TAREA TEMA 12 PRO

09/03/2023

Autor: Derimán Tejera Fumero

Actividades de comprobación

12.1. ¿Qué es Collection?
a) Una interfaz.
b) Una clase.
c) Un sistema operativo.
d) Un método.

12.2. Los tipos genéricos sirven para:
a) Usar objetos de la clase Object.
b) Usar variables primitivas.
c) Usar tipos parametrizados.
d) No tener que usar ningún tipo.

12.3. ¿Para qué sirve una lista?
a) Guardar datos primitivos.
b) Guardar datos que no se pueden repetir.
c) No tener que ordenar ixi conjunto de datos.
d) Guardar, de forma dinámica, datos que se pueden repetir y ordenar.

12.4. Un conjunto es una colección de elementos:
a) Que no admiten orden.
b) Que admiten repeticiones.
c) Que no se pueden alterar.
d) Cuyo criterio fundamental es el de pertenecer al conjunto.

12.5. ArrayList y LinkedList se diferencian:
a) En el número de elementos.
b) En el rendimiento.
c) En el orden de los elementos.
d) En nada.

12.6. Los métodos de la interfaz Set:
a) Son los mismos que los de List.
b) Son los mismos que los de Collection.
c) Son implementados en la clase ArrayList.
d) Esta interfaz no tiene métodos.

12.7. Si la variable a referencia un objeto ArrayList, la expresión new TreeSet(a):
a) Devuelve un conjunto ordenado con los elementos de a.
b) Es incorrecta.
c) Devuelve una lista ordenada.
d) Devuelve una tabla.

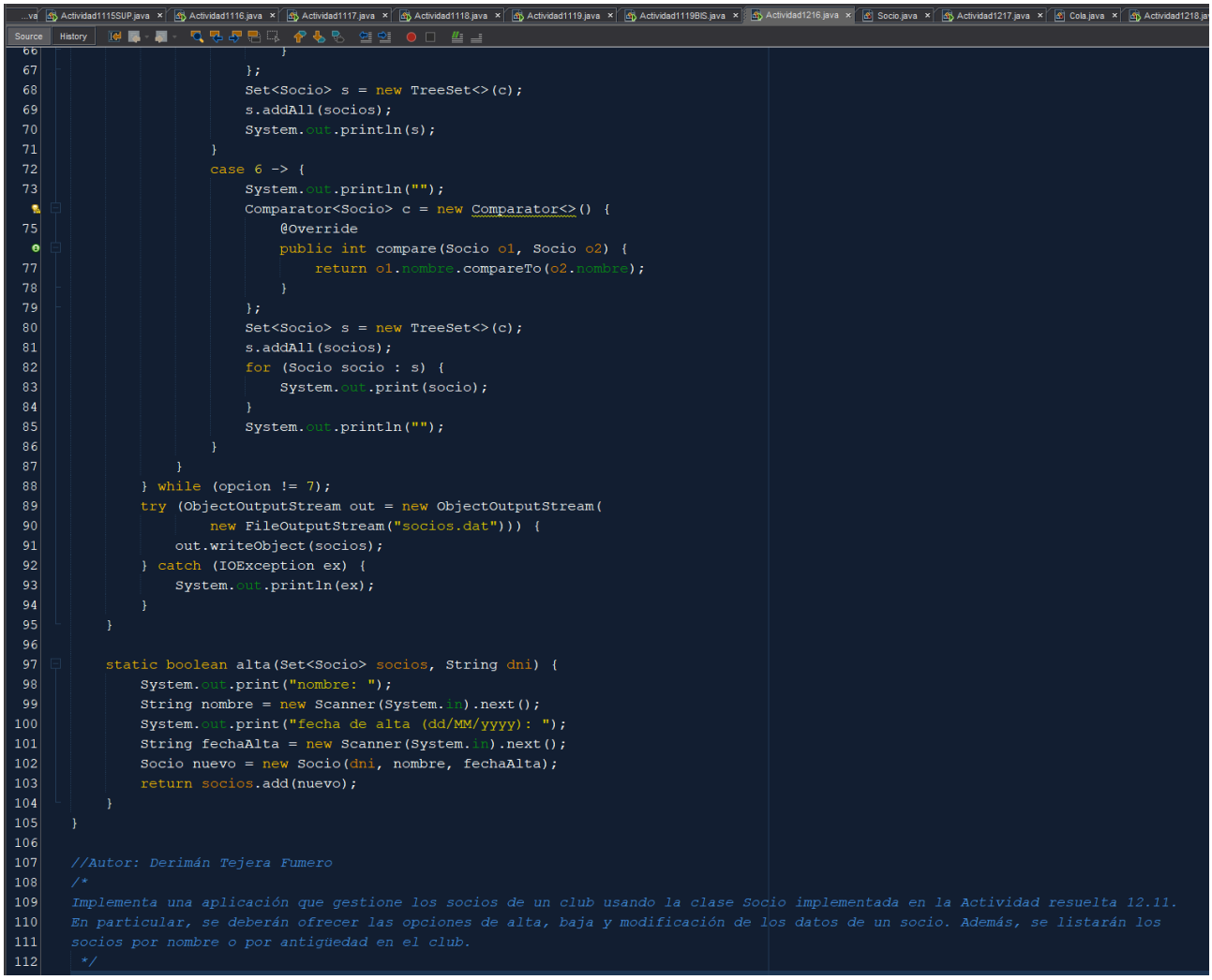
12.8. ¿Qué es Collections?
a) Una dase cuyos objetos están repetidos.
b) Una interfaz de la que heredan todas las colecciones.
c) Una clase con métodos estáticos que sirven para gestionar colecciones.
d) Nada, le sobra la ese.

12.9. Un mapa en Java es:
a) Un gráfico con las relaciones de herencia entre interfaces.
b) Una colección.
c) Una representación de los datos por pantalla.
d) Una estructura dinámica cuyos elementos son parejas clave-valor.

