

PROGRAMACIÓN II

Trabajo Práctico 6: Colecciones y Sistema de Stock

Alumno: Gonzalo Barrios.

GitHub: <https://github.com/Goonza88/Programacion-II>

Caso Practico 1:

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 String getNombre() {
        return nombre;
    }

    public double getPrecio() {
        return precio;
    }

    public int getCantidad() {
        return cantidad;
    }

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

    public CategoriaProducto getCategoria() {
        return categoria;
    }

    public void mostrarInfo() {
        System.out.println(
            "\nNombre: " + nombre +
            "\nPrecio: " + precio +
            "\nCantidad: " + cantidad +
            "\nCategoría: " + categoria + ".");
    }
}
```

Clase Inventario:

```
package colecciones;
import java.util.ArrayList;

/**...4 lines */

public class Inventario {
    public final ArrayList <Producto> productos;

    public Inventario(ArrayList<Producto> productos) {
        this.productos = productos;
    }

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

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

    public void buscarProductoPorId(String id) {
        for (Producto p : productos) {
            if (p.getId().equals(id)) {
                p.mostrarInfo();
            }
        }
    }

    public void eliminarProducto(String id) {
        productos.removeIf(p -> p.getId().equals(id));
        System.out.println("Producto " + id + " eliminado.");
    }

    public void actualizarStock(String id, int nuevaCantidad) {
        for (Producto p : productos) {
            if (p.getId().equals(id)) {
                p.setCantidad(nuevaCantidad);
                System.out.println("Cantidad actualizada a " + nuevaCantidad + ".");
            }
        }
    }

    public void filtrarPorCategoria(CategoriaProducto categoria) {
        for (Producto p : productos) {
            if (p.getCategoria().equals(categoria)) {
                p.mostrarInfo();
            }
        }
    }
}
```

```
public void obtenerTotalStock() {  
    int total = 0;  
    for (Producto p : productos) {  
        total += p.getCantidad();  
    }  
    System.out.println("Stock total: " + total + ".");  
}  
  
public void obtenerProductoConMayorStock() {  
    if (productos.isEmpty()) {  
        System.out.println("No hay productos en el inventario.");  
        return;  
    }  
    Producto max = productos.get(0);  
    for (Producto p : productos) {  
        if (p.getCantidad() > max.getCantidad()) {  
            max = p;  
        }  
    }  
    max.mostrarInfo();  
}
```

```
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.name() + ": " + c.getDescripcion());  
    }  
}
```

Clase enum CategoriaProducto:

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

Main y Resultados:

```
import java.util.ArrayList;

/*...4 lines */

public class Colecciones {
    public static void main(String[] args) {
        ArrayList<Producto> lista = new ArrayList<>();
        Inventario inventario = new Inventario(lista);

        // Tarea 1:
        Producto Pan = new Producto("A1", "Pan Blanco", 1500.50, 30, CategoriaProducto.ALIMENTOS);
        Producto Celular = new Producto("B2", "Motorola", 1000000.00, 10, CategoriaProducto.ELECTRONICA);
        Producto Escoba = new Producto("C3", "Escoba", 2500.99, 20, CategoriaProducto.HOGAR);
        Producto Gorra = new Producto("D4", "Gorra", 5000.50, 25, CategoriaProducto.ROPA);
        Producto Tomate = new Producto("A2", "Tomate", 500.50, 50, CategoriaProducto.ALIMENTOS);

        inventario.agregarProducto(Pan);
        inventario.agregarProducto(Celular);
        inventario.agregarProducto(Escoba);
        inventario.agregarProducto(Gorra);
        inventario.agregarProducto(Tomate);
    }
}
```

```
//Tarea 2:
System.out.println("Lista de Productos:");
inventario.listarProductos();
```

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II

run:

Lista de Productos:

Nombre: Pan Blanco.
Precio: 1500.5.
Cantidad: 30.
Categoria: ALIMENTOS.

Nombre: Motorola.
Precio: 1000000.0.
Cantidad: 10.
Categoria: ELECTRONICA.

Nombre: Escoba.
Precio: 2500.99.
Cantidad: 20.
Categoria: HOGAR.

Nombre: Gorra.
Precio: 5000.5.
Cantidad: 25.
Categoria: ROPA.

Nombre: Tomate.
Precio: 500.5.
Cantidad: 50.
Categoria: ALIMENTOS.

```
//Tarea 3:
System.out.println("\nProducto buscado por ID:");
inventario.buscarProductoPorId("D4");
```

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II

Producto buscado por ID:

Nombre: Gorra.
Precio: 5000.5.
Cantidad: 25.
Categoria: ROPA.

