

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

TRABAJO PRÁCTICO N° 6 – COLECCIONES

DATOS ACADÉMICOS

- Materia: Programación II
- Trabajo Práctico: N° 6 – Colecciones
- Profesora: Cinthia Rigoni
- Tutor: Jerónimo Cortez
- Estudiante: Fabricio Puccio
- Fecha de Entrega: 02 de Octubre de 2025
- Carrera: Tecnicatura Universitaria en Programación a Distancia
- Repositorio GitHub:
<https://github.com/FabricioPuccio/TrabajoPractico6-Colecciones-Java.git>

ÍNDICE

1. INTRODUCCIÓN.....	
2. OBJETIVOS DEL TRABAJO.....	
2.1 Objetivos Generales.....	
2.2 Objetivos Específicos.....	
3. SISTEMAS IMPLEMENTADOS.....	
3.1 Sistema de Gestión de Stock.....	
3.2 Sistema de Biblioteca.....	
3.3 Sistema Universitario.....	
4. DESARROLLO TÉCNICO.....	
4.1 Arquitectura del Proyecto.....	
4.2 Conceptos de POO Aplicados.....	
4.3 Colecciones Utilizadas.....	
4.4 Sincronización de Relaciones Bidireccionales.....	
5. CONCLUSIONES.....	
5.1 Logros Alcanzados.....	
5.2 Competencias Desarrolladas.....	
5.3 Dificultades Superadas.....	
6. ANEXOS.....	
6.1 Anexo A: Diagramas UML.....	
6.2 Anexo B: Estructura de Paquetes.....	
6.3 Anexo C: Compilación y Ejecución.....	

1. INTRODUCCIÓN

El presente trabajo práctico tiene como finalidad aplicar los conceptos de colecciones en Java mediante la implementación de tres sistemas que gestionan relaciones entre objetos. Se utilizaron ArrayList como estructura principal y enumeraciones para representar categorías predefinidas, demostrando el manejo de relaciones 1:N unidireccionales y bidireccionales.

2. OBJETIVOS DEL TRABAJO

2.1 Objetivos Generales

- Desarrollar estructuras de datos dinámicas en Java usando colecciones
- Implementar sistemas con funcionalidades progresivas que refuercen P00
- Aplicar principios de diseño orientado a objetos

2.2 Objetivos Específicos

- Utilizar ArrayList como estructura de almacenamiento principal
- Implementar enumeraciones con métodos y descripciones
- Gestionar relaciones 1:N unidireccionales y bidireccionales
- Aplicar el principio "Tell Don't Ask"
- Implementar operaciones CRUD sobre colecciones

3. SISTEMAS IMPLEMENTADOS

3.1 Sistema de Gestión de Stock

Descripción

Sistema para gestionar productos en una tienda, controlando disponibilidad, precios y categorías.

Clases Implementadas

- Producto: id, nombre, precio, cantidad, categoría
- Inventario: Gestión de colección de productos
- CategoriaProducto: Enum (ALIMENTOS, ELECTRONICA, ROPA, HOGAR)

Funcionalidades

- Agregar, listar, buscar y eliminar productos
- Actualización de stock
- Filtrado por categoría y precio
- Reportes de stock

Relación

- 1 a N unidireccional (Inventario → Productos)

Resultados de Ejecución:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2
SISTEMA DE GESTIÓN DE INVENTARIO - EJERCICIO 1
=====
Producto agregado: Arroz Trimacer
Producto agregado: Smartphone Xiaomi 17
Producto agregado: Camisa Blanca
Producto agregado: Silla
Producto agregado: Laptop CX
Producto agregado: Masitas Paseo
Producto agregado: Smart TV Samsung
Producto agregado: Pantalón Cargo
Producto agregado: Mesa Redonda
Producto agregado: Monitor Samsung