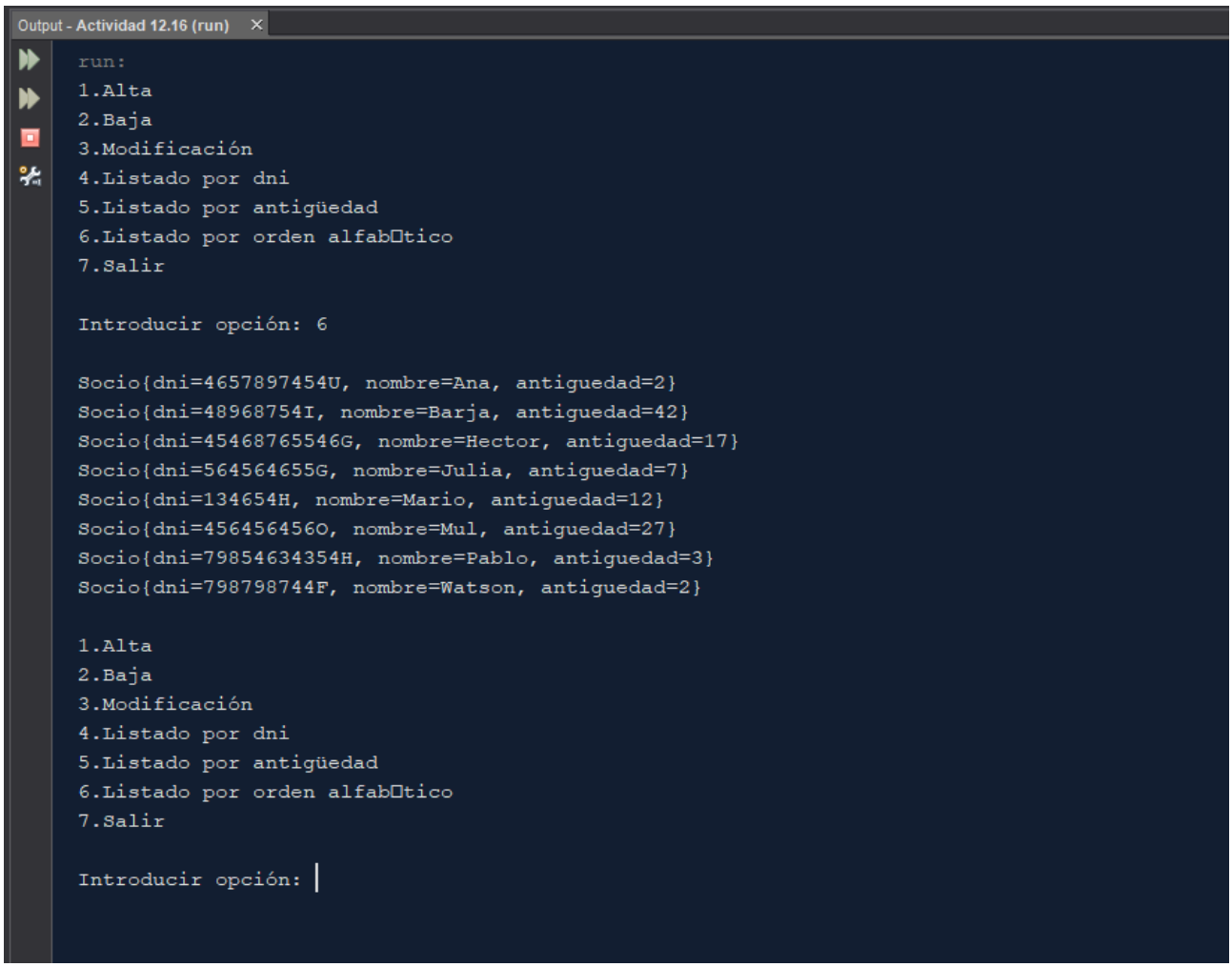
12.10. Si queremos cambiar el valor de una entrada en un mapa, usaremos el método:
a) put().
b) set().
c) add().
d) insert().

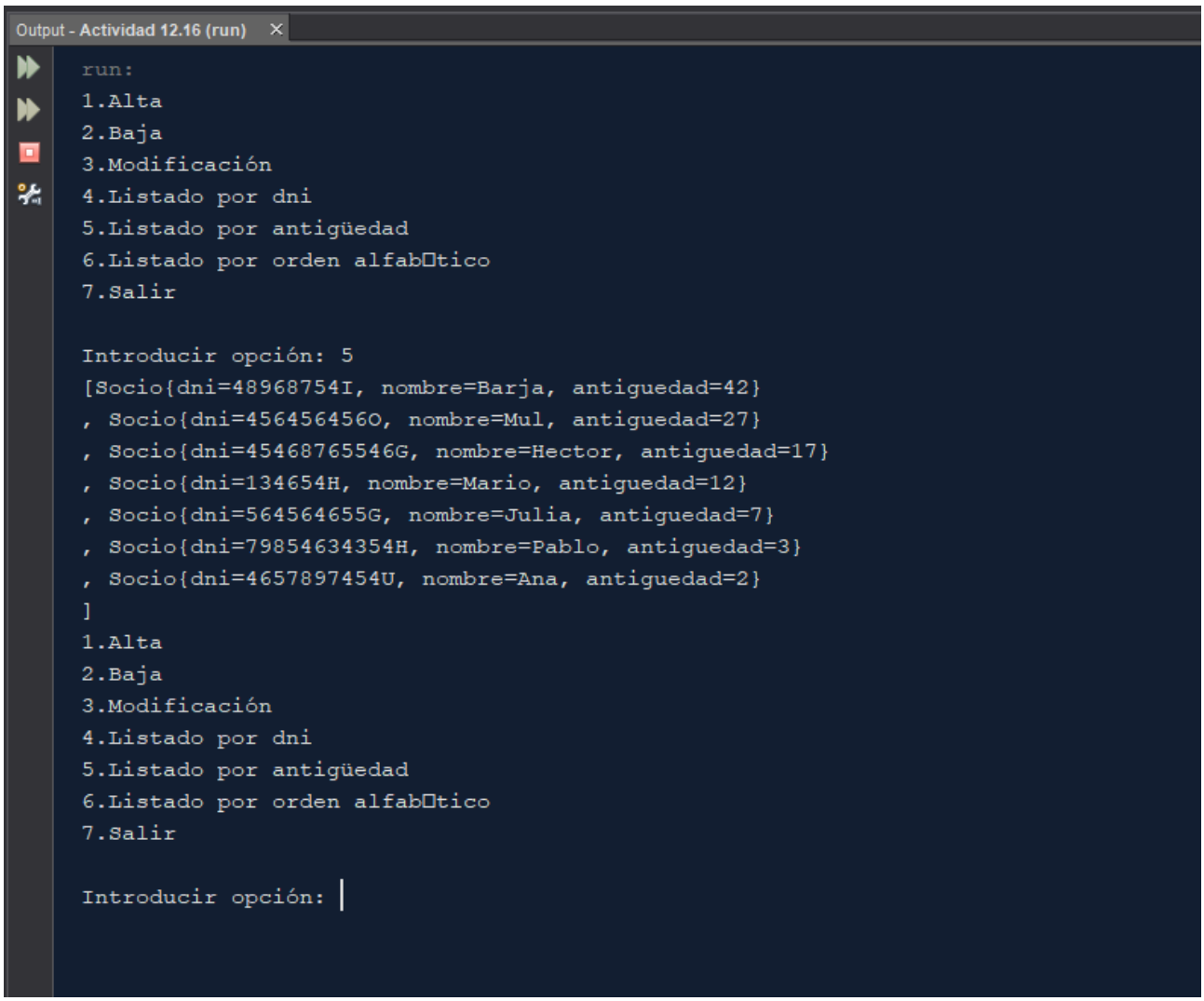
Actividades de aplicación

12.16. Implemento una aplicación que gestione los socios de un club usando la clase Socio implementada en la Actividad resuelta 12.11. En particular, se deberán ofrecer las opciones de alta, baja y modificación de los datos de un socio. Además, se listarán los socios por nombre o por antigüedad en el club.