```
//Tarea 4:
System.out.println("\nMostrar productos de cierta categoria:");
inventario.filtrarPorCategoria(CategoriaProducto.ALIMENTOS);
```

colecciones.Colecciones > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II x Debugger Console

Mostrar productos de cierta categoria:

Nombre: Pan Blanco.
Precio: 1500.5.
Cantidad: 30.
Categoria: ALIMENTOS.

Nombre: Tomate.
Precio: 500.5.
Cantidad: 50.
Categoria: ALIMENTOS.

```
//Tarea 5:
System.out.println("\nEliminar producto por ID y mostrar restantes:");
inventario.eliminarProducto("A1");
inventario.listarProductos();
```

colecciones.Colecciones > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II x Debugger Console x

Eliminar producto por ID y mostrar restantes:
Producto A1 eliminado.

Nombre: Motorola.
Precio: 1000000.0.
Cantidad: 10.
Categoria: ELECTRONICA.

Nombre: Escoba.
Precio: 2500.99.
Cantidad: 20.
Categoria: HOGAR.

Nombre: Gorra.
Precio: 5000.5.
Cantidad: 25.
Categoria: ROPA.

Nombre: Tomate.
Precio: 500.5.
Cantidad: 50.
Categoria: ALIMENTOS.

```
//Tarea 6:
System.out.println("\nActualizar stock:");
inventario.actualizarStock("B2", 15);
inventario.buscarProductoPorId("B2");
```

colecciones.Colecciones > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Reposito

Actualizar stock:

Cantidad actualizada a 15.

Nombre: Motorola.
Precio: 1000000.0.
Cantidad: 15.
Categoria: ELECTRONICA.

```
//Tarea 7:  
System.out.println("\nMostrar el total de stock disponible:");  
inventario.obtenerTotalStock();
```

coleccion.Colecciones > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II x

Mostrar el total de stock disponible:
Stock total: 110.

```
//Tarea 8:  
System.out.println("\nMostrar producto con mayor stock:");  
inventario.obtenerProductoConMayorStock();
```

coleccion.Colecciones > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II x

Mostrar producto con mayor stock:

Nombre: Tomate.
Precio: 500.5.
Cantidad: 50.
Categoria: ALIMENTOS.

```
//Tarea 9:  
System.out.println("\nFiltrar producto por precios:");  
inventario.filtrarProductosPorPrecio(1000, 3000);
```

coleccion.Colecciones > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II x

Filtrar producto por precios:

Nombre: Escoba.
Precio: 2500.99.
Cantidad: 20.
Categoria: HOGAR.

```
//Tarea 10:  
System.out.println("\nCategorias disponibles y sus descripciones:\n");  
inventario.mostrarCategoriasDisponibles();  
}
```

coleccion.Colecciones > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II x

Debugger Console x

Categorias disponibles y sus descripciones:

ALIMENTOS: Productos comestibles.
ELECTRONICA: Dispositivos electronicos.
ROPA: Prendas de vestir.
HOGAR: Articulos para el hogar.
BUILD SUCCESSFUL (total time: 0 seconds)

Caso Practico 2:

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 String getNombre() {  
        return nombre;  
    }  
  
    public String getNacionalidad() {  
        return nacionalidad;  
    }  
  
    public void mostrarInfo() {  
        System.out.println(  
            "\nNombre: " + nombre +  
            "\nNacionalidad: " + nacionalidad + ".");  
    }  
}
```

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 String getTitulo() {  
        return titulo;  
    }  
  
    public int getAnioPublicacion() {  
        return anioPublicacion;  
    }  
  
    public Autor getAutor() {  
        return autor;  
    }  
  
    public void mostrarInfo() {  
        System.out.println(  
            "\nTitulo: " + titulo +  
            "\nISBN: " + isbn +  
            "\nAnio de publicacion: " + anioPublicacion +  
            "\nAutor: " + autor.getNombre() + ".");  
    }  
}
```


Clase Biblioteca:

```
public class Biblioteca {
    private String nombre;
    private final ArrayList <Libro> libros;
    private final ArrayList <Autor> autores;

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

    public String getNombre() {
        return nombre;
    }

    public void agregarLibro(String isbn, String titulo, int anioPublicacion, Autor autor) {
        Libro nuevoLibro = new Libro(isbn, titulo, anioPublicacion, autor);
        libros.add(nuevoLibro);
        boolean autorExiste = false;
        for (Autor a : autores) {
            if (a.getId().equals(autor.getId())) {
                autorExiste = true;
                break;
            }
        }
        if (!autorExiste) {
            autores.add(autor);
        }
    }

