Collections

(Java)

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Java)

Una colección es un objeto que sirve para almacenar objetos de manera Dinámica, donde cada elemento puede ser otro objeto.

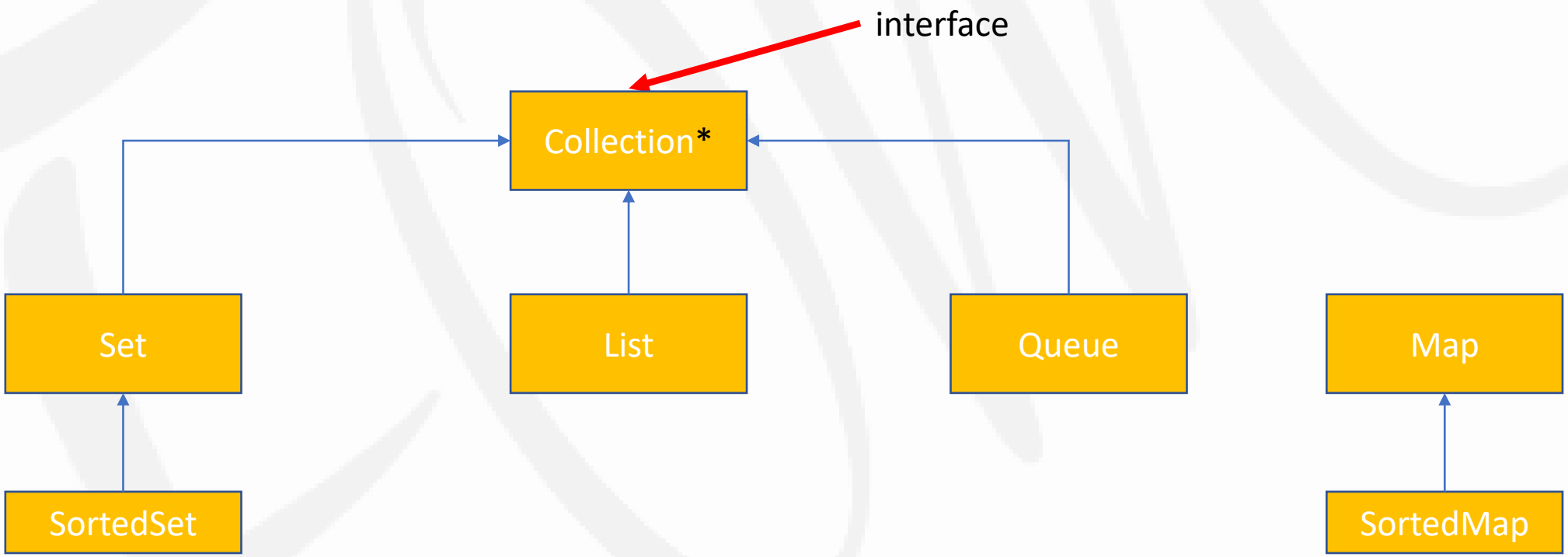
Se pueden almacenar:

- Objetos no repetidos o no
- Se pueden ordenar (según el tipo de colección)
- NO pueden almacenar datos primitivos
- Se puede insertar o eliminar objetos

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Java)



