

TP 6 – Colecciones

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:

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.

2. Clases a implementar:

Clase Producto

Atributos:

- **id (String)** → Identificador único del producto.
- **nombre (String)** → Nombre del producto.
- **precio (double)** → Precio del producto.
- **cantidad (int)** → Cantidad en stock.
- **categoria (CategoriaProducto)** → Categoría del producto.

Métodos:

- **mostrarInfo()** → Muestra en consola la información del producto.

Enum CategoriaProducto

Valores:

- ALIMENTOS
- ELECTRONICA
- ROPA
- HOGAR

Método adicional:

java 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 Inventario

Atributos:

- ArrayList productos

Métodos requeridos:

- agregarProducto(Producto p)
- listarProductos()
- buscarProductoPorId(String id)
- eliminarProducto(String id)
- actualizarStock(String id, int nuevaCantidad)
- filtrarPorCategoria(CategoriaProducto categoria)
- obtenerTotalStock()
- obtenerProductoConMayorStock()
- filtrarProductosPorPrecio(double min, double max)
- mostrarCategoriasDisponibles()

3. Tareas a realizar

1. Crear al menos cinco productos con diferentes categorías y agregarlos al inventario.

2. Listar todos los productos mostrando su información y categoría.
3. Buscar un producto por ID y mostrar su información.
4. Filtrar y mostrar productos que pertenezcan a una categoría específica.
5. Eliminar un producto por su ID y listar los productos restantes.
6. Actualizar el stock de un producto existente.
7. Mostrar el total de stock disponible.
8. Obtener y mostrar el producto con mayor stock.
9. Filtrar productos con precios entre \$1000 y \$3000.
10. Mostrar las categorías disponibles con sus descripciones.

CONCLUSIONES ESPERADAS

- Comprender el uso de **this** para acceder a atributos de instancia.
- Aplicar constructores sobrecargados para flexibilizar la creación de objetos.
- Implementar métodos con el mismo nombre y distintos parámetros.
- Representar objetos con **toString()** para mejorar la depuración.
- Diferenciar y aplicar atributos y métodos estáticos en Java.
- Reforzar el diseño modular y reutilizable mediante el paradigma orientado a objetos.

CASO PRACTICO 2:

1. Descripción general

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.

2. Clases a implementar

Clase Autor

Atributos:

- **id (String)** → Identificador único del autor.
- **nombre (String)** → Nombre del autor.
- **nacionalidad (String)** → Nacionalidad del autor.

Métodos:

- **mostrarInfo()** → Muestra la información del autor en consola.

Clase Libro

Atributos:

- **isbn (String)** → Identificador único del libro.
- **titulo (String)** → Título del libro.
- **anioPublicacion (int)** → Año de publicación.
- **autor (Autor)** → Autor del libro.

Métodos:

- **mostrarInfo()** → Muestra título, ISBN, año y autor.

Clase Biblioteca

Atributo:

- **String nombre**
- **List libros** → Colección de libros de la biblioteca

Métodos requeridos:

- **agregarLibro(String isbn, String titulo,int anioPublicacion, Autor autor)**
- **listarLibros()**
- **buscarLibroPorIsbn(String isbn)**
- **eliminarLibro(String isbn)**
- **obtenerCantidadLibros()**
- **filtrarLibrosPorAnio(int anio)**
- **mostrarAutoresDisponibles()**

3. Tareas a realizar

1. Creamos una biblioteca.
2. Crear al menos tres autores.
3. Agregar 5 libros asociados a alguno de los Autores a la biblioteca.
4. Listar todos los libros con su información y la del autor.
5. Buscar un libro por su ISBN y mostrar su información.
6. Filtrar y mostrar los libros publicados en un año específico.
7. Eliminar un libro por su ISBN y listar los libros restantes.
8. Mostrar la cantidad total de libros en la biblioteca.
9. Listar todos los autores de los libros disponibles en la biblioteca.

Conclusiones esperadas

- Comprender la **composición 1 a N** entre Biblioteca y Libro.
- Reforzar el manejo de **colecciones dinámicas** (ArrayList).
- Practicar el uso de **métodos de búsqueda, filtrado y eliminación**.
- Mejorar la modularidad aplicando el paradigma de **programación orientada a objetos**.

Caso práctico 3:

1. Descripción general

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)

2. Clases a implementar

Clase Profesor

Atributos:

- **id (String)** → Identificador único.
- **nombre (String)** → Nombre completo.
- **especialidad (String)** → Área principal.
- **List cursos** → Cursos que dicta. Métodos sugeridos:
- **agregarCurso(Curso c)** → Agrega el curso a su lista si no está y sincroniza el lado del curso.
- **eliminarCurso(Curso c)** → Quita el curso y sincroniza el lado del curso (dejar profesor en null si corresponde).
- **listarCursos()** → Muestra códigos y nombres.
- **mostrarInfo()** → Imprime datos del profesor y cantidad de cursos.

