UNIVERSIDAD TECNOLOGICA NACIONAL

Trabajo Práctico 6

Colecciones

Programación II

Alumno

Alex Pedro Dauria

Fecha de Entrega

20 de Octubre de 2025

Índice

Link al Repositorio en GitHub

https://github.com/Alex-Dauria/UTN-TUPaD-P2/tree/main/06%20COLECCIONES/TPColeccionesAlexDauria

Sistema de Stock

Se implementó el sistema de stock utilizando las clases Producto, CategoriaProducto (enum) e Inventario, con un MainStock para demostrar las tareas.

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

Código

```
// 1. Crear 5 productos
Producto p1 = new Producto("P001", "Arroz", 120.0, 50, CategoriaProducto.ALIMENTOS);
Producto p2 = new Producto("P002", "Laptop", 2500.0, 10, CategoriaProducto.ELECTRONICA);
Producto p3 = new Producto("P003", "Remera", 800.0, 30, CategoriaProducto.ROPA);
Producto p4 = new Producto("P004", "Sartén", 1500.0, 15, CategoriaProducto.HOGAR);
Producto p5 = new Producto("P005", "Leche", 90.0, 100, CategoriaProducto.ALIMENTOS);

// Agregar al inventario
inventario.agregarProducto(p1);
inventario.agregarProducto(p2);
inventario.agregarProducto(p3);
inventario.agregarProducto(p4);
inventario.agregarProducto(p5);
```

2. Listar todos los productos mostrando su información y categoría.

Código

```
// 2. Listar productos
System.out.println("=== LISTA DE PRODUCTOS ===");
inventario.listarProductos();
```

Resultado

```
=== LISTA DE PRODUCTOS ===
ID: P001
Nombre: Arroz
Precio: $120.0
Cantidad: 50
Categor?a: ALIMENTOS - Productos comestibles
----------------------------
ID: P002
Nombre: Laptop
Precio: $2500.0
Cantidad: 10
Categor?a: ELECTRONICA - Dispositivos electr?nicos
----------------------------
ID: P003
Nombre: Remera
Precio: $800.0
Cantidad: 30
Categor?a: ROPA - Prendas de vestir
----------------------------
ID: P004
Nombre: Sart?n
Precio: $1500.0
Cantidad: 15
Categor?a: HOGAR - Art?culos para el hogar
----------------------------
ID: P005
Nombre: Leche
Precio: $90.0
Cantidad: 100
Categor?a: ALIMENTOS - Productos comestibles
----------------------------
```

3. Buscar un producto por ID y mostrar su información.

Código

```
// 3. Buscar por ID
System.out.println("=== BUSCAR POR ID P003 ===");
Producto encontrado = inventario.buscarProductoPorId("P003");
if (encontrado != null) encontrado.mostrarInfo();
```

Resultado

```
----------------------------
=== BUSCAR POR ID P003 ===
ID: P003
Nombre: Remera
Precio: $800.0
Cantidad: 30
Categor?a: ROPA - Prendas de vestir
----------------------------
```

4. Filtrar y mostrar productos que pertenezcan a una categoría específica.

Código

```
// 4. Filtrar por categoríap
System.out.println("=== FILTRAR ALIMENTOS ===");
inventario.filtrarPorCategoria(CategoriaProducto.ALIMENTOS);
```

Resultado

```
----------------------------
=== FILTRAR ALIMENTOS ===
ID: P001
Nombre: Arroz
Precio: $120.0
Cantidad: 50
Categor?a: ALIMENTOS - Productos comestibles
----------------------------
ID: P005
Nombre: Leche
Precio: $90.0
Cantidad: 100
Categor?a: ALIMENTOS - Productos comestibles
----------------------------
```

5. Eliminar un producto por su ID y listar los productos restantes.

Código

```
// 5. Eliminar producto
System.out.println("=== ELIMINAR P004 ===");
inventario.eliminarProducto("P004");
inventario.listarProductos();
```