EJECUTANDO LAS 10 TAREAS REQUERIDAS
=====
1) LISTAR TODOS LOS PRODUCTOS:
```

LISTA DE PRODUCTOS (10 productos):

PRODUCTO:

— ID: P001
— Nombre: Arroz Trimacer
— Precio: \$700,50
— Stock: 100 unidades
— Categoría: ALIMENTOS (Productos comestibles)

PRODUCTO:

— ID: P002
— Nombre: Smartphone Xiaomi 17
— Precio: \$550000,00
— Stock: 12 unidades
— Categoría: ELECTRONICA (Dispositivos electrónicos)

PRODUCTO:

— ID: P003
— Nombre: Camisa Blanca
— Precio: \$32000,00
— Stock: 30 unidades
— Categoría: ROPA (Prendas de vestir)

PRODUCTO:

— ID: P004
— Nombre: Silla
— Precio: \$5540,00
— Stock: 30 unidades
— Categoría: HOGAR (Artículos para el hogar)

PRODUCTO:

— ID: P005
— Nombre: Laptop CX
— Precio: \$750000,00
— Stock: 5 unidades
— Categoría: ELECTRONICA (Dispositivos electrónicos)

PRODUCTO:

— ID: P006
— Nombre: Masitas Paseo
— Precio: \$1250,00
— Stock: 100 unidades
— Categoría: ALIMENTOS (Productos comestibles)

PRODUCTO:

— ID: P007
— Nombre: Smart TV Samsung
— Precio: \$800500,00
— Stock: 4 unidades
— Categoría: ELECTRONICA (Dispositivos electrónicos)

PRODUCTO:

— ID: P008
— Nombre: Pantalón Cargo
— Precio: \$50000,00
— Stock: 20 unidades
— Categoría: ROPA (Prendas de vestir)

PRODUCTO:

— ID: P009
— Nombre: Mesa Redonda
— Precio: \$11500,00
— Stock: 15 unidades
— Categoría: HOGAR (Artículos para el hogar)

PRODUCTO:

— ID: P0010
— Nombre: Monitor Samsung
— Precio: \$255000,00
— Stock: 5 unidades
— Categoría: ELECTRONICA (Dispositivos electrónicos)

2) BUSCAR PRODUCTO POR ID 'P003':

PRODUCTO:

- | ID: P003
- | Nombre: Camisa Blanca
- | Precio: \$32000,00
- | Stock: 30 unidades
- | Categoría: ROPA (Prendas de vestir)

3) FILTRAR PRODUCTOS ELECTRÓNICOS:

Productos electrónicos encontrados: 4

PRODUCTO:

- | ID: P002
- | Nombre: Smartphone Xiaomi 17
- | Precio: \$550000,00
- | Stock: 12 unidades
- | Categoría: ELECTRONICA (Dispositivos electrónicos)

PRODUCTO:

- | ID: P005
- | Nombre: Laptop CX
- | Precio: \$750000,00
- | Stock: 5 unidades
- | Categoría: ELECTRONICA (Dispositivos electrónicos)

PRODUCTO:

- | ID: P007
- | Nombre: Smart TV Samsung
- | Precio: \$800500,00
- | Stock: 4 unidades
- | Categoría: ELECTRONICA (Dispositivos electrónicos)

PRODUCTO:

- | ID: P0010
- | Nombre: Monitor Samsung
- | Precio: \$255000,00
- | Stock: 5 unidades
- | Categoría: ELECTRONICA (Dispositivos electrónicos)

4) ELIMINAR PRODUCTO 'P004':

Producto eliminado exitosamente

5) ACTUALIZAR STOCK DE 'P001' A 150 UNIDADES:

Stock actualizado exitosamente

6) TOTAL DE STOCK DISPONIBLE:

Stock total en inventario: 341 unidades

7) PRODUCTO CON MAYOR STOCK:

Producto con mayor stock:

PRODUCTO:

- | ID: P001
- | Nombre: Arroz Trimacer
- | Precio: \$700,50
- | Stock: 150 unidades
- | Categoría: ALIMENTOS (Productos comestibles)

8) FILTRAR PRODUCTOS ENTRE \$1000 Y \$3000:

Productos encontrados: 2

PRODUCTO:

- | ID: P006
- | Nombre: Masitas Paseo
- | Precio: \$1250,00
- | Stock: 100 unidades
- | Categoría: ALIMENTOS (Productos comestibles)

<p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P009 Nombre: Mesa Redonda Precio: \$11500,00 Stock: 15 unidades Categoría: HOGAR (Artículos para el hogar) <p>9) CATEGORÍAS DISPONIBLES:</p> <p>Categorías del sistema:</p> <ul style="list-style-type: none"> ALIMENTOS - Productos comestibles ELECTRONICA - Dispositivos electrónicos ROPA - Prendas de vestir HOGAR - Artículos para el hogar
<p>10) LISTA FINAL DE PRODUCTOS:</p> <p>LISTA DE PRODUCTOS (9 productos):</p> <p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P001 Nombre: Arroz Trimacer Precio: \$700,50 Stock: 150 unidades Categoría: ALIMENTOS (Productos comestibles) <p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P002 Nombre: Smartphone Xiaomi 17 Precio: \$550000,00 Stock: 12 unidades Categoría: ELECTRONICA (Dispositivos electrónicos)
<p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P003 Nombre: Camisa Blanca Precio: \$32000,00 Stock: 30 unidades Categoría: ROPA (Prendas de vestir) <p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P005 Nombre: Laptop CX Precio: \$750000,00 Stock: 5 unidades Categoría: ELECTRONICA (Dispositivos electrónicos)
<p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P006 Nombre: Masitas Paseo Precio: \$1250,00 Stock: 100 unidades Categoría: ALIMENTOS (Productos comestibles) <p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P007 Nombre: Smart TV Samsung Precio: \$800500,00 Stock: 4 unidades Categoría: ELECTRONICA (Dispositivos electrónicos)
<p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P008 Nombre: Pantalón Cargo Precio: \$50000,00 Stock: 20 unidades Categoría: ROPA (Prendas de vestir) <p>PRODUCTO:</p> <ul style="list-style-type: none"> ID: P009 Nombre: Mesa Redonda Precio: \$11500,00 Stock: 15 unidades Categoría: HOGAR (Artículos para el hogar)

