



**UNIVERSIDAD AUTONOMA DE NUEVO LEÓN**  
**FACULTAD DE CIENCIAS FISICO MATEMATICAS**

*“Practica 8”*

**UNIDAD DE APRENDIZAJE:** Programación Orientada a  
Objetos

**MAESTRO:** JORGE ALBERTO ISLAS PINEDA

**CARRERA:** LICENCIATUA EN CIENCIAS  
COMPUTACIONALES

## Output:

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The left sidebar displays a Projects view with several projects listed under 'Practical' and a Navigator view for the current project 'LaboratorioColecciones'. The main area shows the code editor for 'LaboratorioColecciones.java' and the output window for 'Output - Practical (run)'. The code in the editor is as follows:

```
1 // 
2 // Click nb:/nghost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 // Click nb:/nghost/SystemFileSystem/Templates/Main.java to edit this template
4 /*
5  * package practica8;
6  * import java.util.*;
7  */
8 /*
9  * 
10 * @author garza
11 */
12 public class LaboratorioColecciones {
13     public static void main(String[] args) {
14         ArrayList<Exhibicion> ==
15             new ArrayList<Exhibicion>();
16             Exhibicion(1, "Teclado", "Perifericos");
17             Exhibicion(2, "Mouse", "Perifericos");
18             Exhibicion(3, "Monitor", "Pantallas");
19             Exhibicion(4, "Cable HDMI", "Oftalia");
20 
21             == LinkedList: Desarrollador ==
22             new LinkedList();
23             Desarrollador(1, "Webcam", "Perifericos");
24             Desarrollador(2, "Laptop", "Computo");
25 
26             == HashSet: Categorias & Ofertas ==
27             new HashSet();
28             Ofertas();
29 
30             == Hashmap: Consulta por id ==
31             new Hashmap();
32             ConsultaPorId(1, "Monitor", "Pantallas");
33 
34             Recorriendo HashMap (entrySet):
35             1 --> Producto(1, "Teclado", "Perifericos")
36             2 --> Producto(2, "Mouse", "Perifericos")
37             3 --> Producto(3, "Monitor", "Pantallas")
38             4 --> Producto(4, "Cable HDMI", "Oftalia")
39             5 --> Producto(5, "Laptop", "Computo")
40             6 --> Producto(6, "Webcam", "Perifericos")
41 
42             Iterarset: almacenar categoria & 'Ofertas' en listaExhibicion ==
43             Desarrollador de eliminar:
44             Producto(1, "Teclado", "Perifericos")
45             Producto(2, "Mouse", "Perifericos")
46             Producto(3, "Monitor", "Pantallas")
47             Producto(4, "Cable HDMI", "Oftalia")
48             Producto(5, "Laptop", "Computo")
49             Producto(6, "Webcam", "Perifericos")
50 
51             BUILD SUCCESSFUL (total time: 0 seconds)
```

The output window shows the execution results, including the creation of an ArrayList named 'Exhibicion' containing six 'Exhibicion' objects, the creation of a LinkedList named 'Desarrollador' containing two 'Desarrollador' objects, the creation of a HashSet named 'Categorias & Ofertas' containing one 'Ofertas' object, and the iteration over a HashMap to store categories and offers into a 'listaExhibicion' list. Finally, it shows the removal of specific products from the HashMap.

### • LaboratorioColecciones

```
package practica8;

import java.util.*;

/**
 *
 * @author garza
 */

public class LaboratorioColecciones {

    public static void main(String[] args) {

        // ===== A. ArrayList =====
        System.out.println("== ArrayList: Exhibición ==");

        ArrayList<Producto> listaExhibicion = new ArrayList<>();

        listaExhibicion.add(new Producto(1, "Teclado", "Perifericos"));
        listaExhibicion.add(new Producto(2, "Mouse", "Perifericos"));
        listaExhibicion.add(new Producto(3, "Monitor", "Pantallas"));

    }
}
```

```
listaExhibicion.add(new Producto(4, "Cable HDMI", "Oferta"));

for (Producto p : listaExhibicion) {
    System.out.println(p);
}

// ===== B. LinkedList =====
System.out.println("\n== LinkedList: Reabastecer ==");

LinkedList<Producto> colaReabastecer = new LinkedList<>();
colaReabastecer.addLast(new Producto(5, "Laptop", "Computo"));
colaReabastecer.addFirst(new Producto(6, "Webcam", "Perifericos"));

for (Producto p : colaReabastecer) {
    System.out.println(p);
}

// ===== C. HashSet =====
System.out.println("\n== HashSet: Categorías únicas ==");

HashSet<String> categorias = new HashSet<>();
for (Producto p : listaExhibicion) {
    categorias.add(p.getCategoría());
}
for (Producto p : colaReabastecer) {
    categorias.add(p.getCategoría());
}
```

```

System.out.println(categorias);

// ===== D. HashMap =====
System.out.println("\n== HashMap: Consulta por id ==");

HashMap<Integer, Producto> mapaPorId = new HashMap<>();
for (Producto p : listaExhibicion) {
    mapaPorId.put(p.getId(), p);
}
for (Producto p : colaReabastecer) {
    mapaPorId.put(p.getId(), p);
}

int idConsulta = 3;
System.out.println("id=" + idConsulta + " -> " + mapaPorId.get(idConsulta));

System.out.println("\nRecorriendo HashMap (entrySet):");
for (Map.Entry<Integer, Producto> entry : mapaPorId.entrySet()) {
    System.out.println(entry.getKey() + " -> " + entry.getValue());
}

// ===== E. Iterator =====
System.out.println("\n== Iterator: eliminar categoría 'Oferta' en listaExhibicion ==");

Iterator<Producto> it = listaExhibicion.iterator();
while (it.hasNext()) {
    Producto p = it.next();
    if (p.getCategoría().equals("Oferta")) {

```

```

        it.remove(); // eliminación segura
    }
}

System.out.println("Después de eliminar:");
for (Producto p : listaExhibicion) {
    System.out.println(p);
}
}
}
}

```

- Operable

```

package practica8;

/**
 *
 * @author garza
 */

public class Producto {

    private int id;
    private String nombre;
    private String categoria;

    public Producto(int id, String nombre, String categoria) {
        this.id = id;
        this.nombre = nombre;
        this.categoria = categoria;
    }
}

```

```
public int getId() {
    return id;
}

public String getNombre() {
    return nombre;
}

public String getCategoría() {
    return categoría;
}

@Override
public String toString() {
    return "Producto{id=" + id + ", nombre='" + nombre + "', categoría='" + categoría +
"}";
}

// equals y hashCode basados solo en id
@Override
public boolean equals(Object obj) {
    if (this == obj) return true;
    if (!(obj instanceof Producto)) return false;
    Producto otro = (Producto) obj;
    return this.id == otro.id;
}

@Override
public int hashCode() {
```

```
    return Integer.hashCode(id);  
}  
}
```