

Práctica #08 – Colecciones en Java.

Práctica “Mini-inventario con colecciones”.

Laboratorio de Programación Orientada a Objetos.

Alumno: Jeremy Uriel Rossell Segura

Matrícula: 2173396

Materia: Laboratorio de Programación Orientada a Objetos.

Docente: Jorge Alberto Islas Pineda

Compilación

El programa compiló adecuadamente.

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** 2173396_practica08
- Current File:** LaboratorioColecciones.java
- Run:** LaboratorioColecciones
- Code Content:**

```
...  
+ /usr/java/jdk-25-oracle-x64/bin/java -javaagent:/home/jeremy/.local/share/JetBrains/Toolbox/apps/intellij-idea-ultimate/lib/idea_rt.jar=37349 -Dfile.encoding=UTF-8 -Dsun.stdout.er  
+== ArrayList: Exhibición ==  
+ Producto{id=1, nombre='Teclado', categoria='Perifericos'}  
+ Producto{id=2, nombre='Mouse', categoria='Perifericos'}  
+ Producto{id=3, nombre='Monitor', categoria='Pantallas'}  
+ Producto{id=4, nombre='Cable HDMI', categoria='Oferta'}  
  
+== LinkedList: Reabastecer ==  
+ Producto{id=6, nombre='Webcam', categoria='Perifericos'}  
+ Producto{id=5, nombre='Laptop', categoria='Computo'}  
  
+== HashSet: Categorías únicas ==  
+ [Pantallas, Computo, Perifericos, Oferta]  
  
+== HashMap: Consulta por id ==  
+ id=3 -> Producto{id=3, nombre='Monitor', categoria='Pantallas'}  
  
+ Recorriendo HashMap (entrySet):  
+ 1 -> Producto{id=1, nombre='Teclado', categoria='Perifericos'}  
+ 2 -> Producto{id=2, nombre='Mouse', categoria='Perifericos'}  
+ 3 -> Producto{id=3, nombre='Monitor', categoria='Pantallas'}  
+ 4 -> Producto{id=4, nombre='Cable HDMI', categoria='Oferta'}  
+ 5 -> Producto{id=5, nombre='Laptop', categoria='Computo'}  
+ 6 -> Producto{id=6, nombre='Webcam', categoria='Perifericos'}  
  
+== Iterator: eliminar categoría 'Oferta' en listaExhibicion ==  
+ Despues de eliminar:  
+ Producto{id=1, nombre='Teclado', categoria='Perifericos'}  
+ Producto{id=2, nombre='Mouse', categoria='Perifericos'}  
+ Producto{id=3, nombre='Monitor', categoria='Pantallas'}  
  
+ Process finished with exit code 0
```

Depuración

El programa se depuró adecuadamente.

```
Project: 2173396_practica08 Version control: Current File

Debug: Threads & Variables Console

... /usr/java/jdk-25-oracle-x64/bin/java -agentlib:jdwp=transport=dt_socket,address=127.0.0.1:54335,suspend=y,server=n -javaagent:/home/jeremy/.local/share/JetBrains/Toolbox/apps/inte
Connected to the target VM, address: '127.0.0.1:54335', transport: 'socket'
== ArrayList: Exhibición ==
Producto{id=1, nombre='Teclado', categoria='Perifericos'}
Producto{id=2, nombre='Mouse', categoria='Perifericos'}
Producto{id=3, nombre='Monitor', categoria='Pantallas'}
Producto{id=4, nombre='Cable HDMI', categoria='Oferta'}

== LinkedList: Reabastecer ==
Producto{id=6, nombre='Webcam', categoria='Perifericos'}
Producto{id=5, nombre='Laptop', categoria='Computo'}

== HashSet: Categorías únicas ==
[Pantallas, Computo, Perifericos, Oferta]

== HashMap: Consulta por id ==
id=3 -> Producto{id=3, nombre='Monitor', categoria='Pantallas'}

Recorriendo HashMap (entrySet):
1 -> Producto{id=1, nombre='Teclado', categoria='Perifericos'}
2 -> Producto{id=2, nombre='Mouse', categoria='Perifericos'}
3 -> Producto{id=3, nombre='Monitor', categoria='Pantallas'}
4 -> Producto{id=4, nombre='Cable HDMI', categoria='Oferta'}
5 -> Producto{id=5, nombre='Laptop', categoria='Computo'}
6 -> Producto{id=6, nombre='Webcam', categoria='Perifericos'}

== Iterator: eliminar categoría 'Oferta' en listaExhibicion ==
Después de eliminar:
Producto{id=1, nombre='Teclado', categoria='Perifericos'}
Producto{id=2, nombre='Mouse', categoria='Perifericos'}
Producto{id=3, nombre='Monitor', categoria='Pantallas'}

Disconnected from the target VM, address: '127.0.0.1:54335', transport: 'socket'

Cannot record performance: Cannot start the profiler: kernel variables are not configured
Disable Recording Time with Run to Cursor Configure
```