```

PRODUCTO:
├─ ID: P0010
├─ Nombre: Monitor Samsung
├─ Precio: $255000,00
├─ Stock: 5 unidades
├─ Categoría: ELECTRONICA (Dispositivos electrónicos)

```

```

ESTADÍSTICAS DEL INVENTARIO:
├─ Productos totales: 9
├─ Stock total: 341 unidades
├─ Producto con mayor stock: Arroz Trimacer

```

EJECUCIÓN COMPLETADA - EJERCICIO 1 LISTO!

Process finished with exit code 0

3.2 Sistema de Biblioteca

Descripción

Sistema para gestionar libros y autores con relación de agregación.

Clases Implementadas

- Biblioteca: Contenedor principal de libros
- Libro: isbn, título, año, autor
- Autor: id, nombre, nacionalidad

Funcionalidades

- Gestión completa de libros
- Búsqueda por ISBN
- Filtrado por año
- Listado de autores

Relación

- Agregación 1 a N (Biblioteca → Libros)

Resultados de Ejecución:

```

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.
SISTEMA DE GESTIÓN DE BIBLIOTECA - EJERCICIO 2
=====
Libro agregado: Cien años de soledad
Libro agregado: El Aleph
Libro agregado: Rayuela
Libro agregado: Crónica de una muerte anunciada
Libro agregado: Ficciones

EJECUTANDO TAREAS REQUERIDAS - BIBLIOTECA
|||||

1) BIBLIOTECA CREADA:
Biblioteca: Biblioteca Central TUPAD
Biblioteca inicializada con datos de prueba

2) LISTAR TODOS LOS LIBROS:

LIBROS EN BIBLIOTECA 'Biblioteca Central TUPAD' (5 libros):
LIBRO:
├─ ISBN: ISBN001
├─ Título: Cien años de soledad
├─ Año: 1967
├─ Autor: Gabriel García Márquez

LIBRO:
├─ ISBN: ISBN002
├─ Título: El Aleph
├─ Año: 1949
├─ Autor: Jorge Luis Borges

```

LIBRO:
| ISBN: ISBN003
| Título: Rayuela
| Año: 1963
| Autor: Julio Cortázar

LIBRO:
| ISBN: ISBN004
| Título: Crónica de una muerte anunciada
| Año: 1981
| Autor: Gabriel García Márquez

LIBRO:
| ISBN: ISBN005
| Título: Ficciones
| Año: 1944
| Autor: Jorge Luis Borges

3) BUSCAR LIBRO POR ISBN 'ISBN004':

LIBRO:
| ISBN: ISBN004
| Título: Crónica de una muerte anunciada
| Año: 1981
| Autor: Gabriel García Márquez

Información del autor:
AUTOR:
| ID: A001
| Nombre: Gabriel García Márquez
| Nacionalidad: Colombiana

4) FILTRAR LIBROS DEL AÑO 1949:
Libros encontrados del año 1949: 1

LIBRO:
| ISBN: ISBN002
| Título: El Aleph
| Año: 1949
| Autor: Jorge Luis Borges

5) ELIMINAR LIBRO 'ISBN002':
Libro eliminado exitosamente

6) CANTIDAD TOTAL DE LIBROS:
Total de libros en biblioteca: 4

7) AUTORES DISPONIBLES EN BIBLIOTECA:
Autores encontrados: 3

AUTOR:
| ID: A001
| Nombre: Gabriel García Márquez
| Nacionalidad: Colombiana

AUTOR:
| ID: A003
| Nombre: Julio Cortázar
| Nacionalidad: Argentina

AUTOR:
| ID: A002
| Nombre: Jorge Luis Borges
| Nacionalidad: Argentina