Resultado

```
----------------------------
=== ELIMINAR P004 ===
ID: P001
Nombre: Arroz
Precio: $120.0
Cantidad: 50
Categor□a: ALIMENTOS - Productos comestibles
----------------------------
ID: P002
Nombre: Laptop
Precio: $2500.0
Cantidad: 10
Categor□a: ELECTRONICA - Dispositivos electr□nicos
----------------------------
ID: P003
Nombre: Remera
Precio: $800.0
Cantidad: 30
Categor□a: ROPA - Prendas de vestir
----------------------------
ID: P005
Nombre: Leche
Precio: $90.0
Cantidad: 100
Categor□a: ALIMENTOS - Productos comestibles
----------------------------
```

6. Actualizar el stock de un producto existente.

Código

```
// 6. Actualizar stock
System.out.println("=== ACTUALIZAR STOCK P001 a 200 ===");
inventario.actualizarStock("P001", 200);
inventario.listarProductos();
```

Resultado

```
----------------------------
=== ACTUALIZAR STOCK P001 a 200 ===
ID: P001
Nombre: Arroz
Precio: $120.0
Cantidad: 200
Categor□a: ALIMENTOS - Productos comestibles
----------------------------
ID: P002
Nombre: Laptop
Precio: $2500.0
Cantidad: 10
Categor□a: ELECTRONICA - Dispositivos electr□nicos
----------------------------
ID: P003
Nombre: Remera
Precio: $800.0
Cantidad: 30
Categor□a: ROPA - Prendas de vestir
----------------------------
ID: P005
Nombre: Leche
Precio: $90.0
Cantidad: 100
Categor□a: ALIMENTOS - Productos comestibles
----------------------------
```

7. Mostrar el total de stock disponible.

Código

```
// 7. Total stock
System.out.println("=== TOTAL STOCK: " + inventario.obtenerTotalStock());
```

Resultado

```
----------------------------
=== TOTAL STOCK: 340
```

8. Obtener y mostrar el producto con mayor stock.

Código

```
// 8. Producto con mayor stock
System.out.println("=== PRODUCTO CON MAYOR STOCK ===");
Producto mayor = inventario.obtenerProductoConMayorStock();
if (mayor != null) mayor.mostrarInfo();
```

Resultado

```
=== PRODUCTO CON MAYOR STOCK ===
ID: P001
Nombre: Arroz
Precio: $120.0
Cantidad: 200
Categor□a: ALIMENTOS - Productos comestibles
----------------------------
```

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

Código

```
// 9. Filtrar por precio
System.out.println("=== PRODUCTOS ENTRE $1000 Y $3000 ===");
inventario.filtrarProductosPorPrecio(1000, 3000);
```

Resultado

```
----------------------------
=== PRODUCTOS ENTRE $1000 Y $3000 ===
ID: P002
Nombre: Laptop
Precio: $2500.0
Cantidad: 10
Categor□a: ELECTRONICA - Dispositivos electr□nicos
----------------------------
```

10. Mostrar las categorías disponibles con sus descripciones.

Código

```
// 10. Mostrar categorías
System.out.println("=== CATEGORÍAS DISPONIBLES ===");
inventario.mostrarCategoriasDisponibles();
```

Resultado

```
----------------------------
=== CATEGOR□AS DISPONIBLES ===
ALIMENTOS: Productos comestibles
ELECTRONICA: Dispositivos electr□nicos
ROPA: Prendas de vestir
HOGAR: Art□culos para el hogar
```

Sistema de Biblioteca

Se implementó utilizando las clases Autor, Libro y Biblioteca, con MainBiblioteca para las tareas.