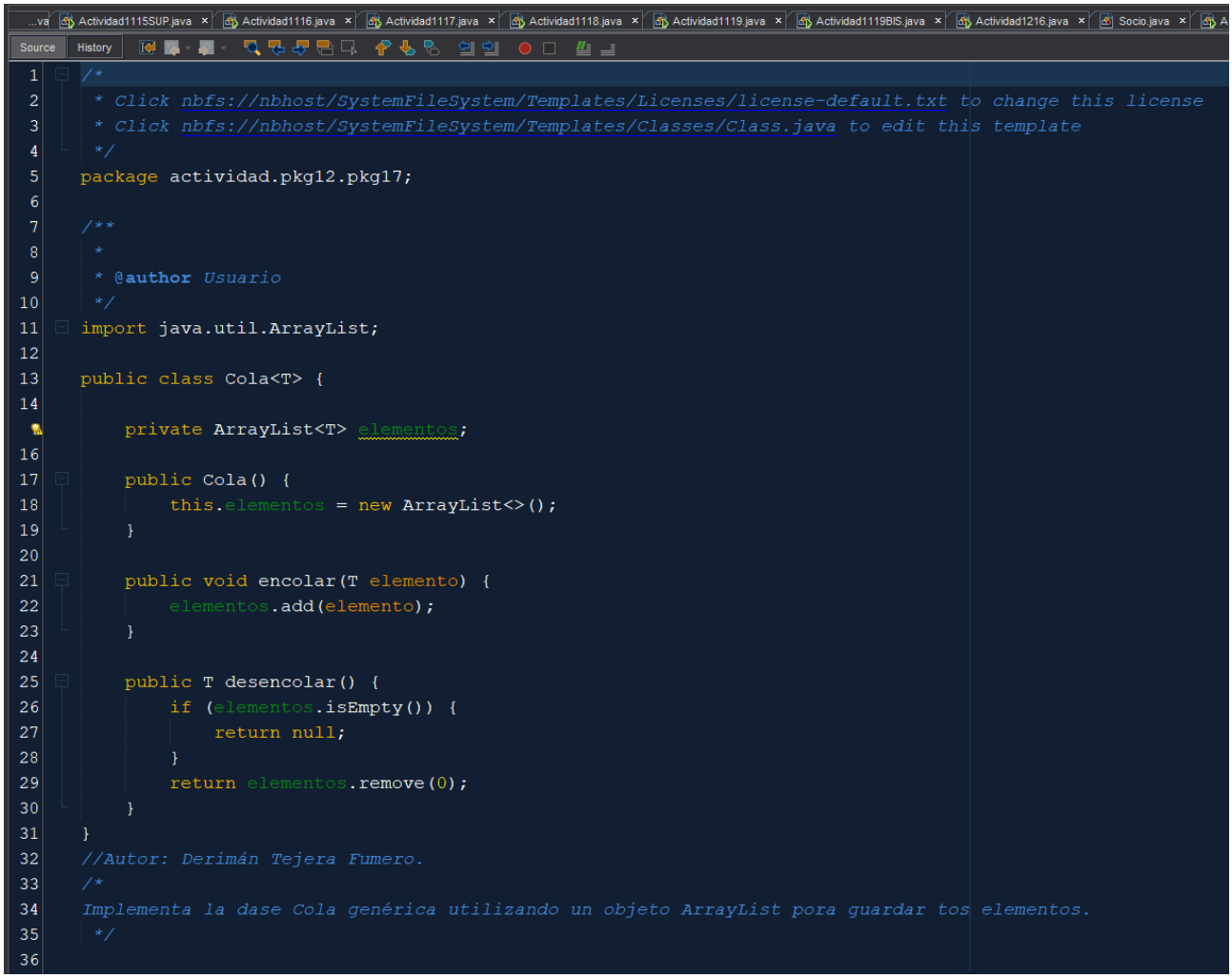



```
                    }
                    };
                    Set<Socio> s = new TreeSet<>(c);
                    s.addAll(socios);
                    System.out.println(s);
                }
                case 6 -> {
                    System.out.println("");
                    Comparator<Socio> c = new Comparator<>() {
                        @Override
                        public int compare(Socio o1, Socio o2) {
                            return o1.nombre.compareTo(o2.nombre);
                        }
                    };
                    Set<Socio> s = new TreeSet<>(c);
                    s.addAll(socios);
                    for (Socio socio : s) {
                        System.out.print(socio);
                    }
                    System.out.println("");
                }
            }
        } while (opcion != 7);
        try (ObjectOutputStream out = new ObjectOutputStream(
                new FileOutputStream("socios.dat"))) {
            out.writeObject(socios);
        } catch (IOException ex) {
            System.out.println(ex);
        }
    }

    static boolean alta(Set<Socio> socios, String dni) {
        System.out.print("nombre: ");
        String nombre = new Scanner(System.in).next();
        System.out.print("fecha de alta (dd/MM/yyyy): ");
        String fechaAlta = new Scanner(System.in).next();
        Socio nuevo = new Socio(dni, nombre, fechaAlta);
        return socios.add(nuevo);
    }
}

//Autor: Derimán Tejera Fumero
/*
Implementa una aplicación que gestione los socios de un club usando la clase Socio implementada en la Actividad resuelta 12.11.
En particular, se deberán ofrecer las opciones de alta, baja y modificación de los datos de un socio. Además, se listarán los
socios por nombre o por antigüedad en el club.
 */
```

