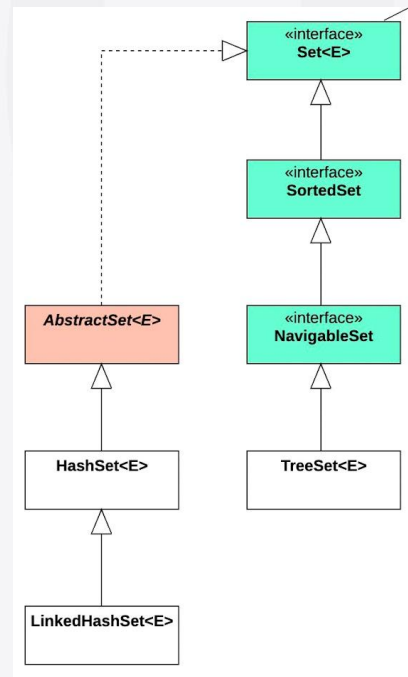


# SET

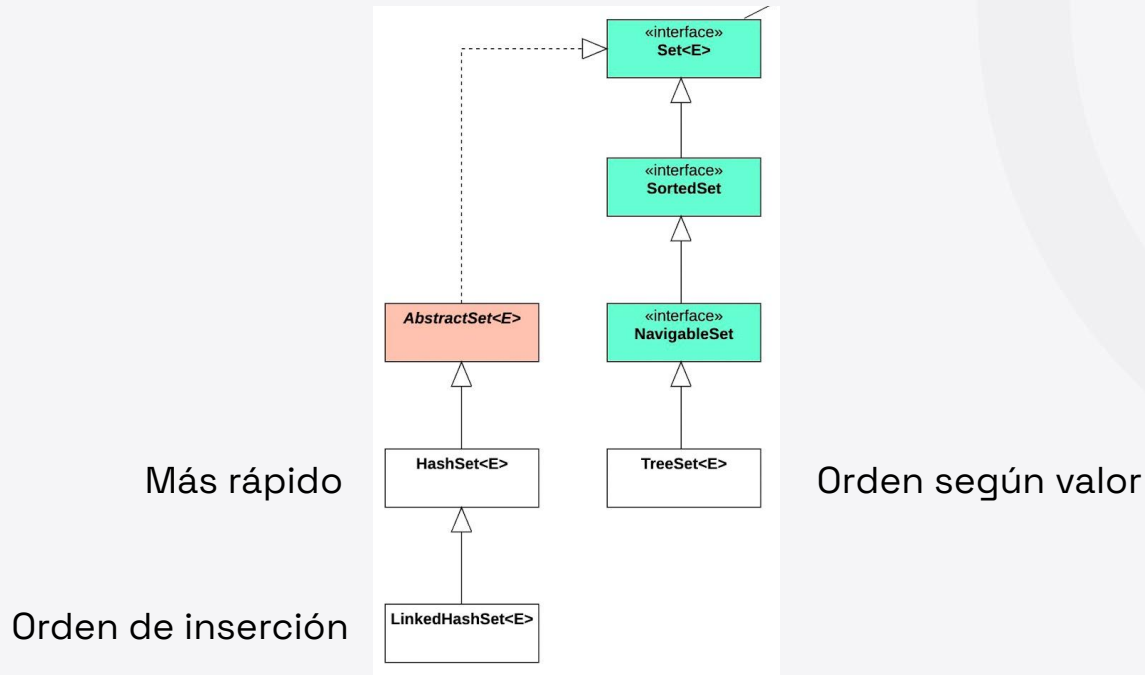
## COLECCIONES

# SET<E>

- Interfaz
- Hereda de Collection<E>
- Abstracción de un conjunto
  - sin elementos repetidos
  - como mucho, un elemento *null*
- No añade métodos nuevos (Collection<E>)
- No hay acceso posicional
- *equals* devuelve true si ambos tienen los mismos elementos.