Clase Curso

Atributos:

- **codigo (String)** → Código único.
- **nombre (String)** → Nombre del curso.
- **profesor (Profesor)** → Profesor responsable. Métodos sugeridos:
- **setProfesor(Profesor p)** → Asigna/cambia el profesor sincronizando ambos lados: o Si tenía profesor previo, quitarse de su lista.
- **mostrarInfo()** → Muestra código, nombre y nombre del profesor (si tiene).

Clase Universidad

Atributos:

- **String nombre**

- **List profesores**
- **List cursos**

Métodos requeridos:

- **agregarProfesor(Profesor p)**
- **agregarCurso(Curso c)**
- **asignarProfesorACurso(String codigoCurso, String idProfesor)** → Usa setProfesor del curso.
- **listarProfesores() / listarCursos()**
- **buscarProfesorPorId(String id)**
- **buscarCursoPorCodigo(String codigo)**
- **eliminarCurso(String codigo)** → Debe romper la relación con su profesor si la hubiera.
- **eliminarProfesor(String id)** → Antes de remover, dejar null los cursos que dictaba.

Tareas a realizar

1. Crear al menos 3 profesores y 5 cursos.
2. Agregar profesores y cursos a la universidad.
3. Asignar profesores a cursos usando asignarProfesorACurso(...).
4. Listar cursos con su profesor y profesores con sus cursos.
5. Cambiar el profesor de un curso y verificar que ambos lados quedan sincronizados.
6. Remover un curso y confirmar que ya no aparece en la lista del profesor.
7. Remover un profesor y dejar profesor = null,
8. Mostrar un reporte: cantidad de cursos por profesor.

Conclusiones esperadas

- Diferenciar bidireccionalidad de una relación unidireccional (navegación desde ambos extremos).
- Mantener invariantes de asociación (coherencia de referencias) al agregar, quitar o reasignar.
- Practicar colecciones (ArrayList), búsquedas y operaciones de alta/baja.
- Diseñar métodos “seguros” que sincronicen los dos lados siempre.

1) Caso práctico 1

```

TAREA 1: CREAR Y AGREGAR PRODUCTOS
Producto agregado: Arroz
Producto agregado: Laptop HP
Producto agregado: Camisa
Producto agregado: Lampara LED
Producto agregado: Mouse Logitech
TAREA 2: LISTAR TODOS LOS PRODUCTOS
LISTADO DE PRODUCTOS
ID: P001 | Nombre: Arroz | Precio: $1500.0 | Cantidad: 50 | Categoria: ALIMENTOS (Productos comestibles)
ID: P002 | Nombre: Laptop HP | Precio: $5500.0 | Cantidad: 10 | Categoria: ELECTRONICA (Dispositivos electronicos)
ID: P003 | Nombre: Camisa | Precio: $2500.0 | Cantidad: 30 | Categoria: ROPA (Prendas de vestir)
ID: P004 | Nombre: Lampara LED | Precio: $1800.0 | Cantidad: 25 | Categoria: HOGAR (Articulos para el hogar)
ID: P005 | Nombre: Mouse Logitech | Precio: $800.0 | Cantidad: 40 | Categoria: ELECTRONICA (Dispositivos electronicos)
TAREA 3: BUSCAR PRODUCTO POR ID
Producto encontrado:
ID: P003 | Nombre: Camisa | Precio: $2500.0 | Cantidad: 30 | Categoria: ROPA (Prendas de vestir)
TAREA 4: FILTRAR POR CATEGORIA
PRODUCTOS DE CATEGORIA: ELECTRONICA
ID: P002 | Nombre: Laptop HP | Precio: $5500.0 | Cantidad: 10 | Categoria: ELECTRONICA (Dispositivos electronicos)
ID: P005 | Nombre: Mouse Logitech | Precio: $800.0 | Cantidad: 40 | Categoria: ELECTRONICA (Dispositivos electronicos)
TAREA 5: ELIMINAR PRODUCTO
Producto eliminado: Mouse Logitech
LISTADO DE PRODUCTOS
ID: P001 | Nombre: Arroz | Precio: $1500.0 | Cantidad: 50 | Categoria: ALIMENTOS (Productos comestibles)
ID: P002 | Nombre: Laptop HP | Precio: $5500.0 | Cantidad: 10 | Categoria: ELECTRONICA (Dispositivos electronicos)
ID: P003 | Nombre: Camisa | Precio: $2500.0 | Cantidad: 30 | Categoria: ROPA (Prendas de vestir)
ID: P004 | Nombre: Lampara LED | Precio: $1800.0 | Cantidad: 25 | Categoria: HOGAR (Articulos para el hogar)
TAREA 6: ACTUALIZAR STOCK
Stock actualizado para: Arroz - Nueva cantidad: 100
TAREA 7: TOTAL DE STOCK
Total de stock disponible: 165 unidades
TAREA 8: PRODUCTO CON MAYOR STOCK
Producto con mayor stock:
ID: P001 | Nombre: Arroz | Precio: $1500.0 | Cantidad: 100 | Categoria: ALIMENTOS (Productos comestibles)
TAREA 9: FILTRAR POR PRECIO
PRODUCTOS CON PRECIO ENTRE $1000.0 Y $3000.0
ID: P001 | Nombre: Arroz | Precio: $1500.0 | Cantidad: 100 | Categoria: ALIMENTOS (Productos comestibles)
ID: P003 | Nombre: Camisa | Precio: $2500.0 | Cantidad: 30 | Categoria: ROPA (Prendas de vestir)
ID: P004 | Nombre: Lampara LED | Precio: $1800.0 | Cantidad: 25 | Categoria: HOGAR (Articulos para el hogar)
TAREA 10: CATEGORIAS DISPONIBLES
CATEGORIAS DISPONIBLES
ALIMENTOS: Productos comestibles
ELECTRONICA: Dispositivos electronicos
ROPA: Prendas de vestir
HOGAR: Articulos para el hogar
BUILD SUCCESSFUL (total time: 0 seconds)