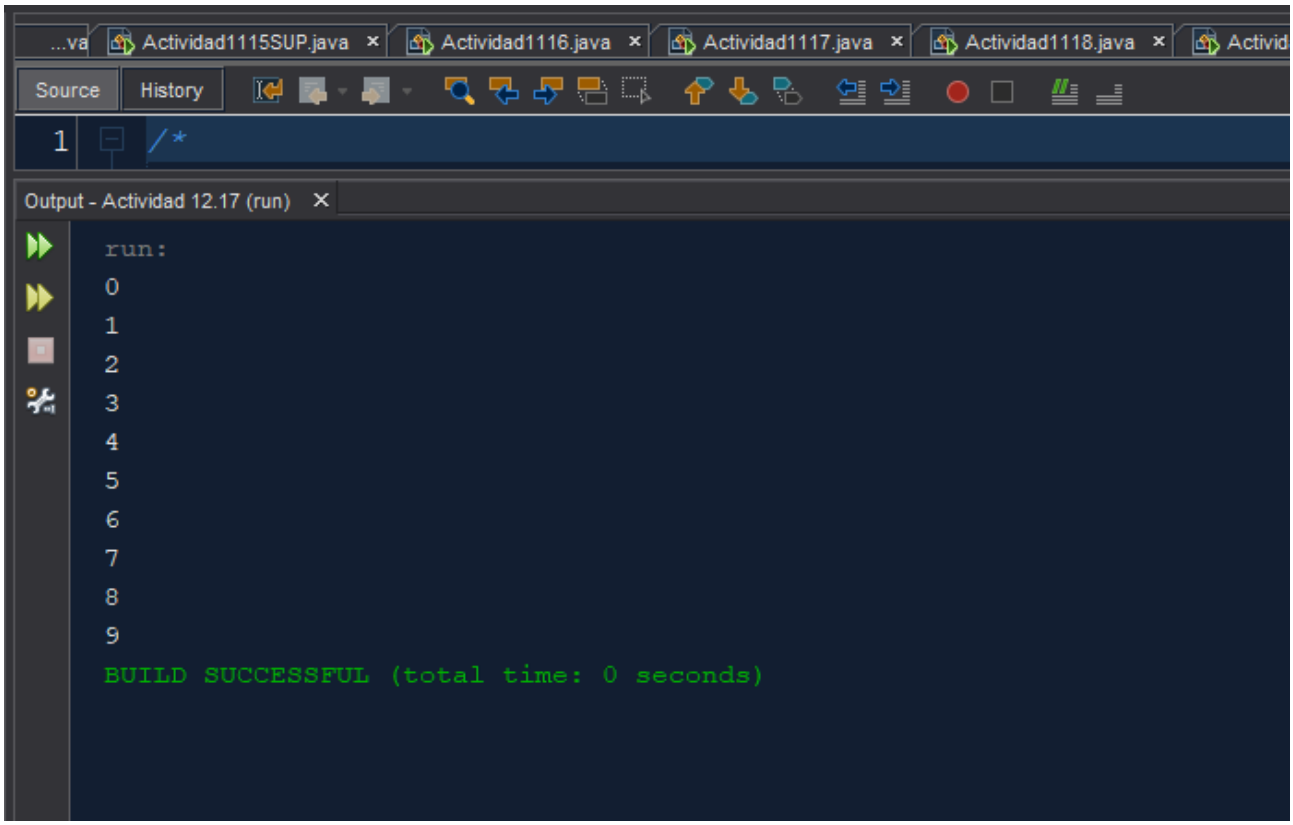

Output - Actividad 12.16 (run)
run:
1.Alta
2.Baja
3.Modificación
4.Listado por dni
5.Listado por antigüedad
6.Listado por orden alfab□tico
7.Salir
Introducir opción: 6
Socio{dni=4657897454U, nombre=Ana, antiguedad=2}
Socio{dni=48968754I, nombre=Barja, antiguedad=42}
Socio{dni=45468765546G, nombre=Hector, antiguedad=17}
Socio{dni=564564655G, nombre=Julia, antiguedad=7}
Socio{dni=134654H, nombre=Mario, antiguedad=12}
Socio{dni=4564564560, nombre=Mul, antiguedad=27}
Socio{dni=79854634354H, nombre=Pablo, antiguedad=3}
Socio{dni=798798744F, nombre=Watson, antiguedad=2}
1.Alta
2.Baja
3.Modificación
4.Listado por dni
5.Listado por antigüedad
6.Listado por orden alfab□tico
7.Salir
Introducir opción:



12.17. Implementa la clase Cola genérica utilizando un objeto ArrayList pora guardar tos elementos.

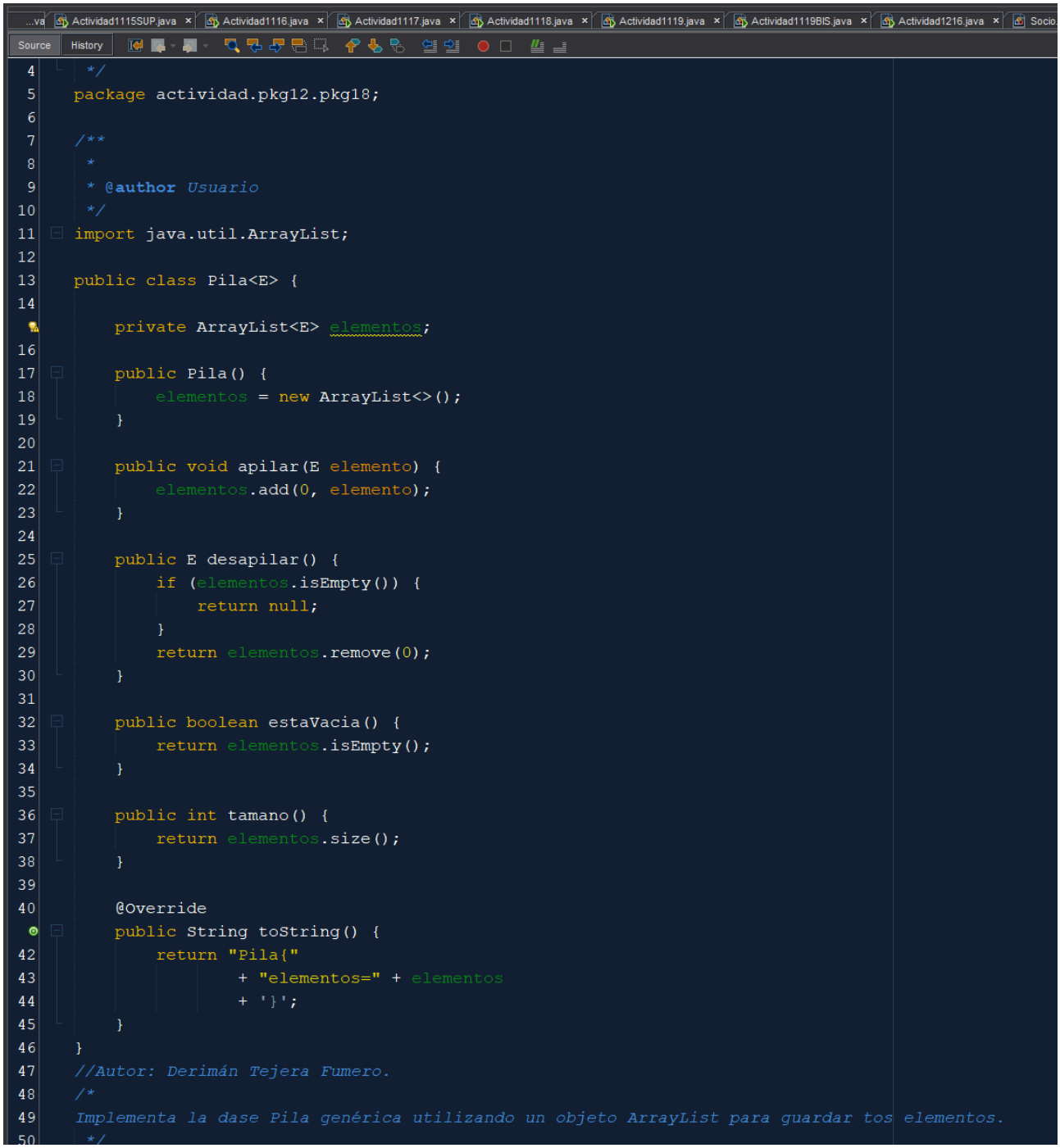


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package actividad.pkg12.pkg17;