# ALGUNAS IMPLEMENTACIONES DE SET



# CREACIÓN DE UN SET

- *Set.of*
  - Crea un Set no modificable y que no permite null
  - No hay orden
- A través de sus implementaciones
  - Uso del constructor

# OBTENCIÓN DE LOS ELEMENTOS DE UN SET

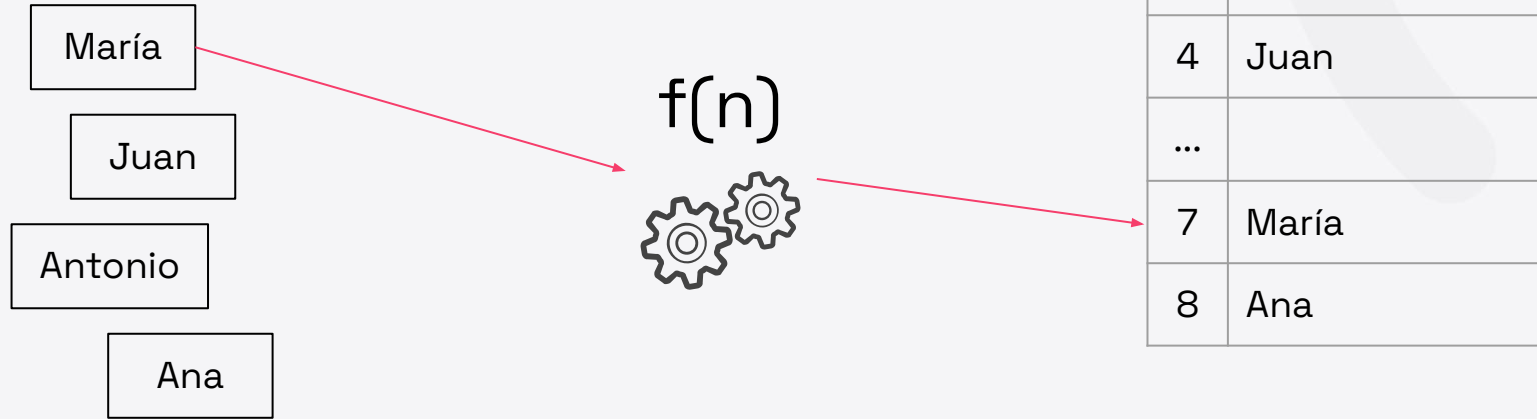
- Solamente se puede hacer iterando
  - bucle for each
  - iterador
- No hay acceso posicional
- “No se puede obtener un único elemento”

# HASHSET<E>

- Implementación de un  $Set<E>$  basada en una tabla hash (*HashMap*)
  - No garantiza el orden
  - Permite un elemento null
  - Operaciones de acceso en tiempo constante ( $O(1)$ )
- Se obtiene instanciando: equilibrio entre
  - tamaño
  - factor de carga (*rehashing*)

# HASHSET<E>

- Función hash: hashCode



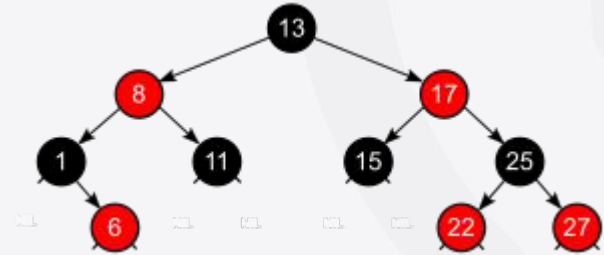
# LINKEDHASHSET<E>

- Similar al anterior.
- Mantiene además una lista doblemente enlazada
  - Mantiene el orden de inserción.
  - No hay acceso posicional (`get(7)`)
- Rendimiento similar a *HashSet<E>*
  - Levemente peor por mantener la lista
  - Levemente mejor para iterar



# TREESet<E>

- Almacena sus valores en un *árbol binario red-black*
  - Mantenimiento del orden (orden natural, ...)
  - Los elementos deben implementar Comparable
  - No se permiten nulos
- Peor rendimiento que el resto de opciones ( $O(\log(N))$ )



# ITERAR SOBRE UN TREESSET<E>

- Orden ascendente
  - for each
  - iterator
- Orden descendente
  - *descendingIterator()*
  - *descendingSet()*
    - Devuelve una vista en orden descendente

# OPERACIONES CON TREESSET<E>

- *first()*, *last()*: primer/último elemento de la colección
- *pollFirst()*, *pollLast()*: obtener y eliminar el primer/último elemento
- *higher(e)*, *ceiling(e)*, *lower(e)*, *floor(e)*: obtener elemento inmediatamente superior/inferior a uno dado.
- *subSet(from, to)*, *headSet(to)*, *tailSet(from)*: obtener una vista de una porción de la colección.

# EN RESUMEN

- Set es una colección que no permite repetidos y no tiene acceso posicional.
- Varias implementaciones
  - *HashSet*: tabla hash. No hay orden. Un elemento null.
  - *LinkedHashSet*: orden de inserción. Un elemento null.
  - *TreeMap*: orden natural. No se permiten nulos.