```

2) Caso práctico 2

```
=== TAREA 1: CREAR BIBLIOTECA ===
Biblioteca creada: Biblioteca Central

=== TAREA 2: CREAR AUTORES ===
Autores creados:
ID: A001 | Nombre: Gabriel García Márquez | Nacionalidad: Colombiano
ID: A002 | Nombre: J.K. Rowling | Nacionalidad: Británica
ID: A003 | Nombre: Jorge Luis Borges | Nacionalidad: Argentino

=== TAREA 3: AGREGAR LIBROS ===
Libro agregado: Cien años de soledad
Libro agregado: El amor en los tiempos del cólera
Libro agregado: Harry Potter y la piedra filosofal
Libro agregado: Ficciones
Libro agregado: El Aleph

=== TAREA 4: LISTAR TODOS LOS LIBROS ===

=== LIBROS EN LA BIBLIOTECA Biblioteca Central ===
ISBN: ISBN001 | Título: Cien años de soledad | Año: 1967 | Autor: Gabriel García Márquez
ISBN: ISBN002 | Título: El amor en los tiempos del cólera | Año: 1985 | Autor: Gabriel García Márquez
ISBN: ISBN003 | Título: Harry Potter y la piedra filosofal | Año: 1997 | Autor: J.K. Rowling
ISBN: ISBN004 | Título: Ficciones | Año: 1944 | Autor: Jorge Luis Borges
ISBN: ISBN005 | Título: El Aleph | Año: 1949 | Autor: Jorge Luis Borges

=== TAREA 5: BUSCAR LIBRO POR ISBN ===
Libro encontrado:
ISBN: ISBN003 | Título: Harry Potter y la piedra filosofal | Año: 1997 | Autor: J.K. Rowling
Información del autor:
ID: A002 | Nombre: J.K. Rowling | Nacionalidad: Británica

=== TAREA 6: FILTRAR POR AÑO ===

=== LIBROS PUBLICADOS EN 1949 ===
ISBN: ISBN005 | Título: El Aleph | Año: 1949 | Autor: Jorge Luis Borges

=== TAREA 7: ELIMINAR LIBRO ===
Libro eliminado: El amor en los tiempos del cólera

=== LIBROS EN LA BIBLIOTECA Biblioteca Central ===
ISBN: ISBN001 | Título: Cien años de soledad | Año: 1967 | Autor: Gabriel García Márquez
ISBN: ISBN003 | Título: Harry Potter y la piedra filosofal | Año: 1997 | Autor: J.K. Rowling
ISBN: ISBN004 | Título: Ficciones | Año: 1944 | Autor: Jorge Luis Borges
ISBN: ISBN005 | Título: El Aleph | Año: 1949 | Autor: Jorge Luis Borges

=== TAREA 8: CANTIDAD TOTAL DE LIBROS ===
Total de libros en la biblioteca: 4

=== TAREA 9: LISTAR AUTORES DISPONIBLES ===

=== AUTORES DISPONIBLES EN LA BIBLIOTECA ===
ID: A001 | Nombre: Gabriel García Márquez | Nacionalidad: Colombiano
ID: A002 | Nombre: J.K. Rowling | Nacionalidad: Británica
ID: A003 | Nombre: Jorge Luis Borges | Nacionalidad: Argentino
BUILD SUCCESSFUL (total time: 0 seconds)
```