/**
 *
 * @author Usuario
 */
import java.util.ArrayList;

public class Cola<T> {

    private ArrayList<T> elementos;

    public Cola() {
        this.elementos = new ArrayList<>();
    }

    public void encolar(T elemento) {
        elementos.add(elemento);
    }

    public T desencolar() {
        if (elementos.isEmpty()) {
            return null;
        }
        return elementos.remove(0);
    }
}
//Autor: Derimán Tejera Fumero.
/*
Implementa la dase Cola genérica utilizando un objeto ArrayList pora guardar tos elementos.
 */

```

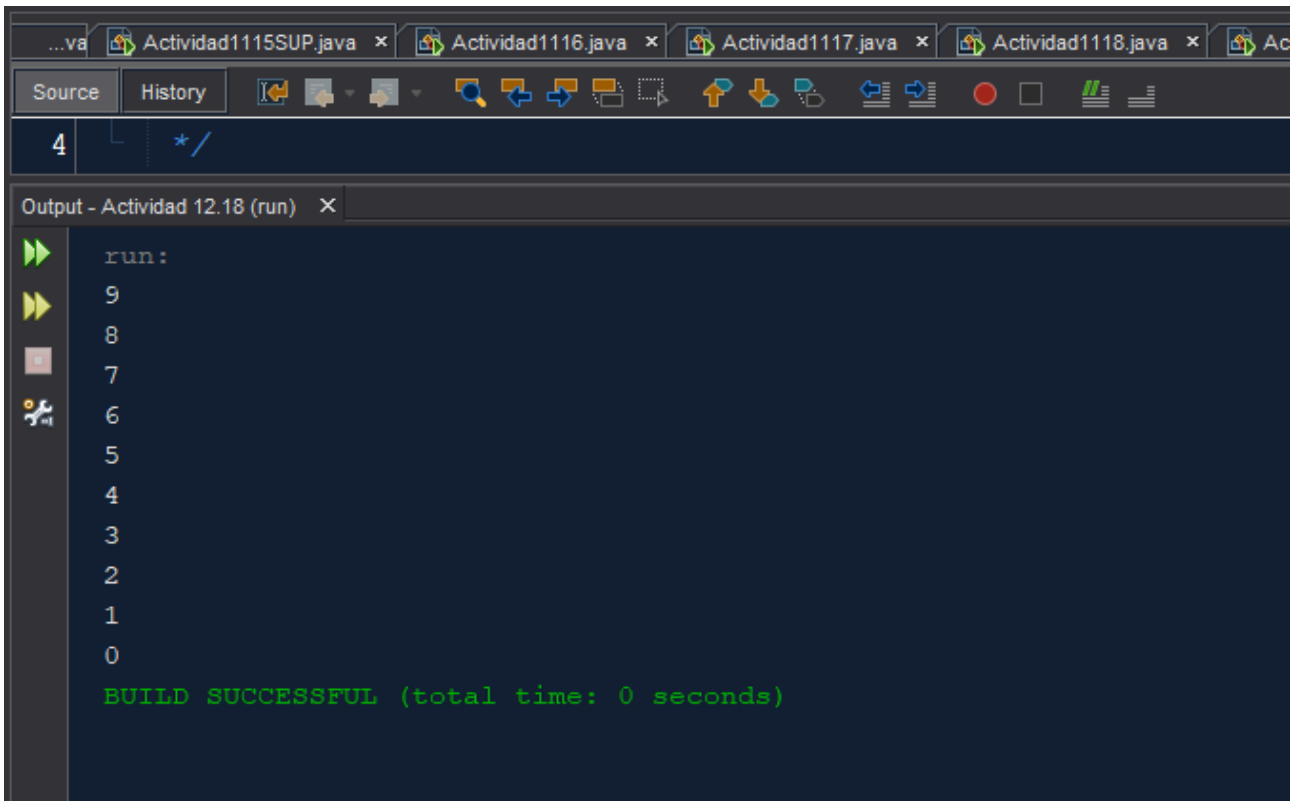


```
run:
0
1
2
3
4
5
6
7
8
9
BUILD SUCCESSFUL (total time: 0 seconds)
```

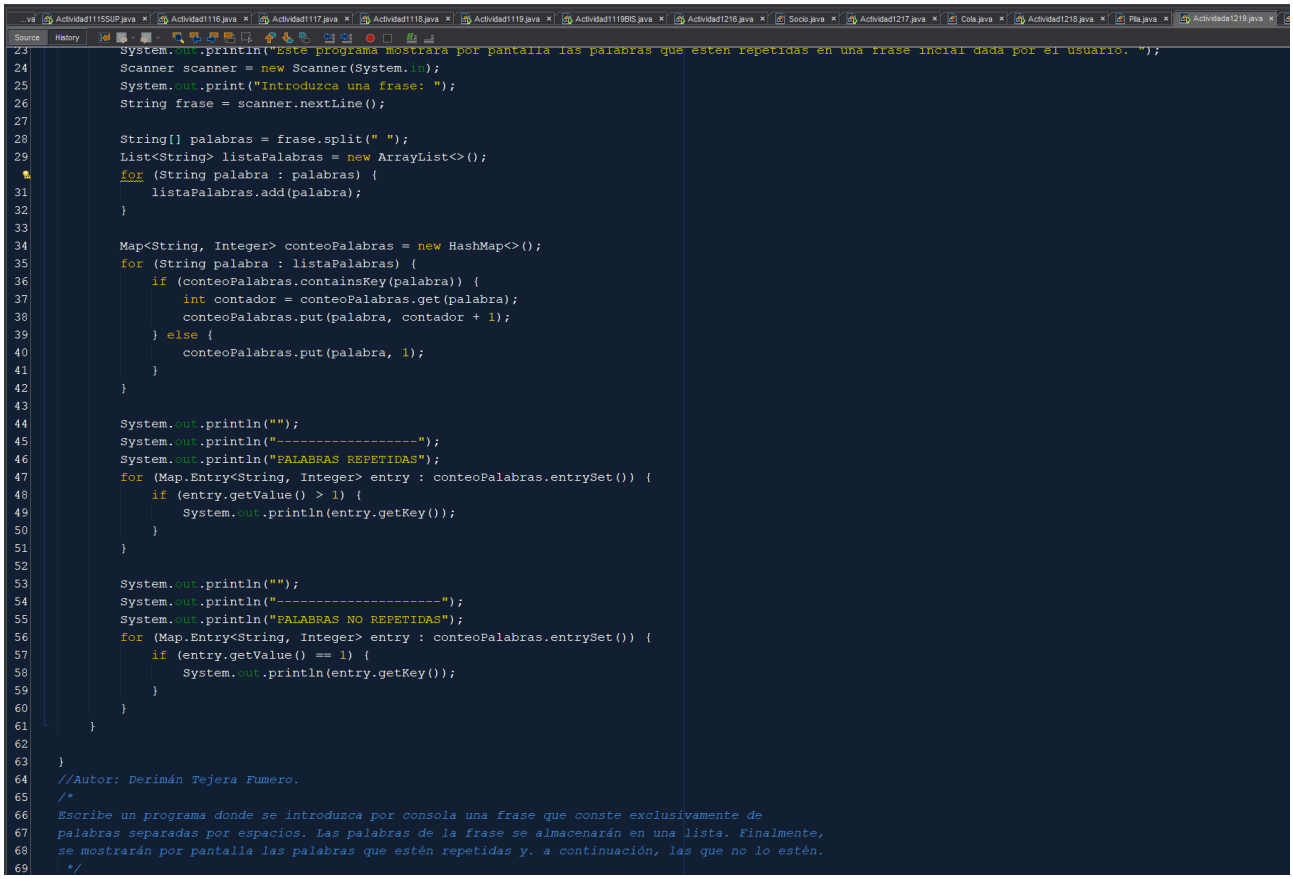
12.18. Implemento la clase Pila genérica utilizando un objeto ArrayList para guardar tos elementos.



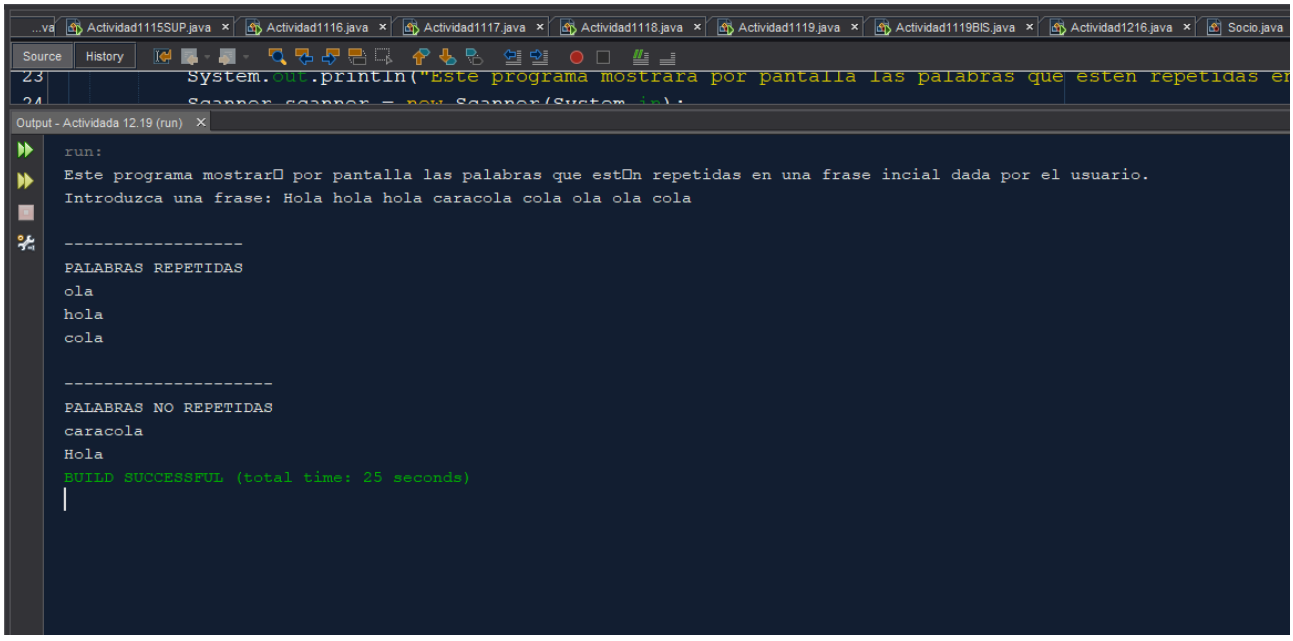
```java
 */
package actividad.pkg12.pkg18;