```
8) LISTA FINAL DE LIBROS:

LIBROS EN BIBLIOTECA 'Biblioteca Central TUPAD' (4 libros):
LIBRO:
| ISBN: ISBN001
| Título: Cien años de soledad
| Año: 1967
| Autor: Gabriel García Márquez

LIBRO:
| ISBN: ISBN003
| Título: Rayuela
| Año: 1963
| Autor: Julio Cortázar

LIBRO:
| ISBN: ISBN004
| Título: Crónica de una muerte anunciada
| Año: 1981
| Autor: Gabriel García Márquez

LIBRO:
| ISBN: ISBN005
| Título: Ficciones
| Año: 1944
| Autor: Jorge Luis Borges

9) ESTADÍSTICAS DE LA BIBLIOTECA:
RESUMEN FINAL:
| Biblioteca: Biblioteca Central TUPAD
| Total de libros: 4
| Autores únicos: 3
| Libros por autor disponibles para consulta

EJECUCIÓN COMPLETADA - EJERCICIO 2 LISTO!

Process finished with exit code 0
```

3.3 Sistema Universitario

Descripción

Sistema académico con relaciones bidireccionales entre profesores y cursos.

Clases Implementadas

- Universidad: Administración central
- Profesor: id, nombre, especialidad, lista de cursos
- Curso: código, nombre, profesor

Funcionalidades

- Asignación bidireccional profesor-curso
- Sincronización automática
- Reportes académicos

Relación

- Bidireccional 1 a N (Profesor ↔ Cursos)

Resultados de Ejecución:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.
SISTEMA DE GESTIÓN UNIVERSITARIA - EJERCICIO 3
=====
RELACIONES BIDIRECCIONALES 1:N
SINCRONIZACIÓN AUTOMÁTICA
=====
Profesor agregado: Jerónimo Cortez
Profesor agregado: Carlos López
Profesor agregado: María Fernández
Curso agregado: Programación Orientada a Objetos
Curso agregado: Algoritmos Avanzados
Curso agregado: Base de Datos I
Curso agregado: Visualizar Datos en R
Curso agregado: Patrones de Diseño

EJECUTANDO 8 TAREAS REQUERIDAS - UNIVERSIDAD
////////////////////////////////////
```

1) AGREGAR PROFESORES Y CURSOS:

Universidad: Universidad Tecnológica Nacional

Resumen inicial:

- |— Profesores: 3
- |— Cursos: 5

2) ASIGNAR PROFESORES A CURSOS:

Curso 'Programación Orientada a Objetos' agregado al profesor Jerónimo Cortez

* Programación Orientada a Objetos → Jerónimo Cortez

Curso 'Algoritmos Avanzados' agregado al profesor María Fernández

* Algoritmos Avanzados → María Fernández

Curso 'Base de Datos I' agregado al profesor Carlos López

* Base de Datos I → Carlos López

Curso 'Visualizar Datos en R' agregado al profesor María Fernández

* Visualizar Datos en R → María Fernández

Curso 'Patrones de Diseño' agregado al profesor Jerónimo Cortez

* Patrones de Diseño → Jerónimo Cortez

3) LISTAR CURSOS Y PROFESORES:

--- LISTA DE CURSOS ---

CURSOS DE UNIVERSIDAD TECNOLÓGICA NACIONAL:

CURSO:

- |— Código: C001
- |— Nombre: Programación Orientada a Objetos
- |— Profesor: Jerónimo Cortez

CURSO:

- |— Código: C002
- |— Nombre: Algoritmos Avanzados
- |— Profesor: María Fernández

CURSO:

- |— Código: C003
- |— Nombre: Base de Datos I
- |— Profesor: Carlos López

CURSO:

- |— Código: C004
- |— Nombre: Visualizar Datos en R
- |— Profesor: María Fernández

CURSO:

- |— Código: C005
- |— Nombre: Patrones de Diseño
- |— Profesor: Jerónimo Cortez

--- LISTA DE PROFESORES ---

PROFESORES DE UNIVERSIDAD TECNOLÓGICA NACIONAL:

PROFESOR:

- └ ID: P001
- └ Nombre: Jerónimo Cortez
- └ Especialidad: Programación II
- └ Cursos asignados: 2