•java.util.Collections

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

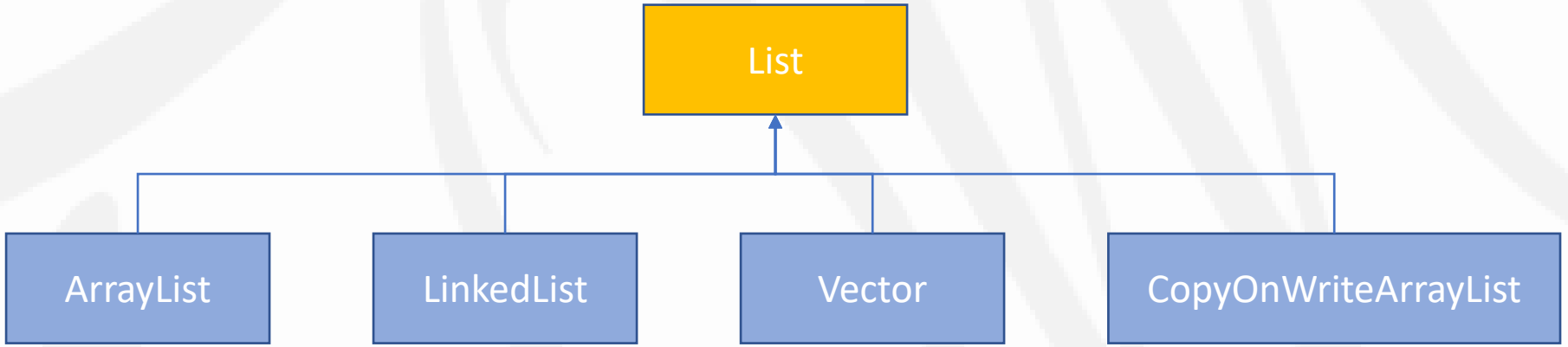
(Java)

Set	Almacena objetos no repetidos y no ordenados
SortedSet	Almacena objetos no repetidos y ordenados
List	Almacena objetos indexados, repetidos y ordenados
Queue	Almacena objetos uno seguido de otro y no permite acceso aleatorio (inicio o fin)
Map	Almacena objetos con clave única arbitraria
SortedMap	Almacena objetos con ordenada mediante clave

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Java)



Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Java)

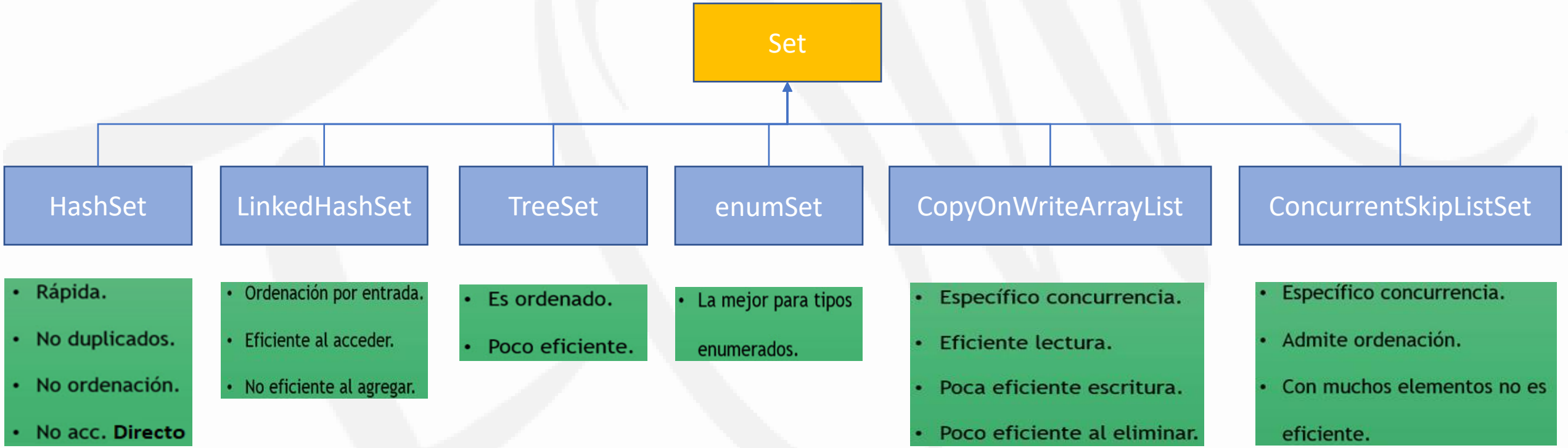
Las colecciones tienen problemas de desempeño

ArrayList	Se puede usar para muchos casos y es rápida para acceder a los datos
LinkedList	Enlaza listas. Se porta bien para agregar o eliminar, pero no para recorrer
Vector	Se considera obsoleta se usa para operaciones de concurrencia (Hilos - Threads)
CopyOnWriteArrayList	Se usa para operaciones de concurrencia. Eficiente para lectura, ineficiente para escritura

