

# **Actividad**

## **UF 1-4. Hilos**

**Ivan Nuñez Rodriguez**

2º DAM

Programación de Procesos y Servicios

## Índice

1. Código.
2. Resultado.
3. Enlace de Github

## Código

### Clase CLienteBiblioteca

```
import java.io.*;
import java.net.Socket;
import java.util.Scanner;

public class ClienteBiblioteca {
    private static final String HOST = "localhost";
    private static final int PUERTO = 12345;

    public static void main(String[] args) {
        try (Socket cliente = new Socket(HOST, PUERTO);
            ObjectOutputStream salida = new
ObjectOutputStream(cliente.getOutputStream());
            ObjectInputStream entrada = new
ObjectInputStream(cliente.getInputStream());
            Scanner scanner = new Scanner(System.in)) {

            String opcion;
            do {
                System.out.println("\n--- Menú de la Biblioteca ---");
                System.out.println("1. Consultar libro por ISBN");
                System.out.println("2. Consultar libro por título");
                System.out.println("3. Consultar libros por autor");
                System.out.println("4. Añadir libro");
                System.out.println("5. Salir de la aplicación");
                System.out.print("Seleccione una opción: ");
                opcion = scanner.nextLine();
                salida.writeObject(opcion);

                switch (opcion) {
                    case "1":
                        System.out.print("Introduzca el ISBN: ");
                        String isbn = scanner.nextLine();
                        salida.writeObject(isbn);
                        System.out.println("Respuesta: " +
entrada.readObject());
                        break;
                    case "2":
                        System.out.print("Introduzca el título: ");
                        String titulo = scanner.nextLine();
                        salida.writeObject(titulo);
                        System.out.println("Respuesta: " +
entrada.readObject());
                        break;
                    case "3":
                        System.out.print("Introduzca el autor: ");
                        String autor = scanner.nextLine();
                        salida.writeObject(autor);
                        System.out.println("Respuesta: " +
entrada.readObject());
                        break;
                    case "4":
                        System.out.print("ISBN: ");
                        String nuevoISBN = scanner.nextLine();
                        System.out.print("Título: ");
                        String nuevoTitulo = scanner.nextLine();
                        System.out.print("Autor: ");
                        String nuevoAutor = scanner.nextLine();
```

```

        salida.writeObject(new Libro(nuevoISBN,
nuevoTitulo, nuevoAutor));
        System.out.println("Respuesta: " +
entrada.readObject());
        break;
    case "5":
        System.out.println("Cerrando la sesión...");
        System.out.println("Respuesta del servidor: "
+ entrada.readObject());
        break;
    default:
        System.out.println("Opción no válida.");
        break;
    }
    } while (!opcion.equals("5"));

    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }
}
}

```

## clase Libro

```

import lombok.*;

import java.io.Serializable;
import java.util.List;
import java.util.stream.Collectors;
@Getter
@Setter
@ToString
@NoArgsConstructor
@AllArgsConstructor
public class Libro implements Serializable {
    private String isbn;
    private String titulo;
    private String autor;

    // Métodos de consulta
    public static Libro consultarPorISBN(List<Libro> libros, String
isbn) {
        return libros.stream().filter(libro ->
libro.getISBN().equals(isbn)).findFirst().orElse(null);
    }

    public static Libro consultarPorTitulo(List<Libro> libros, String
titulo) {
        return libros.stream().filter(libro ->
libro.getTitulo().equalsIgnoreCase(titulo)).findFirst().orElse(null);
    }

    public static List<Libro> consultarPorAutor(List<Libro> libros,
String autor) {
        return libros.stream().filter(libro ->
libro.getAutor().equalsIgnoreCase(autor)).collect(Collectors.toList());
    }
}