Código

LaboratorioProducciones.java y Producto.java

The screenshot shows the IntelliJ IDEA interface with two code editors open. On the left, the project structure for '2173396_practica08' is visible, showing a package named 'inventario' containing 'LaboratorioColecciones.java' and 'Producto.java'. The right side displays the content of these two files.

```
Project: 2173396_practica08 ~/jetbrains/
  src/inventario
    - LaboratorioColecciones.java
    - Producto.java
```

LaboratorioColecciones.java:

```
1 package inventario;
2
3 import java.util.*;
4
5 public class LaboratorioColecciones {
6     public static void main(String[] args) {
7         // A. ArrayList
8         System.out.println("==> ArrayList: Exhibición ==");
9
10        ArrayList<Producto> listaExhibicion = new ArrayList<>();
11        listaExhibicion.add(new Producto(id: 1, nombre: "Teclado", categoria: "Perifericos"));
12        listaExhibicion.add(new Producto(id: 2, nombre: "Mouse", categoria: "Perifericos"));
13        listaExhibicion.add(new Producto(id: 3, nombre: "Monitor", categoria: "Perifericos"));
14        listaExhibicion.add(new Producto(id: 4, nombre: "Cable HDMI", categoria: "Perifericos"));
15
16        for (Producto p : listaExhibicion) {
17            System.out.println(p);
18        }
19
20        // B. LinkedList
21        System.out.println("\n==> LinkedList: Reabastecer ==");
22
23        LinkedList<Producto> colaReabastecer = new LinkedList<>();
24        colaReabastecer.addFirst(new Producto(id: 6, nombre: "Webcam", categoria: "Perifericos"));
25        colaReabastecer.addLast(new Producto(id: 5, nombre: "Laptop", categoria: "Computadoras"));
26
27        for (Producto p : colaReabastecer) {
28            System.out.println(p);
29        }
30
31        // C. HashSet
32        System.out.println("\n==> HashSet: Categorías únicas ==");
33
34        HashSet<String> categorias = new HashSet<>();
35    }
```

Producto.java:

```
1 package inventario;
2
3 import java.util.Objects;
4
5 public class Producto {
6     private int id;
7     private String nombre;
8     private String categoria;
9
10    public Producto(int id, String nombre, String categoria) {
11        this.id = id;
12        this.nombre = nombre;
13        this.categoria = categoria;
14    }
15
16    // getters
17    public int getId() {
18        return id;
19    }
20
21    public String getNombre() {
22        return nombre;
23    }
24
25    public String getCategoría() {
26        return categoria;
27    }
28
29    @Override
30    public String toString() {
31        return "Producto{id=" + id + ", nombre=" + nombre + ", categoria=" + categoria + "}";
32    }
33
34    // equals & hashCode basados solo en id
35}
```