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

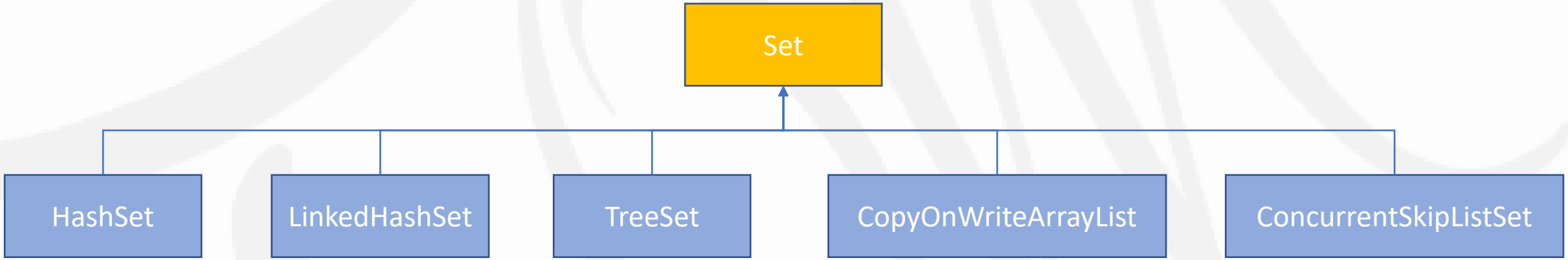
(Java)



Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Java)