PROFESOR:

- └ ID: P002
- └ Nombre: Carlos López
- └ Especialidad: Base de Datos I
- └ Cursos asignados: 1

PROFESOR:

- └ ID: P003
- └ Nombre: María Fernández
- └ Especialidad: Probabilidad y Estadística I
- └ Cursos asignados: 2

--- CURSOS POR PROFESOR ---

Jerónimo Cortez:

CURSOS DEL PROFESOR JERÓNIMO CORTEZ:

- └ C001 - Programación Orientada a Objetos
- └ C005 - Patrones de Diseño

Carlos López:

CURSOS DEL PROFESOR CARLOS LÓPEZ:

- └ C003 - Base de Datos I

María Fernández:

CURSOS DEL PROFESOR MARÍA FERNÁNDEZ:

- └ C002 - Algoritmos Avanzados
- └ C004 - Visualizar Datos en R

4) CAMBIAR PROFESOR DE UN CURSO:

Cambiando profesor de 'Algoritmos Avanzados' (C002)

Profesor actual: María Fernández

Curso 'Algoritmos Avanzados' removido del profesor María Fernández

Curso 'Algoritmos Avanzados' agregado al profesor Jerónimo Cortez

Nuevo profesor asignado: Jerónimo Cortez

Verificando sincronización:

- Curso tiene profesor: Jerónimo Cortez
- Profesor tiene curso: Si

5) REMOVER UN CURSO:

Eliminando curso 'Base de Datos I' (C003)

Profesor asignado antes: Carlos López

Curso 'Base de Datos I' removido del profesor Carlos López

Curso eliminado exitosamente

Cursos del profesor Carlos López después:

El profesor Carlos López no tiene cursos asignados

6) REMOVER UN PROFESOR:

Eliminando profesor 'Carlos López' (P002)

Profesor eliminado exitosamente

Estado de cursos después de eliminar profesor:

CURSOS DE UNIVERSIDAD TECNOLÓGICA NACIONAL:

CURSO:

- └ Código: C001
- └ Nombre: Programación Orientada a Objetos
- └ Profesor: Jerónimo Cortez

CURSO:

- └ Código: C002
- └ Nombre: Algoritmos Avanzados
- └ Profesor: Jerónimo Cortez

```
CURSO:
└─ Código: C004
└─ Nombre: Visualizar Datos en R
└─ Profesor: María Fernández

CURSO:
└─ Código: C005
└─ Nombre: Patrones de Diseño
└─ Profesor: Jerónimo Cortez

7) LISTA FINAL:
ESTADO FINAL DE LA UNIVERSIDAD
=====

PROFESORES DE UNIVERSIDAD TECNOLÓGICA NACIONAL:
PROFESOR:
└─ ID: P001
└─ Nombre: Jerónimo Cortez
└─ Especialidad: Programación II
└─ Cursos asignados: 3

PROFESOR:
└─ ID: P003
└─ Nombre: María Fernández
└─ Especialidad: Probabilidad y Estadística I
└─ Cursos asignados: 1

CURSOS DE UNIVERSIDAD TECNOLÓGICA NACIONAL:
CURSO:
└─ Código: C001
└─ Nombre: Programación Orientada a Objetos
└─ Profesor: Jerónimo Cortez

CURSO:
└─ Código: C002
└─ Nombre: Algoritmos Avanzados
└─ Profesor: Jerónimo Cortez

CURSO:
└─ Código: C004
└─ Nombre: Visualizar Datos en R
└─ Profesor: María Fernández

CURSO:
└─ Código: C005
└─ Nombre: Patrones de Diseño
└─ Profesor: Jerónimo Cortez

8) REPORTE: CURSOS POR PROFESOR

REPORTE: CURSOS POR PROFESOR
=====
Jerónimo Cortez: 3 cursos
María Fernández: 1 cursos

ESTADÍSTICAS FINALES:
└─ Universidad: Universidad Tecnológica Nacional
└─ Profesores activos: 2
└─ Cursos activos: 4
└─ Relaciones bidireccionales verificadas ✓

¡EJECUCIÓN COMPLETADA! - TRES EJERCICIOS TERMINADOS
=====

Process finished with exit code 0
```