/**
 *
 * @author Usuario
 */
import java.util.ArrayList;

public class Pila<E> {

    private ArrayList<E> elementos;

    public Pila() {
        elementos = new ArrayList<>();
    }

    public void apilar(E elemento) {
        elementos.add(0, elemento);
    }

    public E desapilar() {
        if (elementos.isEmpty()) {
            return null;
        }
        return elementos.remove(0);
    }

    public boolean estaVacia() {
        return elementos.isEmpty();
    }

    public int tamano() {
        return elementos.size();
    }

    @Override
    public String toString() {
        return "Pila{"
                + "elementos=" + elementos
                + '}';
    }
}
//Autor: Derimán Tejera Fumero.
/*
Implementa la dase Pila genérica utilizando un objeto ArrayList para guardar tos elementos.
 */
```



12.19. Escribe un programa donde se introduzca por consola una frase que conste exclusivamente de palabras separadas por espacios. Las palabras de la frase se almacenarán en una lista. Finalmente, se mostrarán por pantalla las palabras que estén repetidas y. a continuación, las que no lo estén.

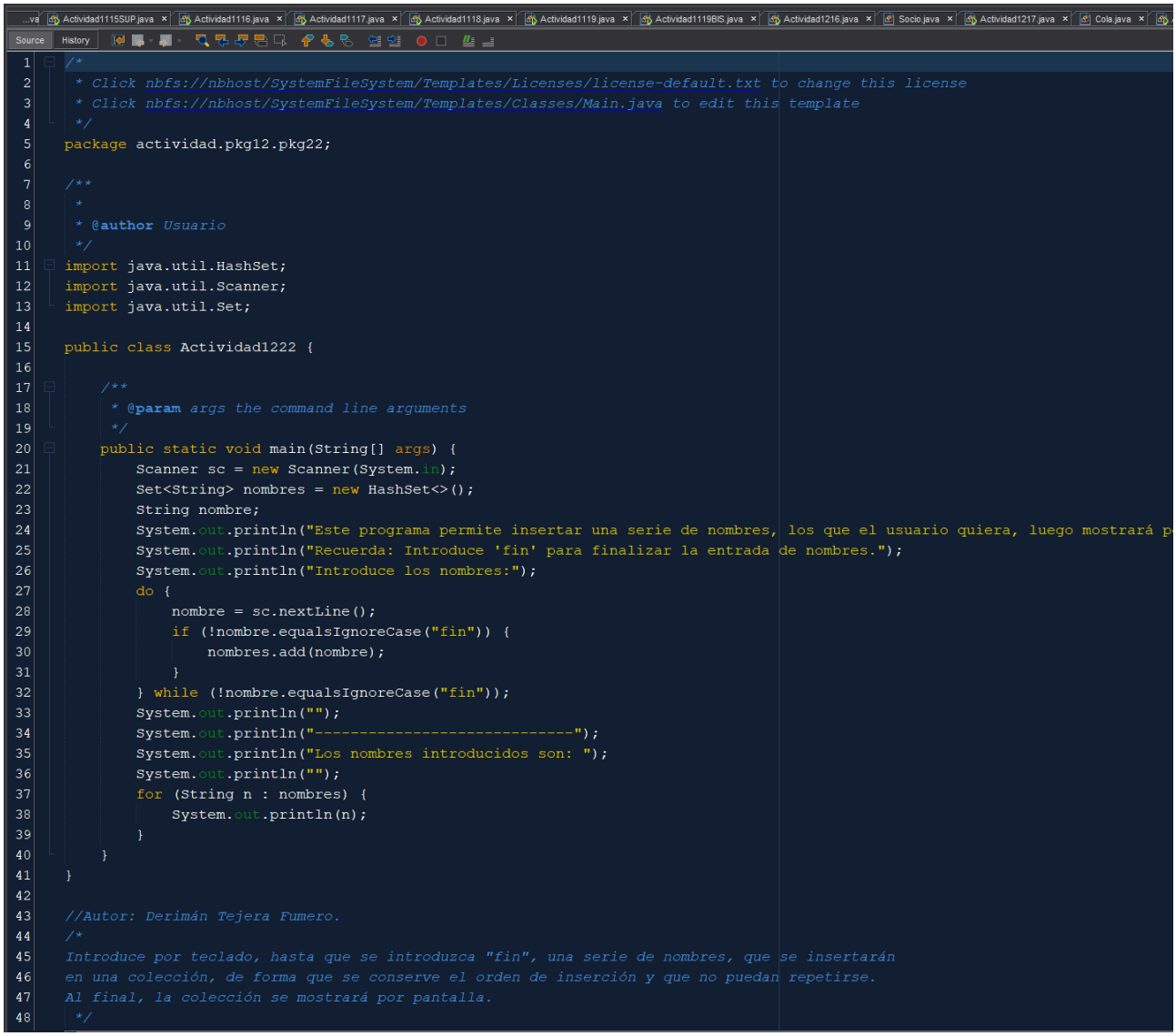