1. Creamos una biblioteca.

Código

```
// 1. Crear biblioteca
Biblioteca bib = new Biblioteca("Biblioteca Central");
```

2. Crear al menos tres autores

Código

```
// 2. Crear 3 autores
Autor a1 = new Autor("A1", "Gabriel García Márquez", "Colombiana");
Autor a2 = new Autor("A2", "Jorge Luis Borges", "Argentina");
Autor a3 = new Autor("A3", "Julio Cortázar", "Argentina");
```

3. Agregar 5 libros asociados a alguno de los Autores a la biblioteca.

Código

```
// 3. Agregar 5 libros
bib.agregarLibro("L001", "Cien años de soledad", 1967, a1);
bib.agregarLibro("L002", "El Aleph", 1949, a2);
bib.agregarLibro("L003", "Rayuela", 1963, a3);
bib.agregarLibro("L004", "Crónica de una muerte anunciada", 1981, a1);
bib.agregarLibro("L005", "Ficciones", 1944, a2);
```

4. Listar todos los libros con su información y la del autor.

Código

```
// 4. Listar Libros
System.out.println("=== TODOS LOS LIBROS ===");
bib.listarLibros();
```

Resultado

```
=== TODOS LOS LIBROS ===
ISBN: L001
T□tulo: Cien a□os de soledad
A□o: 1967
Autor: Gabriel Garc□a M□rquez
----------------------------
ISBN: L002
T□tulo: El Aleph
A□o: 1949
Autor: Jorge Luis Borges
----------------------------
ISBN: L003
T□tulo: Rayuela
A□o: 1963
Autor: Julio Cort□zar
----------------------------
ISBN: L004
T□tulo: Cr□nica de una muerte anunciada
A□o: 1981
Autor: Gabriel Garc□a M□rquez
----------------------------
ISBN: L005
T□tulo: Ficciones
A□o: 1944
Autor: Jorge Luis Borges
----------------------------
```

5. Buscar un libro por su ISBN y mostrar su información.

Código

```
// 5. Buscar por ISBN
System.out.println("=== BUSCAR POR ISBN L003 ===");
Libro encontrado = bib.buscarLibroPorIsbn("L003");
if (encontrado != null) encontrado.mostrarInfo();
```

Resultado

```
----------------------------
=== BUSCAR POR ISBN L003 ===
ISBN: L003
T□tulo: Rayuela
A□o: 1963
Autor: Julio Cort□zar
----------------------------
```

6. Filtrar y mostrar los libros publicados en un año específico.

Código

```
// 6. Filtrar por año
System.out.println("=== LIBROS DE 1949 ===");
bib.filtrarLibrosPorAnio(1949);
```

Resultado

```
----------------------------
=== LIBROS DE 1949 ===
ISBN: L002
T□tulo: El Aleph
A□o: 1949
Autor: Jorge Luis Borges
----------------------------
```

7. Eliminar un libro por su ISBN y listar los libros restantes.

Código

```
// 7. Eliminar libro
System.out.println("=== ELIMINAR L004 ===");
bib.eliminarLibro("L004");
bib.listarLibros();
```

Resultado

```
----------------------------
=== ELIMINAR L004 ===
ISBN: L001
T□tulo: Cien a□os de soledad
A□o: 1967
Autor: Gabriel Garc□a M□rquez
----------------------------
ISBN: L002
T□tulo: El Aleph
A□o: 1949
Autor: Jorge Luis Borges
----------------------------
ISBN: L003
T□tulo: Rayuela
A□o: 1963
Autor: Julio Cort□zar
----------------------------
ISBN: L005
T□tulo: Ficciones
A□o: 1944
Autor: Jorge Luis Borges
----------------------------
```

8. Mostrar la cantidad total de libros en la biblioteca.

Código

```
// 8. Cantidad total
System.out.println("=== CANTIDAD TOTAL: " + bib.obtenerCantidadLibros());
```

Resultado

```
----------------------------
=== CANTIDAD TOTAL: 4
```

9. Listar todos los autores de los libros disponibles en la biblioteca.

Código

```
// 9. Mostrar autores
System.out.println("=== AUTORES DISPONIBLES ===");
bib.mostrarAutoresDisponibles();
```

Resultado

```
=== AUTORES DISPONIBLES ===
ID: A1
Nombre: Gabriel Garc?a M?rquez
Nacionalidad: Colombiana
----------------------------
ID: A2
Nombre: Jorge Luis Borges
Nacionalidad: Argentina
----------------------------
ID: A3
Nombre: Julio Cort?zar
Nacionalidad: Argentina
----------------------------
```

Sistema de Universidad

Se implementó con clases Profesor, Curso y Universidad, enfocando la bidireccionalidad, y MainUniversidad para las tareas.