4. DESARROLLO TÉCNICO

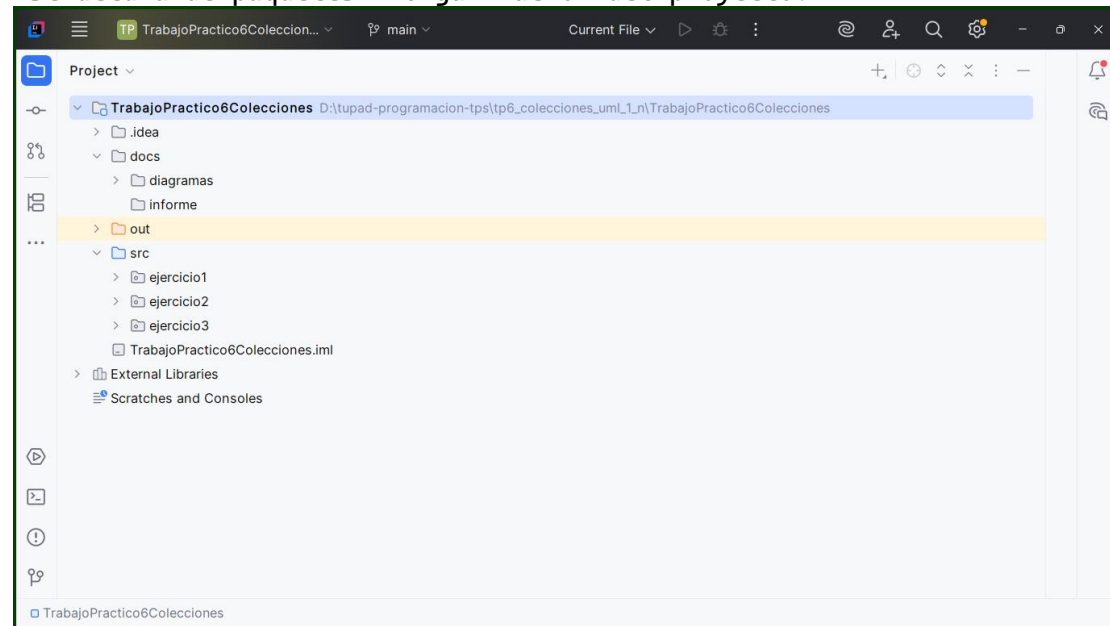
4.1 Arquitectura del Proyecto

```

TrabajoPractico6_Colecciones/
├── src/
│   ├── ejercicio1/      # Sistema de Stock
│   ├── ejercicio2/      # Sistema de Biblioteca
│   └── ejercicio3/      # Sistema Universitario
├── docs/
│   ├── diagramas/      # Diagramas UML
│   └── informe/        # Este documento
└── README.md

```

Estructura de paquetes - Organización del proyecto:

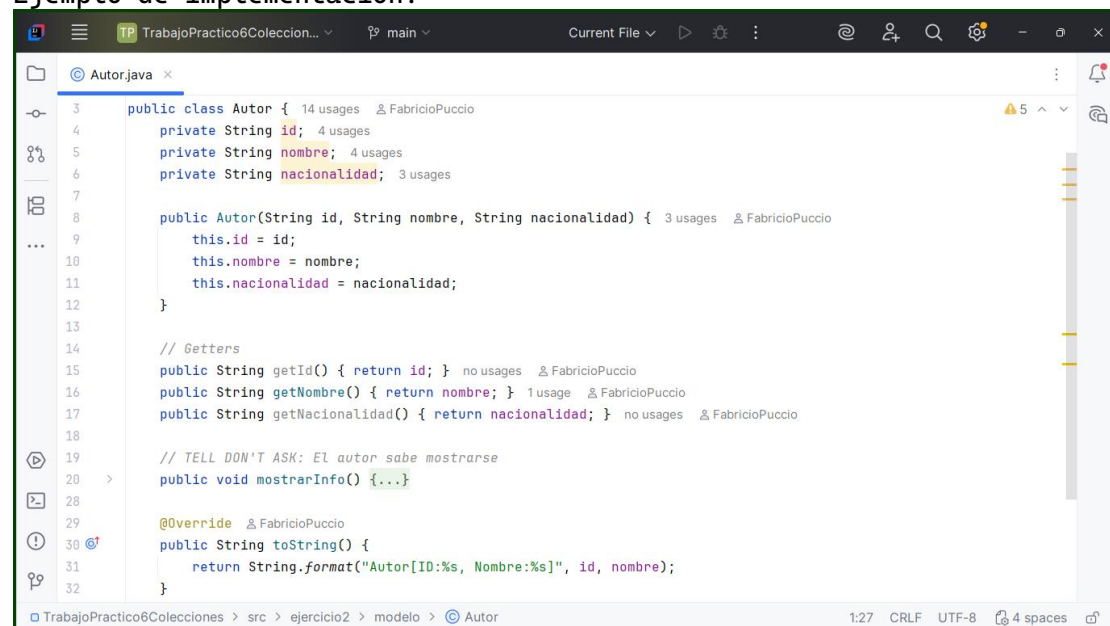


4.2 Conceptos de P00 Aplicados

Encapsulamiento

Todos los atributos de las clases son privados, con métodos públicos de acceso.