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

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

    public void eliminarLibro(String isbn) {
        libros.removeIf(l -> l.getIsbn().equals(isbn));
        System.out.println("Libro " + isbn + " eliminado.");
    }

    public void obtenerCantidadLibros() {
        int total = 0;
        for (Libro l : libros) {
            total += 1;
        }
        System.out.println("Hay una cantidad de " + total + " libros.");
    }

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

    public void mostrarAutoresDisponibles() {
        for (Autor a : autores) {
            a.mostrarInfo();
        }
    }
}
```

Main y Resultados:

```
//Tarea 1:  
Biblioteca biblioteca = new Biblioteca("Gran Biblioteca Nacional");  
  
//Tarea 2:  
Autor a1 = new Autor("A1", "Gabriel Garcia Marquez", "Colombia");  
Autor a2 = new Autor("A2", "Stephen King", "Estados Unidos");  
Autor a3 = new Autor("A3", "Agatha Christie", "Gran Bretania");  
  
//Tarea 3:  
biblioteca.agregarLibro("0-1860-4848-3", "Cien años de soledad", 1967, a1);  
biblioteca.agregarLibro("0-4627-3631-8", "El amor en los tiempos del colera", 1985, a1);  
biblioteca.agregarLibro("0-3845-8829-8", "Misery", 1987, a2);  
biblioteca.agregarLibro("0-1331-5481-5", "El resplandor", 1977, a2);  
biblioteca.agregarLibro("0-4667-0845-9", "Asesinato en el Orient Express", 1934, a3);  
biblioteca.agregarLibro("0-5487-2683-3", "El misterio del tren azul", 1928, a3);
```

```
//Tarea 4:  
System.out.println("Lista de libros:");  
biblioteca.listarLibros();  
System.out.println("\nLista de autores:");  
biblioteca.mostrarAutoresDisponibles();
```

coleccion.Colecciones2 > main >

but X

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repository\Programacion II

```
run:  
Lista de libros:  
  
Titulo: Cien años de soledad.  
ISBN: 0-1860-4848-3.  
Año de publicacion: 1967.  
Autor: Gabriel Garcia Marquez.  
  
Titulo: El amor en los tiempos del colera.  
ISBN: 0-4627-3631-8.  
Año de publicacion: 1985.  
Autor: Gabriel Garcia Marquez.  
  
Titulo: Misery.  
ISBN: 0-3845-8829-8.  
Año de publicacion: 1987.  
Autor: Stephen King.  
  
Titulo: El resplandor.  
ISBN: 0-1331-5481-5.  
Año de publicacion: 1977.  
Autor: Stephen King.  
  
Titulo: Asesinato en el Orient Express.  
ISBN: 0-4667-0845-9.  
Año de publicacion: 1934.  
Autor: Agatha Christie.  
  
Titulo: El misterio del tren azul.  
ISBN: 0-5487-2683-3.  
Año de publicacion: 1928.  
Autor: Agatha Christie.
```

```
Lista de autores:  
  
Nombre: Gabriel Garcia Marquez.  
Nacionalidad: Colombia.  
  
Nombre: Stephen King.  
Nacionalidad: Estados Unidos.  
  
Nombre: Agatha Christie.  
Nacionalidad: Gran Bretania.
```

```
7 //Tarea 5:
8 System.out.println("\nBuscar libro por ISBN:");
9 biblioteca.buscarLibroPorIsbn("0-1331-5481-5");
```

coleccion.Colecciones2 > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion

Buscar libro por ISBN:

Titulo: El resplandor.
ISBN: 0-1331-5481-5.
Anio de publicacion: 1977.
Autor: Stephen King.

```
1 //Tarea 6:
2 System.out.println("\nFiltrar por anio:");
3 biblioteca.filtrarLibrosPorAnio(1985);
```

coleccion.Colecciones2 > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion

Filtrar por anio:

Titulo: El amor en los tiempos del colera.
ISBN: 0-4627-3631-8.
Anio de publicacion: 1985.
Autor: Gabriel Garcia Marquez.