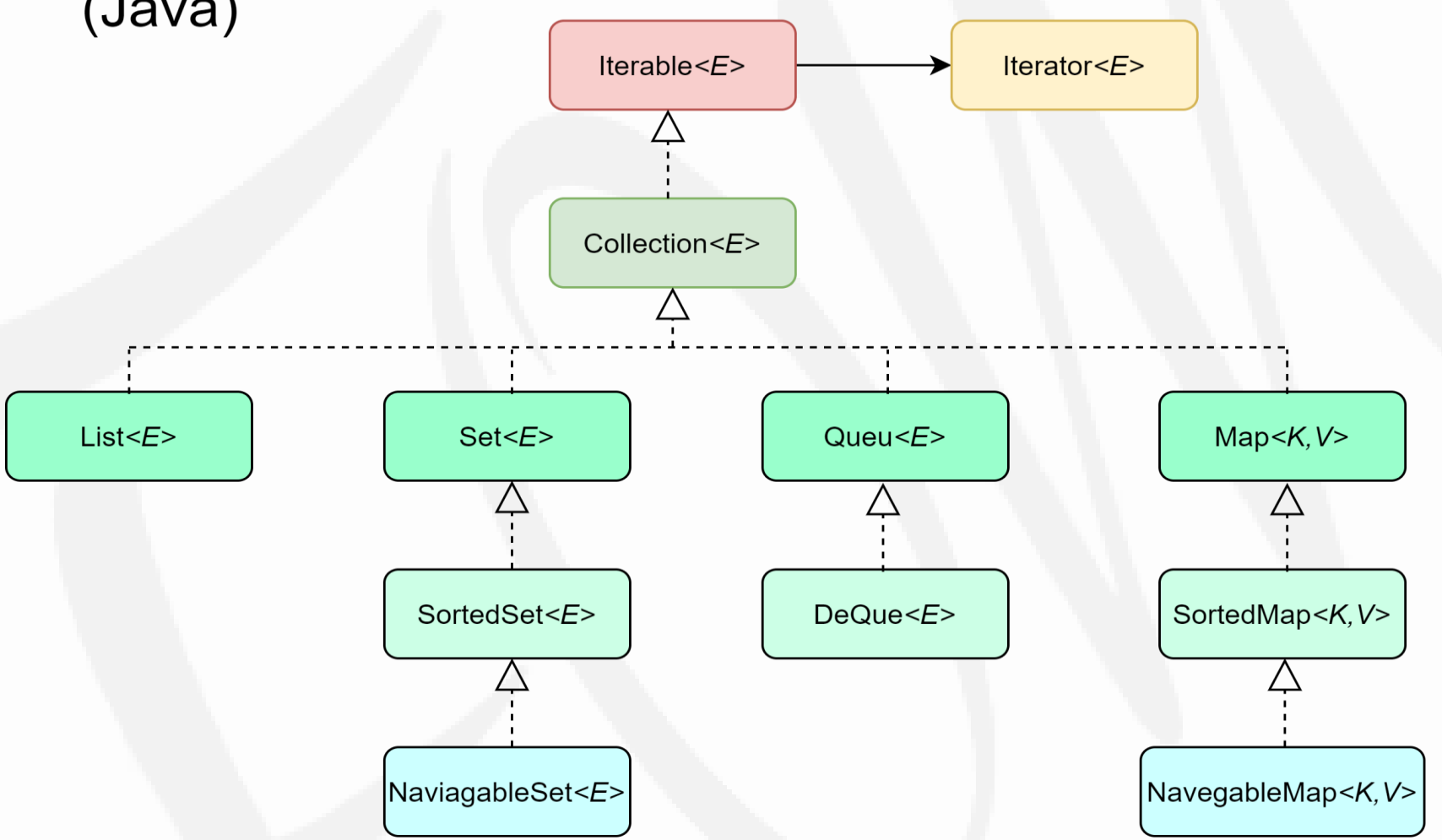
NO permite objetos duplicados

No permite el acceso directo

Ineficiente al ordenar (cuando se puede)

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection Framework (Java)



ArrayList
LinkedList
Stack
Vector(sync)

HashSet
LinkedHashSet
Treeset (SortedSet)

PriorityQueue
ArrayDeque(Deque)
LinkedList(Deque)

HashMap
HashLinkedMap
HashTable(sync)
TreeMap(SortedMap)

Clases de implementación o Implementation Classes

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Operaciones Básicas)

int size(); Determina el tamaño de la colección

boolean isEmpty(); Identifica sí la colección está vacía

boolean contains(Object element); Identifica un objeto dentro de la colección

boolean add(Object element); Añade un objeto a la colección

boolean remove(Object element); Remueve un objeto de la colección

Iterator ***iterator()***; Permite recorrer una colección

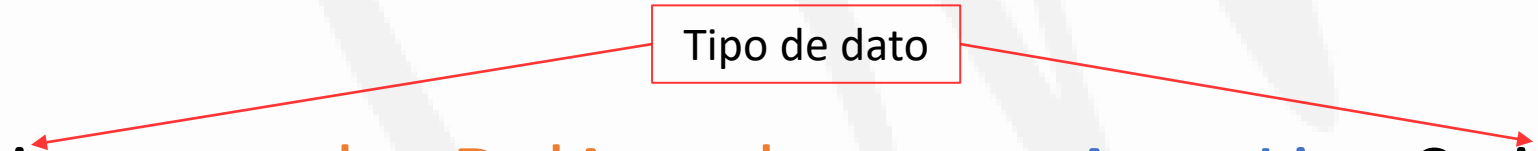
Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(List)

List

```
ArrayList nombreDelArreglo = new ArrayList();
```




```
ArrayList<String> nombreDelArreglo = new ArrayList<String>();
```

```
nombreDelArreglo.add(elemento)
```

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(ArrayList Sintaxis)

List

```
ArrayList nombreDelArreglo = new ArrayList();
```

Tipo de dato

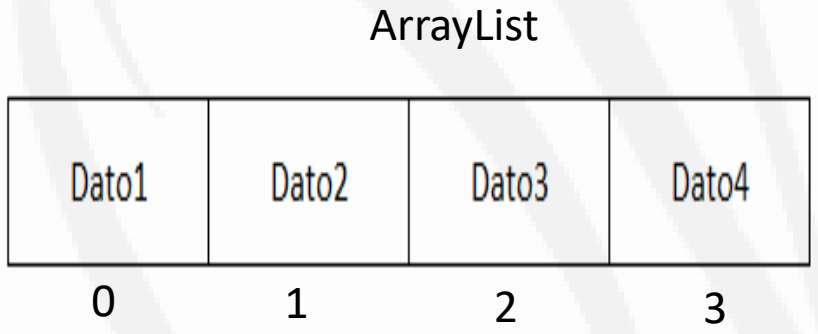
```
ArrayList<String> nombreDelArreglo = new ArrayList<String>();
ArrayList<Class> nombreDelArreglo = new ArrayList<Class>();
```

```
nombreDelArreglo.add(new nombreDeClass(parámetros));
```