```java
        System.out.println("Este programa mostrara por pantalla las palabras que esten repetidas en una frase incial dada por el usuario. ");
        Scanner scanner = new Scanner(System.in);
        System.out.print("Introduzca una frase: ");
        String frase = scanner.nextLine();

        String[] palabras = frase.split(" ");
        List<String> listaPalabras = new ArrayList<>();
        for (String palabra : palabras) {
            listaPalabras.add(palabra);
        }

        Map<String, Integer> conteoPalabras = new HashMap<>();
        for (String palabra : listaPalabras) {
            if (conteoPalabras.containsKey(palabra)) {
                int contador = conteoPalabras.get(palabra);
                conteoPalabras.put(palabra, contador + 1);
            } else {
                conteoPalabras.put(palabra, 1);
            }
        }

        System.out.println("");
        System.out.println("------------------");
        System.out.println("PALABRAS REPETIDAS");
        for (Map.Entry<String, Integer> entry : conteoPalabras.entrySet()) {
            if (entry.getValue() > 1) {
                System.out.println(entry.getKey());
            }
        }

        System.out.println("");
        System.out.println("----------------------");
        System.out.println("PALABRAS NO REPETIDAS");
        for (Map.Entry<String, Integer> entry : conteoPalabras.entrySet()) {
            if (entry.getValue() == 1) {
                System.out.println(entry.getKey());
            }
        }
    }

}
//Autor: Derimán Tejera Fumero.
/*
Escribe un programa donde se introduzca por consola una frase que conste exclusivamente de
palabras separadas por espacios. Las palabras de la frase se almacenarán en una lista. Finalmente,
se mostrarán por pantalla las palabras que estén repetidas y. a continuación, las que no lo estén.
 */
```

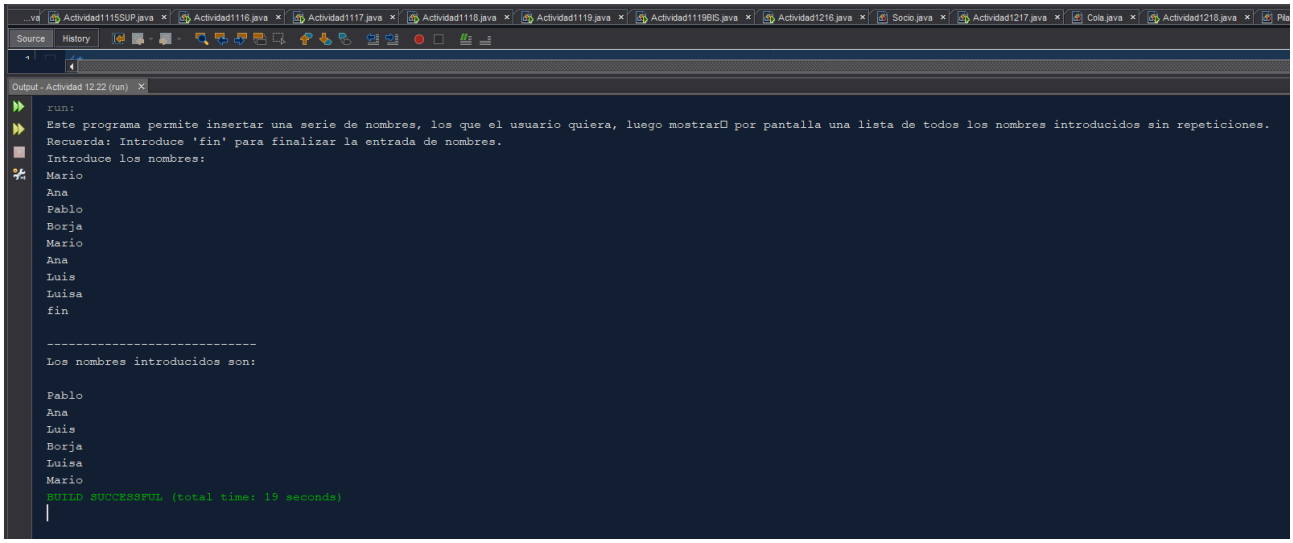


```
run:
Este programa mostrar por pantalla las palabras que est n repetidas en una frase incial dada por el usuario.
Introduzca una frase: Hola hola hola caracola cola ola ola cola