Ejemplo de implementación:



Relaciones entre Objetos

- Ejercicio 1: Relación 1:N unidireccional
- Ejercicio 2: Relación de agregación 1:N
- Ejercicio 3: Relación bidireccional 1:N

Principio "Tell Don't Ask"

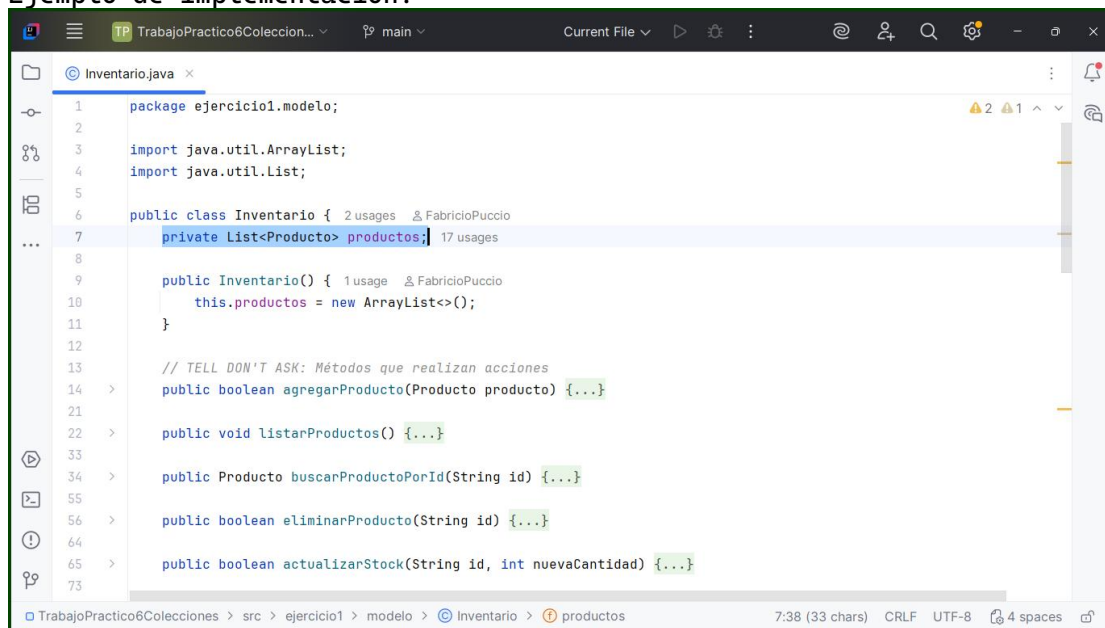
Los objetos realizan sus propias operaciones en lugar de exponer sus datos.

4.3 Colecciones Utilizadas

ArrayList

Estructura principal para todas las colecciones del proyecto.

Ejemplo de implementación:



```
1 package ejercicio1.modelo;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Inventario { 2 usages  & FabricioPuccio
7     private List<Producto> productos; 17 usages
8
9     public Inventario() { 1 usage  & FabricioPuccio
10         this.productos = new ArrayList<>();
11     }
12
13     // TELL DON'T ASK: Métodos que realizan acciones
14     public boolean agregarProducto(Producto producto) {...}
15
16     public void listarProductos() {...}
17
18     public Producto buscarProductoPorId(String id) {...}
19
20     public boolean eliminarProducto(String id) {...}
21
22     public boolean actualizarStock(String id, int nuevaCantidad) {...}
23 }
```

Enumeraciones

Implementadas con constructores y métodos adicionales.

