

# UNIDAD

## COLECCIONES

### EJERCICIOS

Autores:

Cristina Moreno Ruiz y chatGPT  
[cmorenoruiz@educa.madrid.org](mailto:cmorenoruiz@educa.madrid.org)

2023/2024

PROGRAMACIÓN  
CFGS DAW

#### Licencia



**Reconocimiento - NoComercial - CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## UD. COLECCIONES

### EJERCICIOS

#### HashMap:

1. Crea un programa que solicite al usuario que ingrese una lista de nombres y sus respectivos números de teléfono. Utiliza un HashMap para almacenar los nombres como claves y los números de teléfono como valores. Luego, permite al usuario buscar un nombre y mostrar el número de teléfono correspondiente.
2. Implementa un programa que cuente la frecuencia de cada palabra en una oración. Utiliza un HashMap para almacenar las palabras como claves y sus frecuencias como valores. Muestra el resultado al final.

#### HashSet:

3. Crea una clase "Persona" con atributos como nombre, edad y ciudad. Crea un conjunto HashSet de objetos de tipo Persona y agrega varias instancias a él. Luego, recorre el conjunto y muestra los nombres de las personas mayores de 30 años.
4. Implementa un programa que encuentre los elementos duplicados en un arreglo de enteros. Utiliza un HashSet para almacenar los elementos duplicados y muéstralos al final.

#### LinkedList:

5. Crea un programa que simule una lista de reproducción de música. Utiliza una LinkedList para almacenar las canciones en orden. Implementa funciones para agregar canciones al inicio y al final de la lista, reproducir la siguiente canción y mostrar la lista de reproducción completa.
6. Implementa una cola utilizando una LinkedList. Crea funciones para agregar elementos a la cola y para sacar elementos de la cola en orden. Por ejemplo, personas que esperan en la cola del cine.

#### Comparable:

7. Crea una clase "Producto" con atributos como nombre, precio y cantidad. Implementa la interfaz Comparable en la clase Producto para que los productos se puedan ordenar por precio de forma ascendente. Crea una lista de objetos Producto y ordénala utilizando el método sort() de Collections.
8. Crea una clase "Estudiante" con atributos como nombre, edad y promedio. Implementa la interfaz Comparable en la clase Estudiante para que los estudiantes se puedan ordenar por promedio de forma descendente. Crea una lista de objetos Estudiante y ordénala utilizando el método sort() de Collections.

**Comparator:**

9. Crea una clase "Libro" con atributos como título, autor y año de publicación. Implementa un comparador personalizado llamado "ComparadorLibros" que ordene los libros por año de publicación de forma descendente. Crea una lista de objetos Libro y ordénala utilizando el comparador personalizado.
10. Crea una clase "Pelicula" con atributos como título, director y duración. Implementa un comparador personalizado llamado "ComparadorPeliculas" que ordene las películas por duración de forma ascendente. Crea una lista de objetos Pelicula y ordénala utilizando el comparador personalizado.