------------------
PALABRAS REPETIDAS
ola
hola
cola

----------------------
PALABRAS NO REPETIDAS
caracola
Hola
BUILD SUCCESSFUL (total time: 25 seconds)
```

12.22. Introduce por teclado, hasta que se introduzca "fin-, una serie de nombres, que se insertarán en una colección, de forma que se conserve el orden de inserción y que no puedan repetirse. Al final, la colección se mostrará por pantalla.

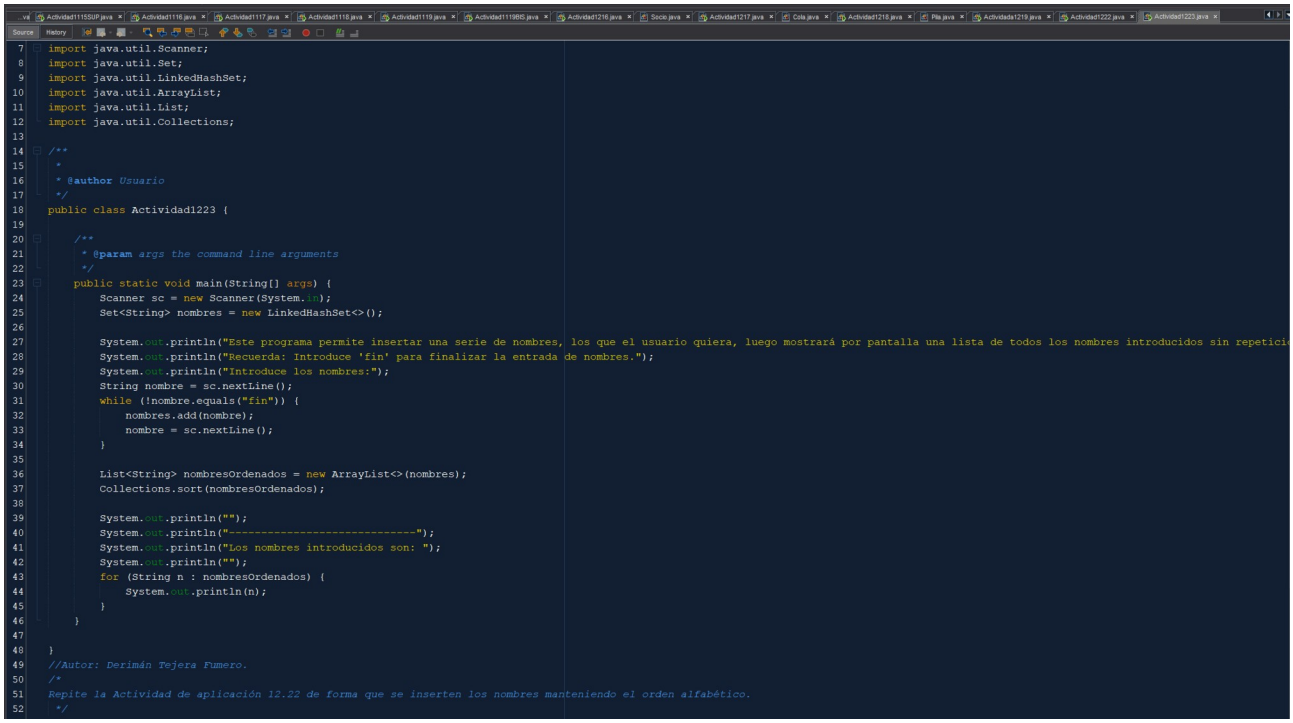


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package actividad.pkg12.pkg22;

/**
 *
 * @author Usuario
 */
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class Actividad1222 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Set<String> nombres = new HashSet<>();
        String nombre;
        System.out.println("Este programa permite insertar una serie de nombres, los que el usuario quiera, luego mostrará p
        System.out.println("Recuerda: Introduce 'fin' para finalizar la entrada de nombres.");
        System.out.println("Introduce los nombres:");
        do {
            nombre = sc.nextLine();
            if (!nombre.equalsIgnoreCase("fin")) {
                nombres.add(nombre);
            }
        } while (!nombre.equalsIgnoreCase("fin"));
        System.out.println("");
        System.out.println("-----------------------------");
        System.out.println("Los nombres introducidos son: ");
        System.out.println("");
        for (String n : nombres) {
            System.out.println(n);
        }
    }
}

//Autor: Derimán Tejera Fumero.
/*
Introduce por teclado, hasta que se introduzca "fin", una serie de nombres, que se insertarán
en una colección, de forma que se conserve el orden de inserción y que no puedan repetirse.
Al final, la colección se mostrará por pantalla.
 */
```

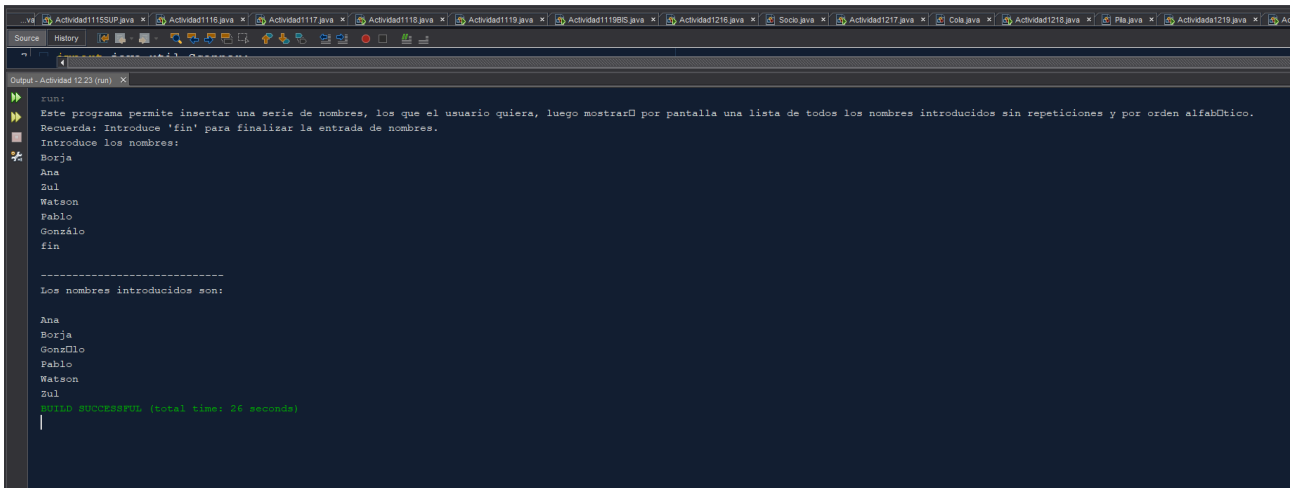


```
run:
Este programa permite insertar una serie de nombres, los que el usuario quiera, luego mostrar por pantalla una lista de todos los nombres introducidos sin repeticiones.
Recuerda: Introduce 'fin' para finalizar la entrada de nombres.
Introduce los nombres:
Mario
Ana
Pablo
Borja
Mario
Ana
Luis
Luisa
fin

-----------------------------
Los nombres introducidos son:

Pablo
Ana
Luis
Borja
Luisa
Mario
BUILD SUCCESSFUL (total time: 19 seconds)
```

12.23. Repite la Actividad de aplicación 12.22 de forma que se inserten los nombres manteniendo el orden alfabético.



```java
import java.util.Scanner;
import java.util.Set;
import java.util.LinkedHashSet;
import java.util.ArrayList;
import java.util.List;
import java.util.Collections;

/**
 *
 * @author Usuario
 */
public class Actividad1223 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Set<String> nombres = new LinkedHashSet<>();

        System.out.println("Este programa permite insertar una serie de nombres, los que el usuario quiera, luego mostrará por pantalla una lista de todos los nombres introducidos sin repeticio
        System.out.println("Recuerda: Introduce 'fin' para finalizar la entrada de nombres.");
        System.out.println("Introduce los nombres:");
        String nombre = sc.nextLine();
        while (!nombre.equals("fin")) {
            nombres.add(nombre);
            nombre = sc.nextLine();
        }

        List<String> nombresOrdenados = new ArrayList<>(nombres);
        Collections.sort(nombresOrdenados);

        System.out.println("");
        System.out.println("------------------------------");
        System.out.println("Los nombres introducidos son: ");
        System.out.println("");
        for (String n : nombresOrdenados) {
            System.out.println(n);
        }
    }

}
//Autor: Derimán Tejera Fumero.
/*
Repite la Actividad de aplicación 12.22 de forma que se inserten los nombres manteniendo el orden alfabético.
 */
```



```
run:
Este programa permite insertar una serie de nombres, los que el usuario quiera, luego mostrar□ por pantalla una lista de todos los nombres introducidos sin repeticiones y por orden alfab□tico.
Recuerda: Introduce 'fin' para finalizar la entrada de nombres.
Introduce los nombres:
Borja
Ana
Zul
Watson
Pablo
Gonzálo
fin

------------------------------
Los nombres introducidos son:

Ana
Borja
Gonz□lo
Pablo
Watson
Zul
BUILD SUCCESSFUL (total time: 26 seconds)
```