**Nombre Alumno/s**: Fernandez Alejandro Facundo **DNI:** 40047821

Gomez Angela Rebeca **DNI:** 40488634

**Nombre Profesor:** Aldo Mettini, Andrea Airaldi

**Grupo Laboratorio:** 3 **TP:** 4 **Fecha de entrega:** 22/09/2025

Conceptos incorporados e implementados para la resolución de los ejercicios de la serie N°4:

Durante el desarrollo se aplicaron conceptos de encapsulamiento, clases, objetos y reutilización de código, diferencias entre contenedores homogéneos y heterogéneos, así como también la utilidad de las estructuras indexadas.

En este trabajo apreciamos el concepto de reutilización teniendo la capacidad de las colecciones de manejar elementos de distintas clases, utilizado para hacer referencia a datos estructurados en los cuales cada elemento tiene un significado similar, aunque su valor dependa de la posición. Un ejemplo sería el ArrayList que trabaja con elementos de tipo OBJECT, los cuales se conocen por almacenar referencias y permiten almacenar tipos primitivos. Para guardar tipos primitivos se deben usar clases conocidas como wrapper (envoltura): Byte para byte, Short para short, Integer para int, Long para long, Boolean para boolean, Float para float, Double para double y Character para char.

Los ArrayList siendo estructuras de datos de tipo Array dinámicas, permiten aumentar el tamaño, agregar o quitar elementos accediendo a cualquier elemento de la lista mediante un índice.

Un contenedor heterogéneo permite almacenar objetos de diferentes clases. Así definimos una lista heterogénea:

private ArrayList productos;

Los contenedores homogéneos son aquellos que contienen elementos de la misma clase, es decir que cada uno de sus elementos es de la misma clase. Así una lista homogénea:

private ArrayList<Producto> productos;

Una colección indexada puede representarse con arreglos, lo que permite trabajar con índices de acceso directo:

Punto[ ] puntos = new Punto[6];

Ejemplos:

Coleccion indexada

public class ArrayDePuntos {

private Punto[] puntos = new Punto[6];

public ArrayDePuntos() {

puntos[0] = new Punto(2,3);

puntos[1] = new Punto(4,5);

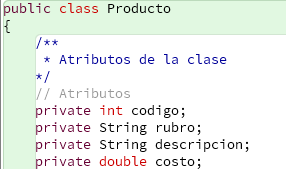
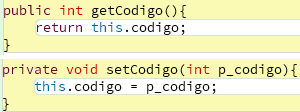
// ...

Colecciones homogéneas y heterogéneas

private ArrayList productos; // heterogénea

private ArrayList<Producto> productos; // homogénea con genéricos

Encapsulamiento

Generic y wrapper