3) Caso práctico 3

```
Universidad Tecnológica
TAREA 1: CREAR PROFESORES Y CURSOS
Profesores y cursos creados correctamente.
TAREA 2: AGREGAR A LA UNIVERSIDAD
Profesor agregado: Dr. Juan Pérez
Profesor agregado: Dra. María González
Profesor agregado: Dr. Carlos López
Curso agregado: Programación I
Curso agregado: Programación II
Curso agregado: Bases de Datos
Curso agregado: Redes de Computadoras
Curso agregado: Estructuras de Datos
TAREA 3: ASIGNAR PROFESORES A CURSOS
Profesor Dr. Juan Pérez asignado al curso Programación I
Profesor Dr. Juan Pérez asignado al curso Programación II
Profesor Dr. Juan Pérez asignado al curso Estructuras de Datos
Profesor Dra. María González asignado al curso Bases de Datos
Profesor Dr. Carlos López asignado al curso Redes de Computadoras
TAREA 4: LISTADOS
CURSOS DE Universidad Tecnológica
Código: CS101 | Nombre: Programación I | Profesor: Dr. Juan Pérez
Código: CS102 | Nombre: Programación II | Profesor: Dr. Juan Pérez
Código: DB201 | Nombre: Bases de Datos | Profesor: Dra. María González
Código: NET301 | Nombre: Redes de Computadoras | Profesor: Dr. Carlos López
Código: CS201 | Nombre: Estructuras de Datos | Profesor: Dr. Juan Pérez
PROFESORES DE Universidad Tecnológica
ID: PR001 | Nombre: Dr. Juan Pérez | Especialidad: Programación | Cantidad de cursos: 3
ID: PR002 | Nombre: Dra. María González | Especialidad: Bases de Datos | Cantidad de cursos: 1
ID: PR003 | Nombre: Dr. Carlos López | Especialidad: Redes | Cantidad de cursos: 1
Detalle de cursos por profesor
Cursos dictados por Dr. Juan Pérez:
- CS101: Programación I
- CS102: Programación II
- CS201: Estructuras de Datos
Cursos dictados por Dra. María González:
- DB201: Bases de Datos
Cursos dictados por Dr. Carlos López:
- NET301: Redes de Computadoras
TAREA 5: CAMBIAR PROFESOR DE UN CURSO
Cambiando profesor de Programación II de Dr. Juan Pérez a Dra. María González...
Profesor Dra. María González asignado al curso Programación II
Verificación - Estado del curso CS102:
Código: CS102 | Nombre: Programación II | Profesor: Dra. María González
Verificación - Cursos de cada profesor:
Cursos dictados por Dr. Juan Pérez:
- CS101: Programación I
- CS201: Estructuras de Datos
Cursos dictados por Dra. María González:
- DB201: Bases de Datos
- CS102: Programación II
TAREA 6: REMOVE UN CURSO
Eliminando el curso 'Estructuras de Datos'...
Curso eliminado: Estructuras de Datos
Verificación - Cursos del Prof. Juan Pérez:
Cursos dictados por Dr. Juan Pérez:
- CS101: Programación I
CURSOS DE Universidad Tecnológica
Código: CS101 | Nombre: Programación I | Profesor: Dr. Juan Pérez
Código: CS102 | Nombre: Programación II | Profesor: Dra. María González
Código: DB201 | Nombre: Bases de Datos | Profesor: Dra. María González
Código: NET301 | Nombre: Redes de Computadoras | Profesor: Dr. Carlos López
TAREA 7: REMOVE UN PROFESOR
Eliminando al Dr. Carlos López...
Profesor eliminado: Dr. Carlos López
```

```
Verificación - Estado del curso NET301:
Código: NET301 | Nombre: Redes de Computadoras | Profesor: Sin asignar
PROFESORES DE Universidad Tecnológica
ID: PR001 | Nombre: Dr. Juan Pérez | Especialidad: Programación | Cantidad de cursos: 1
ID: PR002 | Nombre: Dra. María González | Especialidad: Bases de Datos | Cantidad de cursos: 2
TAREA 8: REPORTE DE CURSOS POR PROFESOR
REPORTE: CANTIDAD DE CURSOS POR PROFESOR
Dr. Juan Pérez (Programación): 1 curso(s)
Dra. María González (Bases de Datos): 2 curso(s)
FIN DEL PROGRAMA
Conclusiones demostradas:
Relación bidireccional correctamente sincronizada
Cambio de profesor actualiza ambos lados
Eliminación de curso se refleja en el profesor
Eliminación de profesor deja cursos sin asignar
BUILD SUCCESSFUL (total time: 0 seconds)
```

[Repositorio de github](#)