1. Crear al menos 3 profesores y 5 cursos.

Código

```
// 1. Crear 3 profesores y 5 cursos
Profesor p1 = new Profesor("P1", "Ana García", "Matemáticas");
Profesor p2 = new Profesor("P2", "Luis Martínez", "Historia");
Profesor p3 = new Profesor("P3", "Carlos López", "Programación");

Curso c1 = new Curso("C1", "Álgebra");
Curso c2 = new Curso("C2", "Cálculo");
Curso c3 = new Curso("C3", "Historia Mundial");
Curso c4 = new Curso("C4", "Programación I");
Curso c5 = new Curso("C5", "Programación II");
```

2. Agregar profesores y cursos a la universidad.

Código

```
// 2. Agregar a universidad
uni.agregarProfesor(p1);
uni.agregarProfesor(p2);
uni.agregarProfesor(p3);

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

3. Asignar profesores a cursos usando asignarProfesorACurso(...).

Código

```
// 3. Asignar profesores a cursos
uni.asignarProfesorACurso("C1", "P1");
uni.asignarProfesorACurso("C2", "P1");
uni.asignarProfesorACurso("C3", "P2");
uni.asignarProfesorACurso("C4", "P3");
uni.asignarProfesorACurso("C5", "P3");
```

4. Listar cursos con su profesor y profesores con sus cursos.

Código

```
// 4. Listar
System.out.println("=== CURSOS ===");
uni.listarCursos();
System.out.println("=== PROFESORES ===");
uni.listarProfesores();
```

Resultado

```
=== CURSOS ===
C□digo: C1
Nombre: □lgebra
Profesor: Ana Garc□a
----------------------------
C□digo: C2
Nombre: C□lculo
Profesor: Ana Garc□a
----------------------------
C□digo: C3
Nombre: Historia Mundial
Profesor: Luis Mart□nez
----------------------------
C□digo: C4
Nombre: Programaci□n I
Profesor: Carlos L□pez
----------------------------
C□digo: C5
Nombre: Programaci□n II
Profesor: Carlos L□pez
----------------------------
=== PROFESORES ===
ID: P1
Nombre: Ana Garc□a
Especialidad: Matem□ticas
Cursos a cargo: 2
 - C1: □lgebra
 - C2: C□lculo
----------------------------
ID: P2
Nombre: Luis Mart□nez
Especialidad: Historia
Cursos a cargo: 1
 - C3: Historia Mundial
----------------------------
ID: P3
Nombre: Carlos L□pez
Especialidad: Programaci□n
Cursos a cargo: 2
 - C4: Programaci□n I
 - C5: Programaci□n II
----------------------------
```

5. Cambiar el profesor de un curso y verificar que ambos lados quedan sincronizados.

Código

```
// 5. Cambiar profesor
System.out.println("=== CAMBIAR PROFESOR DE C5 A P2 ===");
uni.asignarProfesorACurso("C5", "P2");
uni.listarCursos();
uni.listarProfesores();
```

Resultado

