> autores = new HashSet<>();
    for (Libro l : libros) {
        autores.add(l.getAutor());
    }
    for (Autor a : autores) {
        a.mostrarInfo();
    }
}
}

```

Clase Libro

```

package tp6;

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("Título: " + titulo);
        System.out.println("ISBN: " + isbn);
        System.out.println("Año: " + anioPublicacion);
        System.out.println("Autor: " + autor.getNombre() + " (" + autor.getNacionalidad() + ")");
        System.out.println("-----");
    }
}

```

Main

```

package tp6;

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("Título: " + titulo);
    }
}

```

```

        System.out.println("ISBN: " + isbn);
        System.out.println("Año: " + anioPublicacion);
        System.out.println("Autor: " + autor.getNombre() + " (" + autor.getNacionalidad() + ")");
        System.out.println("-----");
    }
}

```

Consola

run:

Libro agregado: Cien Años de Soledad

Libro agregado: Harry Potter y la Piedra Filosofal

Libro agregado: 1984

Libro agregado: Harry Potter y la Cámara Secreta

Libro agregado: Crónica de una muerte anunciada

--- 4. Listar todos los libros ---

Título: Cien Años de Soledad

ISBN: L001

Año: 1967

Autor: Gabriel García Márquez (Colombiano)

Título: Harry Potter y la Piedra Filosofal

ISBN: L002

Año: 1997

Autor: J.K. Rowling (Británica)

Título: 1984

ISBN: L003

Año: 1949

Autor: George Orwell (Británico)

Título: Harry Potter y la Cámara Secreta

ISBN: L004

Año: 1998

Autor: J.K. Rowling (Británica)

Título: Crónica de una muerte anunciada

ISBN: L005

Año: 1961

Autor: Gabriel García Márquez (Colombiano)

--- 5. Buscar un libro por ISBN (L003) ---

Título: 1984

ISBN: L003

Año: 1949

Autor: George Orwell (Británico)

--- 6. Filtrar libros publicados en 1997 ---

Título: Harry Potter y la Piedra Filosofal

ISBN: L002

Año: 1997
Autor: J.K. Rowling (Británica)

--- 7. Eliminar un libro por ISBN (L004) ---
Libro eliminado: Harry Potter y la Cámara Secreta
Libros restantes:
Título: Cien Años de Soledad
ISBN: L001
Año: 1967
Autor: Gabriel García Márquez (Colombiano)

Título: Harry Potter y la Piedra Filosofal
ISBN: L002
Año: 1997
Autor: J.K. Rowling (Británica)

Título: 1984
ISBN: L003
Año: 1949
Autor: George Orwell (Británico)

Título: Crónica de una muerte anunciada
ISBN: L005
Año: 1961
Autor: Gabriel García Márquez (Colombiano)

--- 8. Cantidad total de libros ---
Total libros: 4

--- 9. Listar todos los autores de la biblioteca ---
ID: A002
Nombre: J.K. Rowling
Nacionalidad: Británica

ID: A001
Nombre: Gabriel García Márquez
Nacionalidad: Colombiano

ID: A003
Nombre: George Orwell
Nacionalidad: Británico

BUILD SUCCESSFUL (total time: 0 seconds)

Caso Práctico 3

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).

Clase Profesor

```
package tp6;

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 String getId() {
        return id;
    }

    public String getNombre() {
        return nombre;
    }

    public String getEspecialidad() {
        return especialidad;
    }

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

    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.contains(c)) {
        cursos.remove(c);
        if (c.getProfesor() == this) {
            c.setProfesor(null);
        }
    }
}

public void listarCursos() {
    if (cursos.isEmpty()) {
        System.out.println(nombre + " no dicta cursos.");
    } else {
        System.out.println("Cursos de " + nombre + ":");
        for (Curso c : cursos) {
            System.out.println("- " + c.getCodigo() + " | " + c.getNombre());
        }
    }
}

public void mostrarInfo() {
    System.out.println("ID: " + id);
    System.out.println("Nombre: " + nombre);
    System.out.println("Especialidad: " + especialidad);
    System.out.println("Cantidad de cursos: " + cursos.size());
    System.out.println("-----");
}
}

```

Clase Curso

