

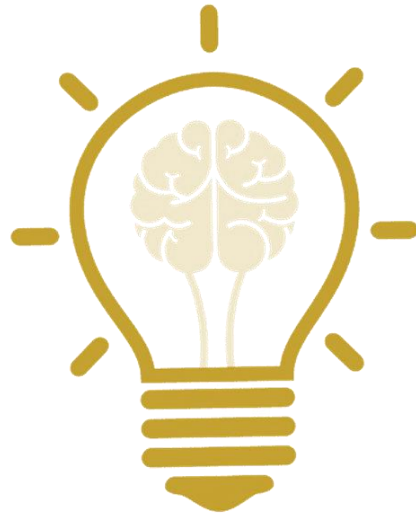
iTDS

**Instituto Técnico
Domingo Savio**
Profesionales en Educación

PROGRAMACIÓN EN JAVA

Programación Orientada a Objetos en JAVA

(Manejo de Excepciones en Java)



Objetivos de la sesión

- Comprender el framework de Colecciones en Java
- Aprender a utilizar las principales interfaces de colecciones: List, Set y Map
- Entender las diferencias entre ArrayList, LinkedList, HashSet y HashMap
- Practicar con ejercicios que involucren el uso de colecciones

Introducción a las Colecciones

- Las Colecciones son estructuras de datos que permiten almacenar y manipular grupos de objetos
- El framework de Colecciones en Java proporciona interfaces y clases para manejar colecciones de manera eficiente
- Principales interfaces: Collection, List, Set, Map



Interface List

- Representa una colección ordenada de elementos
- Permite elementos duplicados
- Principales implementaciones: ArrayList y LinkedList
- Ejemplo de HashSet;

```
List<String> nombres = new ArrayList<>();  
nombres.add("Ana");  
nombres.add("Carlos");  
nombres.add("Beatriz");  
System.out.println(nombres); // [Ana, Carlos, Beatriz]
```



Interface Map

- Representa una colección de pares clave-valor
- Las claves son únicas en el map
- Principales implementaciones: HashMap, LinkedHashMap, TreeMap
- Ejemplo de HashMap;

```
Map<String, Integer> edades = new HashMap<>();  
edades.put("Ana", 25);  
edades.put("Carlos", 30);  
edades.put("Beatriz", 28);  
System.out.println(edades.get("Carlos")); // 30
```



Iterando sobre Colecciones

- Usando for-each:

```
List<String> frutas = Arrays.asList("Manzana", "Banana", "Cereza");  
for (String fruta : frutas) {  
    System.out.println(fruta);  
}
```

- Usando un iterador:

```
List<String> frutas = Arrays.asList("Manzana", "Banana", "Cereza");  
for (String fruta : frutas) {  
    System.out.println(fruta);  
}
```

Métodos útiles de Collections

`Collections.sort(list)`: Ordena una lista

`Collections.reverse(list)`: Invierte el orden de una lista

`Collections.shuffle(list)`: Mezcla aleatoriamente los elementos de una lista

`Collections.max(collection)`: Encuentra el elemento máximo

`Collections.min(collection)`: Encuentra el elemento mínimo



Ejercicio práctico

Crear un sistema de gestión de una biblioteca:

1. Clase `Libro` con atributos título, autor y año de publicación
2. Usar un `ArrayList` para almacenar los libros de la biblioteca
3. Implementar métodos para añadir libros, buscar por autor, y listar todos los libros
4. Usar un `HashSet` para mantener una lista de autores únicos
5. Usar un `HashMap` para contar cuántos libros hay de cada autor



Tarea

1. Implementa un método para ordenar los libros por año de publicación
2. Añade funcionalidad para prestar y devolver libros, usando un `Map` para rastrear qué libros están prestados y a quién
3. Implementa un sistema de categorías para los libros, usando un `Map<String, List<Libro>>` para agrupar libros por categoría
4. Crea un método que genere un reporte de los libros más prestados, utilizando un `PriorityQueue`
5. Implementa la interfaz `Comparable` en la clase `Libro` para permitir ordenación natural



Recursos adicionales

- Documentación de Java sobre Colecciones:
docs.oracle.com/javase/tutorial/collections/
- Tutorial sobre Colecciones en Java:
- www.javatpoint.com/collections-in-java

Gracias!