```

## clase SevidorBiblioteca

```
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class ServidorBiblioteca {
    private static final int PUERTO = 12345;
    private static List<Libro> libros = new ArrayList<>();
    private static volatile boolean enEjecucion = true;

    public static void main(String[] args) {
        inicializarLibros();
        ExecutorService pool = Executors.newFixedThreadPool(10);

        try (ServerSocket servidor = new ServerSocket(PUERTO)) {
            System.out.println("Servidor iniciado en el puerto " +
                PUERTO);

            while (enEjecucion) {
                Socket cliente = servidor.accept();
                pool.execute(() -> gestionarCliente(cliente));
            }
            pool.shutdownNow();
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println("Servidor cerrado.");
    }

    private static void inicializarLibros() {
        libros.add(new Libro("1234567890", "El Principito", "Antoine
de Saint-Exupéry"));
        libros.add(new Libro("0987654321", "1984", "George Orwell"));
        libros.add(new Libro("5432167890", "Fahrenheit 451", "Ray
Bradbury"));
        libros.add(new Libro("6789054321", "La Odisea", "Homero"));
        libros.add(new Libro("1029384756", "Moby Dick", "Herman
Melville"));
    }

    private static void gestionarCliente(Socket cliente) {
        try (ObjectInputStream entrada = new
ObjectInputStream(cliente.getInputStream());
            ObjectOutputStream salida = new
ObjectOutputStream(cliente.getOutputStream())) {

            String opcion;
            while ((opcion = (String) entrada.readObject()) != null) {
                switch (opcion) {
                    case "1":
                        String isbn = (String) entrada.readObject();
```

```

salida.writeObject(Libro.consultarPorISBN(libros, isbn));
    break;
    case "2":
        String titulo = (String) entrada.readObject();
salida.writeObject(Libro.consultarPorTitulo(libros, titulo));
        break;
    case "3":
        String autor = (String) entrada.readObject();
salida.writeObject(Libro.consultarPorAutor(libros, autor));
        break;
    case "4":
        Libro nuevoLibro = (Libro)
entrada.readObject();
        synchronized (libros) {
            libros.add(nuevoLibro);
        }
        salida.writeObject("Libro añadido con
éxito.");
        break;
    case "5":
        System.out.println("Cliente " +
cliente.getInetAddress() + " ha cerrado la sesión.");
        salida.writeObject("Sesión cerrada
correctamente.");

        return;
    default:
        salida.writeObject("Opción no válida.");
        break;
    }
}
} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}
}
}

```

## Ejecución

### Ejecución ServidorBiblioteca

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea  
Servidor iniciado en el puerto 12345
```

### Ejecución ClienteBiblioteca

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea  
  
--- Menú de la Biblioteca ---  
1. Consultar libro por ISBN  
2. Consultar libro por título  
3. Consultar libros por autor  
4. Añadir libro  
5. Salir de la aplicación  
Seleccione una opción:
```

### Ejecución opción 1

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ I  
  
--- Menú de la Biblioteca ---  
1. Consultar libro por ISBN  
2. Consultar libro por título  
3. Consultar libros por autor  
4. Añadir libro  
5. Salir de la aplicación  
Seleccione una opción: 1  
Introduzca el ISBN: 1234567890  
Respuesta: Libro(ISBN=1234567890, titulo=El Principito, autor=Antoine de Saint-Exupéry)
```

## Ejecución opción 2

```
--- Menú de la Biblioteca ---
1. Consultar libro por ISBN
2. Consultar libro por título
3. Consultar libros por autor
4. Añadir libro
5. Salir de la aplicación
Seleccione una opción: 2
Introduzca el título: 1984
Respuesta: Libro(ISBN=0987654321, titulo=1984, autor=George Orwell)
```

## Ejecución opción 3

```
--- Menú de la Biblioteca ---
1. Consultar libro por ISBN
2. Consultar libro por título
3. Consultar libros por autor
4. Añadir libro
5. Salir de la aplicación
Seleccione una opción: 3
Introduzca el autor: homero
Respuesta: [Libro(ISBN=6789054321, titulo=La Odisea, autor=Homero)]
```

## Ejecución opción 4

```
--- Menú de la Biblioteca ---
1. Consultar libro por ISBN
2. Consultar libro por título
3. Consultar libros por autor
4. Añadir libro
5. Salir de la aplicación
Seleccione una opción: 4
ISBN: 47049754b
Título: ivan
Autor: nuñez
Respuesta: Libro añadido con éxito.
```

## Ejecución comprobación 4

```
--- Menú de la Biblioteca ---
1. Consultar libro por ISBN
2. Consultar libro por título
3. Consultar libros por autor
4. Añadir libro
5. Salir de la aplicación
Seleccione una opción: 3
Introduzca el autor: nuñez
Respuesta: [Libro(ISBN=47049754b, titulo=ivan, autor=nuñez)]
```



## Ejecucion opción 5

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea_p

--- Menú de la Biblioteca ---
1. Consultar libro por ISBN
2. Consultar libro por título
3. Consultar libros por autor
4. Añadir libro
5. Salir de la aplicación
Seleccione una opción: 5
Cerrando la sesión...
Respuesta del servidor: Sesión cerrada correctamente.

Process finished with exit code 0
|
```

## Ejecución finalización programa parte servidor

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.4\lib\idea
Servidor iniciado en el puerto 12345
Cliente /127.0.0.1 ha cerrado la sesión.
```

## Enalce a Github

[https://github.com/lvannunezrodriguez/Programacion\\_de\\_Servicios\\_y\\_Procesos\\_24-25/tree/85f18768662cc240e4f710689fb74d1859882a38/Actividades/tema\\_1/Actividad%201](https://github.com/lvannunezrodriguez/Programacion_de_Servicios_y_Procesos_24-25/tree/85f18768662cc240e4f710689fb74d1859882a38/Actividades/tema_1/Actividad%201)