```
5 //Tarea 7:
6 System.out.println("\nEliminar libro por ISBN:");
7 biblioteca.eliminarLibro("0-5487-2683-3");
8 biblioteca.listarLibros();
```

coleccion.Colecciones2 > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion

Eliminar libro por ISBN:

Libro 0-5487-2683-3 eliminado.

Titulo: Cien años de soledad.
ISBN: 0-1860-4848-3.
Anio de publicacion: 1967.
Autor: Gabriel Garcia Marquez.

Titulo: El amor en los tiempos del colera.
ISBN: 0-4627-3631-8.
Anio de publicacion: 1985.
Autor: Gabriel Garcia Marquez.

Titulo: Misery.
ISBN: 0-3845-8829-8.
Anio de publicacion: 1987.
Autor: Stephen King.

Titulo: El resplandor.
ISBN: 0-1331-5481-5.
Anio de publicacion: 1977.
Autor: Stephen King.

Titulo: Asesinato en el Orient Express.
ISBN: 0-4667-0845-9.
Anio de publicacion: 1934.
Autor: Agatha Christie.

```
0 //Tarea8:
1 System.out.println("\nMostrar cantidad de libros:");
2 biblioteca.obtenerCantidadLibros();
```

colecciones.Colecciones2 > main >

put x

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion

Mostrar cantidad de libros:
Hay una cantidad de 5 libros.

```
4 //Tarea 9:
5 System.out.println("\nTodos los autores:");
6 biblioteca.mostrarAutoresDisponibles();
7 }
```

colecciones.Colecciones2 > main >

put x

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Progr

Todos los autores:

Nombre: Gabriel Garcia Marquez.
Nacionalidad: Colombia.

Nombre: Stephen King.
Nacionalidad: Estados Unidos.

Nombre: Agatha Christie.
Nacionalidad: Gran Bretania.

Caso Practico 3:

Clase Profesor:

```
public class Profesor {  
    private String id;  
    private String nombre;  
    private String especialidad;  
    private final ArrayList <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);  
            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() {  
        for (Curso c : cursos) {  
            System.out.println(c.getNombre() + ".");  
        }  
    }  
  
    public void mostrarInfo() {  
        System.out.println(  
            "\n- Nombre: " + nombre +  
            "\n- ID: " + id +  
            "\n- Especialidad: " + especialidad +  
            "\n- Cursos: ";  
        this.listarCursos();  
    }  
}
```

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;
        this.profesor = null;
    }

    public String getCodigo() {
        return codigo;
    }

    public String getNombre() {
        return nombre;
    }

    public Profesor getProfesor() {
        return profesor;
    }

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

        this.profesor = nuevoProfesor;
        if (profesorAnterior != null) {
            profesorAnterior.eliminarCurso(this);
        }
        if (nuevoProfesor != null) {
            nuevoProfesor.agregarCurso(this);
        }
    }

    public void mostrarInfo() {
        String nombreProfesor = (this.profesor != null) ? this.profesor.getNombre() : "null";
        System.out.println(
            "\n- Código: " + codigo +
            "\n- Nombre: " + nombre +
            "\n- Profesor: " + nombreProfesor + ".");
    }
}
```

Clase Universidad:

```
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 String getNombre() {
        return nombre;
    }

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

    public void agregarCurso(Curso c) {
        if (!cursos.contains(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);
        } else {
            System.out.println("Error: 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) {  
    for (int i = 0; i < cursos.size(); i++) {  
        Curso curso = cursos.get(i);  
        if (curso.getCodigo().equals(codigo)) {  
            Profesor profesor = curso.getProfesor();  
            profesor.eliminarCurso(curso);  
            cursos.remove(curso);  
        }  
    }  
}
```

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

```
public void reporte() {  
    for (Profesor p : profesores) {  
        int suma = 0;  
        for (int i = 0; i < p.getCursos().size(); i++) {  
            suma += 1;  
        }  
        System.out.println("\n- Cantidad de cursos de " + p.getNombre() + ": " + suma + " ");  
        p.listarCursos();  
    }  
}
```