Ing. Fernando Ospina Marín
fospinam@gmail.com

Representación

(ArrayList)

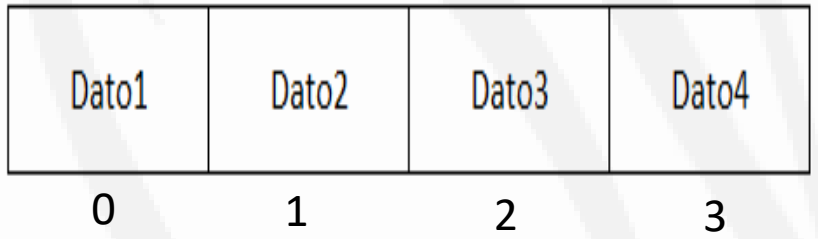


Ing. Fernando Ospina Marín
fospinam@gmail.com

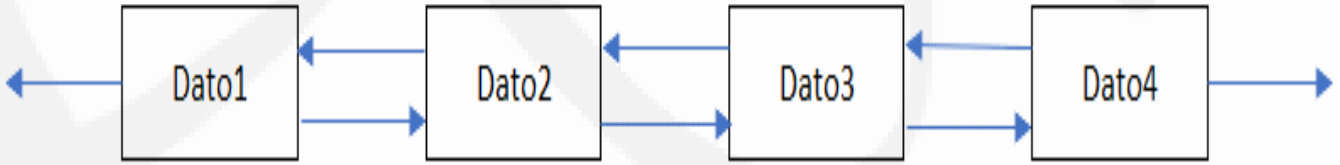
Representación

(ArrayList - LinkedList)

ArrayList



LinkedList



Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(List Sintaxis)

List

```
List objeto = new LinkedList();
```

```
LinkedList<Class> objeto = new LinkedList<Class>();
```

```
objeto.add(new Clase (parámetrosdelconstructor);
```

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Set Sintaxis)

Set (listas de datos NO duplicados y NO ordenados)

```
Set<String> objeto = new HashSet<>();

objeto.add(datos);
```

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Set Sintaxis)

Set (listas de datos NO duplicados y Ordenados)

```
Set<String> objeto = new TreeSet<>();

objeto.add(dato);
```

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Map Sintaxis)

Map (listas de datos NO duplicados y NO ordenados)

```
Set<String> objeto = new HashSet<>();

objeto.add(datos);
```

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Map Sintaxis)

En resumen

Un **Set** es un conjunto genérico de valores sin elementos duplicados.

Un **TreeSet** es un conjunto donde se ordenan los elementos.

Un **HashSet** es una implementación de un Conjunto donde los elementos no están ordenados ni ordenados y es más rápido que un TreeSet.

Ing. Fernando Ospina Marín
fospinam@gmail.com

Collection

(Map Sintaxis)

Map (datos NO duplicados y NO ordenados)

```
HashMap<Llave,Valor> objeto = new HashSet< Llave,Valor >();

HashMap<String,Persona> personal = new HashMap<String,Persona>();

objeto.put("1", new Clase(“parámetros”));
```