```

package tp6;

public class Curso {

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

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

    public String getCodigo() {
        return codigo;
    }

    public String getNombre() {

```



```

        return nombre;
    }

    public Profesor getProfesor() {
        return profesor;
    }

    public void setProfesor(Profesor p) {
        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("Código: " + codigo);
        System.out.println("Nombre: " + nombre);
        if (profesor != null) {
            System.out.println("Profesor: " + profesor.getNombre());
        } else {
            System.out.println("Profesor: Ninguno asignado");
        }
        System.out.println("-----");
    }
}

```

Clase Universidad

```

package tp6;

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 c = buscarCursoPorCodigo(codigoCurso);
    Profesor p = buscarProfesorPorId(idProfesor);
    if (c != null && p != null) {
        c.setProfesor(p);
        System.out.println("Profesor " + p.getNombre() + " asignado a curso " + c.getNombre());
    } else {
        System.out.println("Curso o profesor no encontrado.");
    }
}

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 c = buscarCursoPorCodigo(codigo);
    if (c != null) {
        if (c.getProfesor() != null) {
            c.getProfesor().getCursos().remove(c);
            c.setProfesor(null);
        }
    }
}

```

```

        cursos.remove(c);
        System.out.println("Curso eliminado: " + c.getNombre());
    }
}

public void eliminarProfesor(String id) {
    Profesor p = buscarProfesorPorId(id);
    if (p != null) {

        for (Curso c : new ArrayList<>(p.getCursos())) {
            c.setProfesor(null);
        }
        profesores.remove(p);
        System.out.println("Profesor eliminado: " + p.getNombre());
    }
}

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

```

Main

```

package tp6;

public class Main {

    public static void main(String[] args) {
        Universidad uni = new Universidad("Universidad Nacional");

        // 1. Crear 3 profesores
        Profesor prof1 = new Profesor("P001", "Ana Torres", "Matemáticas");
        Profesor prof2 = new Profesor("P002", "Luis Gómez", "Física");
        Profesor prof3 = new Profesor("P003", "María López", "Historia");

        // 2. Crear 5 cursos
        Curso c1 = new Curso("C001", "Álgebra");
        Curso c2 = new Curso("C002", "Física I");
        Curso c3 = new Curso("C003", "Historia Universal");
        Curso c4 = new Curso("C004", "Cálculo");
        Curso c5 = new Curso("C005", "Física II");

        // 3. Agregar profesores y cursos a la universidad
        uni.agregarProfesor(prof1);
        uni.agregarProfesor(prof2);
        uni.agregarProfesor(prof3);

        uni.agregarCurso(c1);
    }
}

```

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

```
// 4. Asignar profesores a cursos
```

```
uni.asignarProfesorACurso("C001", "P001"); // Álgebra -> Ana
uni.asignarProfesorACurso("C004", "P001"); // Cálculo -> Ana
uni.asignarProfesorACurso("C002", "P002"); // Física I -> Luis
uni.asignarProfesorACurso("C005", "P002"); // Física II -> Luis
uni.asignarProfesorACurso("C003", "P003"); // Historia -> María
```

```
// 5. Listar cursos con profesor y profesores con cursos
```

```
System.out.println("\n--- Cursos con sus profesores ---");
uni.listarCursos();
```

```
System.out.println("\n--- Profesores con sus cursos ---");
uni.listarProfesores();
```

```
// 6. Cambiar el profesor de un curso
```

```
System.out.println("\n--- Cambiar profesor de Física II a María ---");
uni.asignarProfesorACurso("C005", "P003"); // Física II ahora María
uni.listarCursos();
uni.listarProfesores();
```