Main y Resultados:

```
public class Colecciones3 {  
    public static void main(String[] args) {  
        //Tarea 1:  
        Universidad U = new Universidad("UBA");  
  
        Profesor p1 = new Profesor("A1", "Jose Ramirez", "Artes");  
        Profesor p2 = new Profesor("B2", "Maria Fernandez", "Matematicas");  
        Profesor p3 = new Profesor("C3", "Nicolas Perez Gonzalez", "Geografia");  
  
        Curso c1 = new Curso("1A", "Matematicas");  
        Curso c2 = new Curso("2B", "Artes");  
        Curso c3 = new Curso("3C", "Geografia");  
        Curso c4 = new Curso("4D", "Programacion");  
        Curso c5 = new Curso("5E", "Finanzas");  
    }  
}
```

```
//Tarea 2:  
U.agregarProfesor(p1);  
U.agregarProfesor(p2);  
U.agregarProfesor(p3);  
  
U.agregarCurso(c1);  
U.agregarCurso(c2);  
U.agregarCurso(c3);  
U.agregarCurso(c4);  
U.agregarCurso(c5);
```

```
//Tarea 3:  
U.asignarProfesorACurso("1A", "B2");  
U.asignarProfesorACurso("2B", "A1");  
U.asignarProfesorACurso("3C", "C3");  
U.asignarProfesorACurso("4D", "B2");  
U.asignarProfesorACurso("5E", "C3");
```

```
//Tarea 4:  
System.out.println("Cursos con su profesor:");  
U.listarCursos();  
System.out.println("\nProfesores con sus cursos:");  
U.listarProfesores();
```

Cursos con su profesor:

- Código: 1A.
- Nombre: Matematicas.
- Profesor: Maria Fernandez.
- Código: 2B.
- Nombre: Artes.
- Profesor: Jose Ramirez.
- Código: 3C.
- Nombre: Geografia.
- Profesor: Nicolas Perez Gonzalez.
- Código: 4D.
- Nombre: Programacion.
- Profesor: Maria Fernandez.
- Código: 5E.
- Nombre: Finanzas.
- Profesor: Nicolas Perez Gonzalez.

Profesores con sus cursos:

- Nombre: Jose Ramirez.
- ID: A1.
- Especialidad: Artes.
- Cursos:
Artes.
- Nombre: Maria Fernandez.
- ID: B2.
- Especialidad: Matematicas.
- Cursos:
Matematicas.
Programacion.
- Nombre: Nicolas Perez Gonzalez.
- ID: C3.
- Especialidad: Geografia.
- Cursos:
Geografia.
Finanzas.

```
//Tarea 5:
System.out.println("\nCambiar un curso a otro profesor:");
U.asignarProfesorACurso("4D", "C3");
U.listarProfesores();
```

colecciones.Colecciones3 > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II x Debugger Console

Cambiar un curso a otro profesor:

- Nombre: Jose Ramirez.
- ID: A1.
- Especialidad: Artes.
- Cursos:
Artes.

- Nombre: Maria Fernandez.
- ID: B2.
- Especialidad: Matematicas.
- Cursos:
Matematicas.

- Nombre: Nicolas Perez Gonzalez.
- ID: C3.
- Especialidad: Geografia.
- Cursos:
Geografia.
Finanzas.
Programacion.

```
//Tarea 6:
System.out.println("\nEliminar curso:");
U.eliminarCurso("3C");
U.listarProfesores();
```

colecciones.Colecciones3 > main >

Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II x Debugger Console

Eliminar curso:

- Nombre: Jose Ramirez.
- ID: A1.
- Especialidad: Artes.
- Cursos:
Artes.

- Nombre: Maria Fernandez.
- ID: B2.
- Especialidad: Matematicas.
- Cursos:
Matematicas.

- Nombre: Nicolas Perez Gonzalez.
- ID: C3.
- Especialidad: Geografia.
- Cursos:
Finanzas.
Programacion.

```
1 //Tarea 7:
2 System.out.println("\nRemover profesor:");
3 U.eliminarProfesor("B2");
4 U.listarProfesores();
5 U.listarCursos();
```

colecciones.Colecciones3 > main >

Remove profesor:

- Nombre: Jose Ramirez.
- ID: A1.
- Especialidad: Artes.
- Cursos:
Artes.

- Nombre: Nicolas Perez Gonzalez.
- ID: C3.
- Especialidad: Geografia.
- Cursos:
Finanzas.
Programacion.

- Codigo: 1A.
- Nombre: Matematicas.
- Profesor: null.

- Codigo: 2B.
- Nombre: Artes.
- Profesor: Jose Ramirez.

- Codigo: 4D.
- Nombre: Programacion.
- Profesor: Nicolas Perez Gonzalez.

- Codigo: 5E.
- Nombre: Finanzas.
- Profesor: Nicolas Perez Gonzalez.

```
7 //Tarea 8:
8 System.out.println("\nMostrar cantidad de cursos por profesor:");
9 U.reporte();
10 }
```

colecciones.Colecciones3 > main >

Mostrar cantidad de cursos por profesor:

- Cantidad de cursos de Jose Ramirez: 1:
Artes.

- Cantidad de cursos de Nicolas Perez Gonzalez: 2:
Finanzas.
Programacion.