```
----------------------------
=== CAMBIAR PROFESOR DE C5 A P2 ===
C□digo: C1
Nombre: □lgebra
Profesor: Ana Garc□a
----------------------------
C□digo: C2
Nombre: C□lculo
Profesor: Ana Garc□a
----------------------------
C□digo: C3
Nombre: Historia Mundial
Profesor: Luis Mart□nez
----------------------------
C□digo: C4
Nombre: Programaci□n I
Profesor: Carlos L□pez
----------------------------
C□digo: C5
Nombre: Programaci□n II
Profesor: Luis Mart□nez
----------------------------
ID: P1
Nombre: Ana Garc□a
Especialidad: Matem□ticas
Cursos a cargo: 2
 - C1: □lgebra
 - C2: C□lculo
----------------------------
ID: P2
Nombre: Luis Mart□nez
Especialidad: Historia
Cursos a cargo: 2
 - C3: Historia Mundial
 - C5: Programaci□n II
----------------------------
ID: P3
Nombre: Carlos L□pez
Especialidad: Programaci□n
Cursos a cargo: 1
 - C4: Programaci□n I
----------------------------
```

6. Remover un curso y confirmar que ya no aparece en la lista del profesor.

Código

```
// 6. Remover curso
System.out.println("=== ELIMINAR CURSO C1 ===");
uni.eliminarCurso("C1");
uni.listarCursos();
```

Resultado

```
----------------------------
=== ELIMINAR CURSO C1 ===
C□digo: C2
Nombre: C□lculo
Profesor: Ana Garc□a
----------------------------
C□digo: C3
Nombre: Historia Mundial
Profesor: Luis Mart□nez
----------------------------
C□digo: C4
Nombre: Programaci□n I
Profesor: Carlos L□pez
----------------------------
C□digo: C5
Nombre: Programaci□n II
Profesor: Luis Mart□nez
----------------------------
```

7. Remover un profesor y dejar profesor = null,

Código

```
// 7. Remover profesor
System.out.println("=== ELIMINAR PROFESOR P3 ===");
uni.eliminarProfesor("P3");
uni.listarCursos();
uni.listarProfesores();
```

Resultado

```
----------------------------
=== ELIMINAR PROFESOR P3 ===
C□digo: C2
Nombre: C□lculo
Profesor: Ana Garc□a
----------------------------
C□digo: C3
Nombre: Historia Mundial
Profesor: Luis Mart□nez
----------------------------
C□digo: C4
Nombre: Programaci□n I
Profesor: Sin asignar
----------------------------
C□digo: C5
Nombre: Programaci□n II
Profesor: Luis Mart□nez
----------------------------
ID: P1
Nombre: Ana Garc□a
Especialidad: Matem□ticas
Cursos a cargo: 1
 - C2: C□lculo
----------------------------
ID: P2
Nombre: Luis Mart□nez
Especialidad: Historia
Cursos a cargo: 2
 - C3: Historia Mundial
 - C5: Programaci□n II
----------------------------
```

8. Mostrar un reporte: cantidad de cursos por profesor.

Código

```
// 8. Reporte
uni.mostrarReporteCursosPorProfesor();
```

Resultado

```
----------------------------
=== REPORTE: CURSOS POR PROFESOR ===
P1 - Ana Garc□a: 1 cursos
P2 - Luis Mart□nez: 2 cursos
```

Conclusiones

Este trabajo práctico cumple con todas las conclusiones esperadas para cada sistema.

En el sistema de **Stock**, se comprendió el uso de this para acceder a atributos de instancia, se aplicaron constructores y métodos sobrecargados, se implementó toString(), se diferenciaron atributos y métodos estáticos, y se reforzó el diseño modular orientado a objetos.

En el sistema de **Biblioteca**, se reforzó la composición 1 a N entre Biblioteca y Libro, el manejo de colecciones dinámicas (ArrayList), la implementación de métodos de búsqueda, filtrado y eliminación, y la modularidad bajo el paradigma POO.

Finalmente, en el sistema de **Universidad**, se diferenciaron las relaciones bidireccionales y unidireccionales, se mantuvieron las invariantes de asociación, se practicó el uso de colecciones, búsquedas y operaciones de alta/baja, y se diseñaron métodos seguros que sincronizan ambos lados de la relación.