Ejemplo de implementación:

```

1 package ejercicio1.modelo;
2
3 public enum CategoriaProducto { 17 usages  ⚡ FabricioPuccio
4     //Constantes - Instancias - Valores - Miembros
5     ALIMENTOS("Productos comestibles"), 2 usages
6     ELECTRONICA("Dispositivos electrónicos"), 5 usages
7     ROPA("Prendas de vestir"), 2 usages
8     HOGAR("Artículos para el hogar"); 2 usages
9
10    private final String descripcion; 2 usages
11
12    //Este constructor es AUTOMÁTICAMENTE private
13    CategoriaProducto(String descripcion) { 8 usages  ⚡ FabricioPuccio
14        this.descripcion = descripcion;
15    }
16
17    public String getDescription() { 2 usages  ⚡ FabricioPuccio
18        return descripcion;
19    }
20 }
21

```

4.4 Sincronización de Relaciones Bidireccionales

En el sistema universitario, se implementó la sincronización automática:

Ejemplo de implementación:

```

3 public class Curso { 24 usages  ⚡ FabricioPuccio
4
5     // TELL DON'T ASK: El curso gestiona su profesor con sincronización bidireccional
6     public void setProfesor(Profesor nuevoProfesor) { 2 usages  ⚡ FabricioPuccio
7         // Remover del profesor actual si existe
8         if(this.profesor != null && this.profesor != nuevoProfesor) {
9             this.profesor.eliminarCurso(this);
10        }
11
12        // Asignar nuevo profesor
13        this.profesor = nuevoProfesor;
14
15        // Sincronización bidireccional
16        if(nuevoProfesor != null && !nuevoProfesor.getCursos().contains(this)) {
17            nuevoProfesor.agregarCurso(this);
18        }
19    }
20
21    public void removerProfesor() { this.profesor = null; }
22
23    public void mostrarInfo() {...}
24
25    @Override  ⚡ FabricioPuccio
26    public String toString() { return String.format("Curso[Código:%s, Nombre:%s]", this.codigo, this.nombre); }
27 }
28

```

5. CONCLUSIONES

5.1 Logros Alcanzados

1. Manejo de Colecciones: Uso avanzado de ArrayList en diferentes contextos
2. Diseño de Enumeraciones: Implementación de enums con comportamiento
3. Relaciones entre Objetos: Gestión de relaciones 1:N unidireccionales y bidireccionales
4. Arquitectura de Software: Diseño modular y separación de responsabilidades

5. Principios SOLID: Aplicación de buenas prácticas de P00

5.2 Competencias Desarrolladas

- Gestión de dependencias entre objetos
- Sincronización de relaciones bidireccionales
- Implementación de operaciones CRUD sobre colecciones
- Diseño de sistemas extensibles y mantenibles
- Aplicación del principio "Tell Don't Ask"

5.3 Dificultades Superadas

1. Sincronización Bidireccional: Implementación de invariantes de asociación
2. Gestión de Colecciones: Optimización de búsquedas y filtrados
3. Diseño Modular: Separación clara de responsabilidades entre clases

6. ANEXOS

6.1 Anexo A: Diagramas UML

Los diagramas UML de cada sistema se encuentran en la carpeta docs/diagramas/:

- ejercicio1.uxf - Sistema de Stock
- ejercicio2.uxf - Sistema de Biblioteca
- ejercicio3.uxf - Sistema Universitario

6.2 Anexo B: Estructura de Paquetes

```
src/  
├── ejercicio1/  
│   ├── main/  
│   │   └── MainStock.java  
│   ├── modelo/  
│   │   ├── CategoriaProducto.java  
│   │   ├── Inventario.java  
│   │   └── Producto.java  
│   └── servicios/  
│       └── GestorInventario.java  
├── ejercicio2/  
│   ├── main/  
│   │   └── MainBiblioteca.java  
│   ├── modelo/  
│   │   ├── Autor.java  
│   │   ├── Biblioteca.java  
│   │   └── Libro.java  
│   └── servicios/  
│       └── GestorBiblioteca.java  
└── ejercicio3/  
    ├── main/  
    │   └── MainUniversidad.java  
    ├── modelo/  
    │   ├── Curso.java  
    │   ├── Profesor.java  
    │   └── Universidad.java  
    └── servicios/  
        └── GestorUniversidad.java
```


6.3 Anexo C: Compilación y Ejecución

```
# Compilar todos los ejercicios
find src -name "*.java" > sources.txt
javac -d bin @sources.txt

# Ejecutar cada sistema
java -cp bin ejercicio1.MainStock
java -cp bin ejercicio2.MainBiblioteca
java -cp bin ejercicio3.MainUniversidad
```

REFERENCIAS

- Documentación Oficial de Java:
<https://docs.oracle.com/javase/8/docs/>
- Java Collections Framework:
<https://docs.oracle.com/javase/8/docs/technotes/guides/collections/>
- Principios SOLID: https://es.wikipedia.org/wiki/Principios_SOLID

FIRMA DEL ESTUDIANTE

Fabricio Puccio