```
// 7. Remover un curso
```

```
System.out.println("\n--- Eliminar curso C003 (Historia Universal) ---");
uni.eliminarCurso("C003");
uni.listarCursos();
uni.listarProfesores();
```

```
// 8. Remover un profesor
```

```
System.out.println("\n--- Eliminar profesor Luis Gómez ---");
uni.eliminarProfesor("P002");
uni.listarCursos();
uni.listarProfesores();
```

```
// 9. Reporte: cantidad de cursos por profesor
```

```
System.out.println("\n--- Reporte de cursos por profesor ---");
uni.reporteCursosPorProfesor();
```

```
}
```

```
}
```

Consola

run:

Profesor Ana Torres asignado a curso Álgebra

Profesor Ana Torres asignado a curso Cálculo

Profesor Luis Gómez asignado a curso Física I

Profesor Luis Gómez asignado a curso Física II

Profesor María López asignado a curso Historia Universal

--- Cursos con sus profesores ---

Código: C001

Nombre: Álgebra

Profesor: Ana Torres

Código: C002

Nombre: Física I

Profesor: Luis Gómez

Código: C003

Nombre: Historia Universal

Profesor: María López

Código: C004

Nombre: Cálculo

Profesor: Ana Torres

Código: C005

Nombre: Física II

Profesor: Luis Gómez

--- Profesores con sus cursos ---

ID: P001

Nombre: Ana Torres

Especialidad: Matemáticas

Cantidad de cursos: 2

ID: P002

Nombre: Luis Gómez

Especialidad: Física

Cantidad de cursos: 2

ID: P003

Nombre: María López

Especialidad: Historia

Cantidad de cursos: 1

--- Cambiar profesor de Física II a María ---

Profesor María López asignado a curso Física II

Código: C001
Nombre: Álgebra
Profesor: Ana Torres

Código: C002
Nombre: Física I
Profesor: Luis Gómez

Código: C003
Nombre: Historia Universal
Profesor: María López

Código: C004
Nombre: Cálculo
Profesor: Ana Torres

Código: C005
Nombre: Física II
Profesor: María López

ID: P001
Nombre: Ana Torres
Especialidad: Matemáticas
Cantidad de cursos: 2

ID: P002
Nombre: Luis Gómez
Especialidad: Física
Cantidad de cursos: 1

ID: P003
Nombre: María López
Especialidad: Historia
Cantidad de cursos: 2

--- Eliminar curso C003 (Historia Universal) ---

Curso eliminado: Historia Universal

Código: C001
Nombre: Álgebra
Profesor: Ana Torres

Código: C002
Nombre: Física I
Profesor: Luis Gómez

Código: C004
Nombre: Cálculo
Profesor: Ana Torres

Código: C005
Nombre: Física II

Profesor: María López

ID: P001

Nombre: Ana Torres

Especialidad: Matemáticas

Cantidad de cursos: 2

ID: P002

Nombre: Luis Gómez

Especialidad: Física

Cantidad de cursos: 1

ID: P003

Nombre: María López

Especialidad: Historia

Cantidad de cursos: 1

--- Eliminar profesor Luis Gómez ---

Profesor eliminado: Luis Gómez

Código: C001

Nombre: Álgebra

Profesor: Ana Torres

Código: C002

Nombre: Física I

Profesor: Ninguno asignado

Código: C004

Nombre: Cálculo

Profesor: Ana Torres

Código: C005

Nombre: Física II

Profesor: María López

ID: P001

Nombre: Ana Torres

Especialidad: Matemáticas

Cantidad de cursos: 2

ID: P003

Nombre: María López

Especialidad: Historia

Cantidad de cursos: 1

--- Reporte de cursos por profesor ---

Reporte de cursos por profesor:

Ana Torres: 2 curso(s)

María López: 1 curso(s)

BUILD SUCCESSFUL (total time: 0 seconds)