

APUNTES PARCIAL N°1 ESTRUCTURA DE DATOS:

- Estructura básica del algoritmo que soluciona el problema de un laberinto:

```
public void resuelve(int x, int y){
    if(resolverLaberinto(x, y)){
        laberinto[x][y] = 'X';
    }
}

public boolean resolverLaberinto(int x, int y){ //declaracion de la funcion con sus parametros

    //validar que no se desborde el laberinto, es decir, que x e y están dentro de los limites

    if(!esValido(x,y){
        return false;
    }

    //Validar los casos invalidos, (que ya haya pasa por ahí, que el camino sea fallido, o que sea una ubicación por donde
    no se puede transitar)|
    if(!esValido(x,y)){
        return false;
    }

    //validar que ya haya llegado al punto destino
    if(final(x,y)){
        return true;
    }

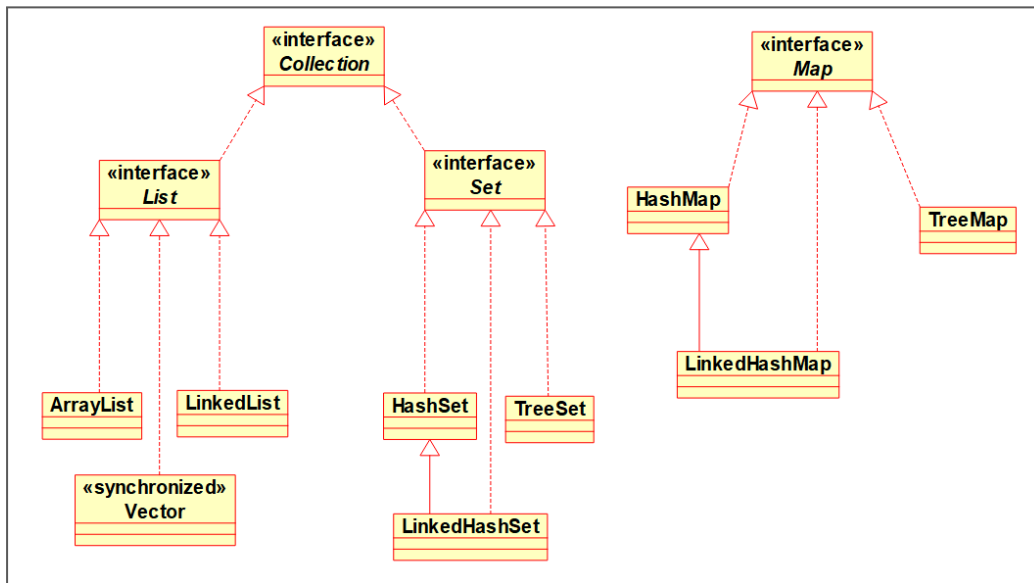
    //Si no se cumple ninguna de esas condiciones, avanzamos
    laberinto[x][y] = "marca de paso(huella)"
    boolean resultado;

    //seguir el patron de movimiento

    //Si no se puede mover a ninguna posicion del patron de movimiento, es un caso fallido
    laberinto[x][y] = "fallido"
    return false;
}
```

Colecciones:

Una colección representa un grupo de objetos (elementos)
interfaz “Collection” para almacenar estos grupos y manipularlos (añadir, eliminar, obtener el tamaño de la colección).



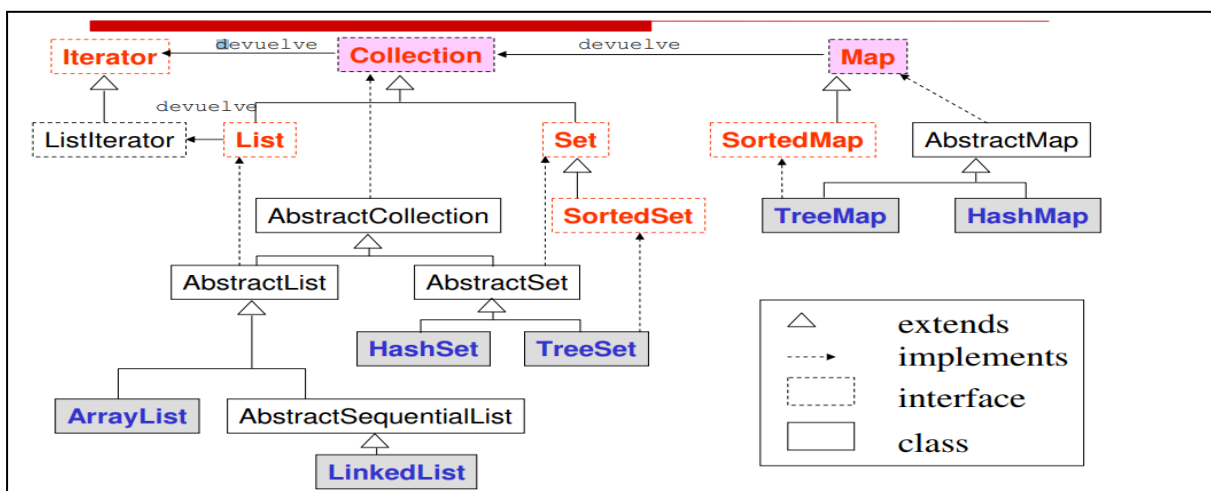
Collection: La interfaz raíz en la jerarquía de colecciones, a partir de la cual se derivan las interfaces Set, Queue y List.

Set: Una colección que no contiene duplicados.

List: Una colección lineal que puede contener elementos duplicados.

Map: Asocia claves con valores y no puede contener claves duplicadas.

Queue: Por lo general, una colección del tipo en entrar primero en salir, que modela a una línea de espera.



Tipos de listas: secuenciales y enlazadas

Collection -> List -> ArrayList, LinkedList, Vector, CopyOnWriteArrayList

Collection -> Set -> HashSet, LinkedHashSet, TreeSet, EnumSet, CopyOnWriteArraySet, ConcurrentSkipListSet

Map -> HashMap, LinkedHashMap, TreeMap, EnumMap, WeakHashMap, Hashtable, ConcurrentHashMap

Collection -> Queue -> ArrayDeque, LinkedBlockingDeque, LinkedList, PriorityQueue, PriorityBlockingQueue

```
Iterator it = datos.keySet().iterator() //obtener Iterator de un HashMap
```

```
Iterator<Integer> it = lista.iterator();  
    int num;  
    while (it.hasNext()) {  
        num = it.next();  
        System.out.println(num);  
    }
```