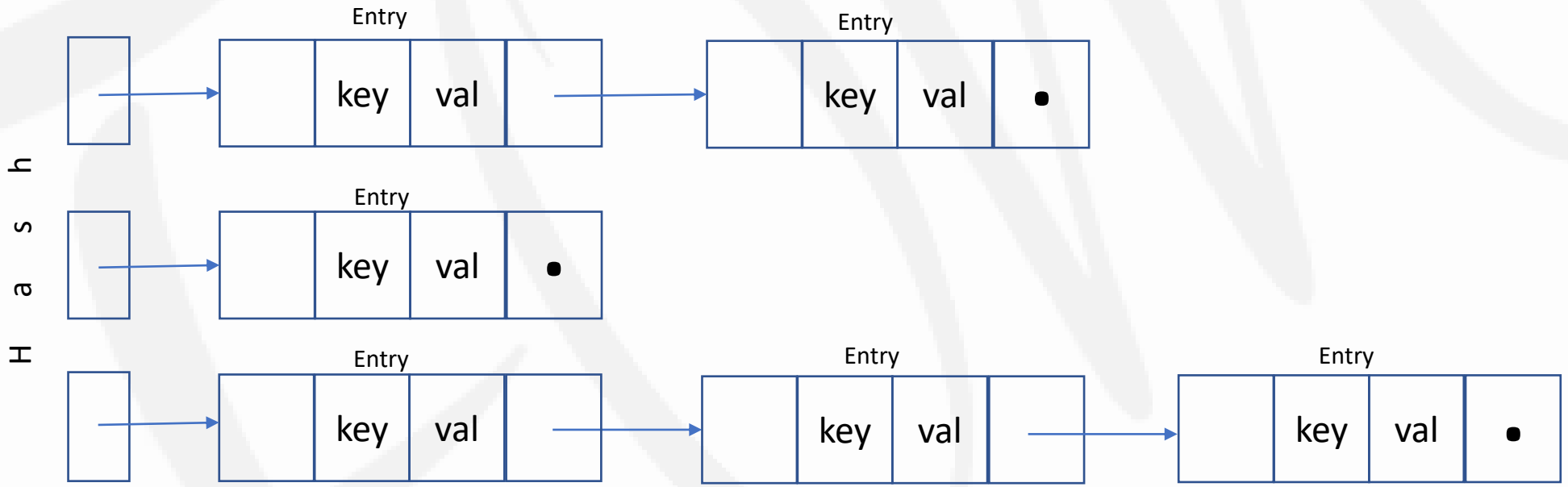
Métodos

```
objeto.entrySet();      objeto.getKey();      objeto.getValue();
```

Ing. Fernando Ospina Marín
fospinam@gmail.com

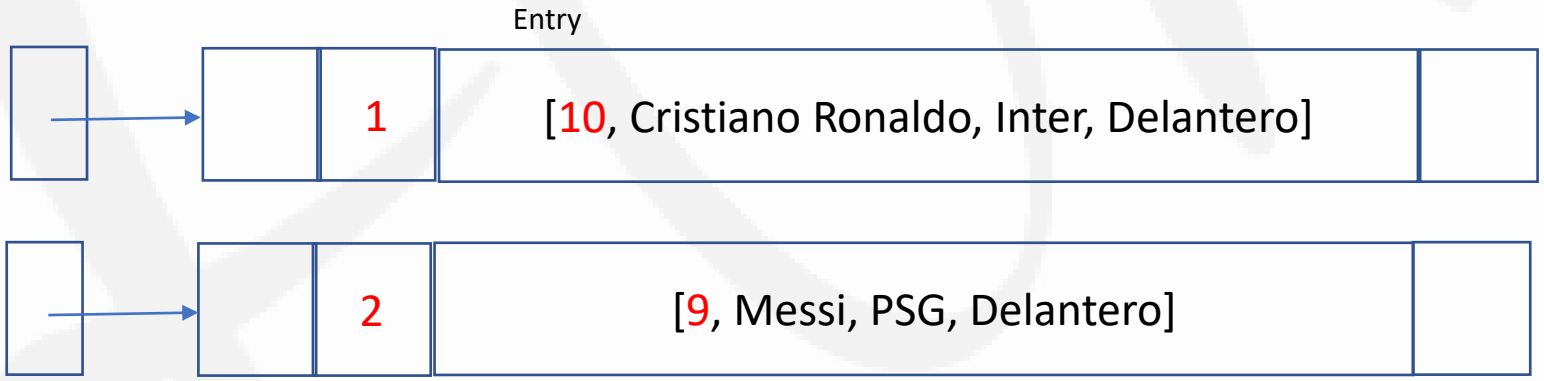
Representación

(Map - Hash)



Ing. Fernando Ospina Marín
fospinam@gmail.com

```
Map<Integer,Jugador> objeto = new HashMap<Integer,Jugador>();
```



Ing. Fernando Ospina Marín
fospinam@gmail.com