

Uso de Iteradores con colecciones

```
package .... ;

import java.util.*;

public class PruebaColecciones {

    static int contador=0;

    public static void main(String[] args) {

        LinkedList<Integer> num = new LinkedList<Integer>();

        ArrayList<String> list = new ArrayList<String>();

        list.add("Text 1");num.add(new Integer(20));

        list.add("Text 2");num.add(new Integer(30));

        list.add("Text 3");num.add(new Integer(40));

        System.out.println("1. -----iterator-----");

        Iterator<String> iteStr = list.iterator();

        Iterator<Integer> iteNum = num.iterator();

        while (iteStr.hasNext()){//¿tiene próximo?{

            contador++;

            System.out.println(iteStr.next()+" /list/ "+contador+" ");

        }

        while (iteNum.hasNext()){//¿tiene próximo?{

            contador++;

            System.out.println(iteNum.next()+" /num/ "+contador+" ");

        }

    }

}
```

```

list.add(2, "Text X");

num.add(2,new Integer(88));

System.out.println("***** Desplacé Text 3 con X y 40 con 88
*****");

Iterator<String> iterator1 = list.iterator();

System.out.println("-----iterator String sin bucle-----");

if(iterator1.hasNext()){//¿tiene próximo?{

    System.out.println(iterator1.next()+" getClass
"+list.getClass().getName());

    //iterator2.forEachRemaining();

    System.out.println("2. -----iterator String con
forEachRemaining-----");

    Iterator<String> iterator2 = list.iterator();

    //formato: forEachRemaining(Consumer <? super String> action) void

    //POR CADA ELEMENTO RESTANTE

    System.out.println("<< Se suma 1 al contador, pero no hay bucle, por
lo tanto se suma una sola vez >>");

    contador++;

    iterator2.forEachRemaining( t -> System.out.println(t+"-- sin bucle:
contador ---> "+contador) );

    Iterator<Integer> iterator3 = num.iterator();

```

```

    if(iterator3.hasNext()){

        //iterator3.forEachRemaining();

        System.out.println(iterator3.next()+" getClass ----->>>
"+num.getClass());

        System.out.println("4. ----- iterator Integer
forEachRemaining -----");

        contador++;

        Iterator<Integer> iterator4 = num.iterator();

        // formato: forEachRemaining(Consumer <? super Integer> action)
void

        //POR CADA ELEMENTO RESTANTE

        iterator4.forEachRemaining( r -> System.out.println(r+" sin bucle,
contador --> "+contador) );

    }

//Uso de Iterator con diferentes tipos de colecciones:

System.out.println("5.----- Prueba con Map: -----");

Map<Integer,String> hm = new HashMap();

System.out.println("se crea map parametrizado: <Integer,String>");

hm.put(new Integer(1),"Luis");

hm.put(new Integer(2),"Amelia");

hm.put(new Integer(3),"Julio");

System.out.println("----->>> Se agregan tres elementos 1,Luis; 2,Amelia y
3,Julio");

System.out.println("Iterador...");

```

```

Iterator it = hm.entrySet().iterator(); //iterar

while (it.hasNext()) { //mientras se encuentre uno más...

    Map.Entry e = (Map.Entry)it.next();

    System.out.println("Mapa: Clave--->>> "+e.getKey() + ", valor --->>> " +
e.getValue());

}

System.out.println("6. ----- Prueba con Set: -----");

HashSet<String> s1 = new HashSet <String>();

HashSet<String> s2 = new HashSet <String>();

s1.add("A"); s1.add("B");

s1.add("A"); // Ok, seguimos teniendo un sólo "A".

s1.add(null); s1.add(null); // Ok, seguimos teniendo un sólo null.

s2.addAll(s1);

    System.out.println("7. -----Ejemplos de uso de métodos con la
colección set-----");

System.out.println("contains s2 A? : " + s2.contains("A"));

for(String x : s2)System.out.print(x + " ");

    Iterator it2 = s2.iterator();          //iterar

System.out.print("\n Contenido de s2: ");

while(it2.hasNext())System.out.print(it2.next() + " ");

s1.clear(); //remueve los elementos del set

System.out.println("\n equals: " + s2.equals(s1));

```

```

        System.out.println("hashcode de s2: " + s2.hashCode() + " y de s1: " +
s1.hashCode());

        System.out.println("isEmpty: " + s1.isEmpty());

        System.out.println("remove: " + s2.remove("A"));

        System.out.println("longitud de toArray de s2 : " + s2.toArray().length);

        System.out.println("size: " + s2.size());

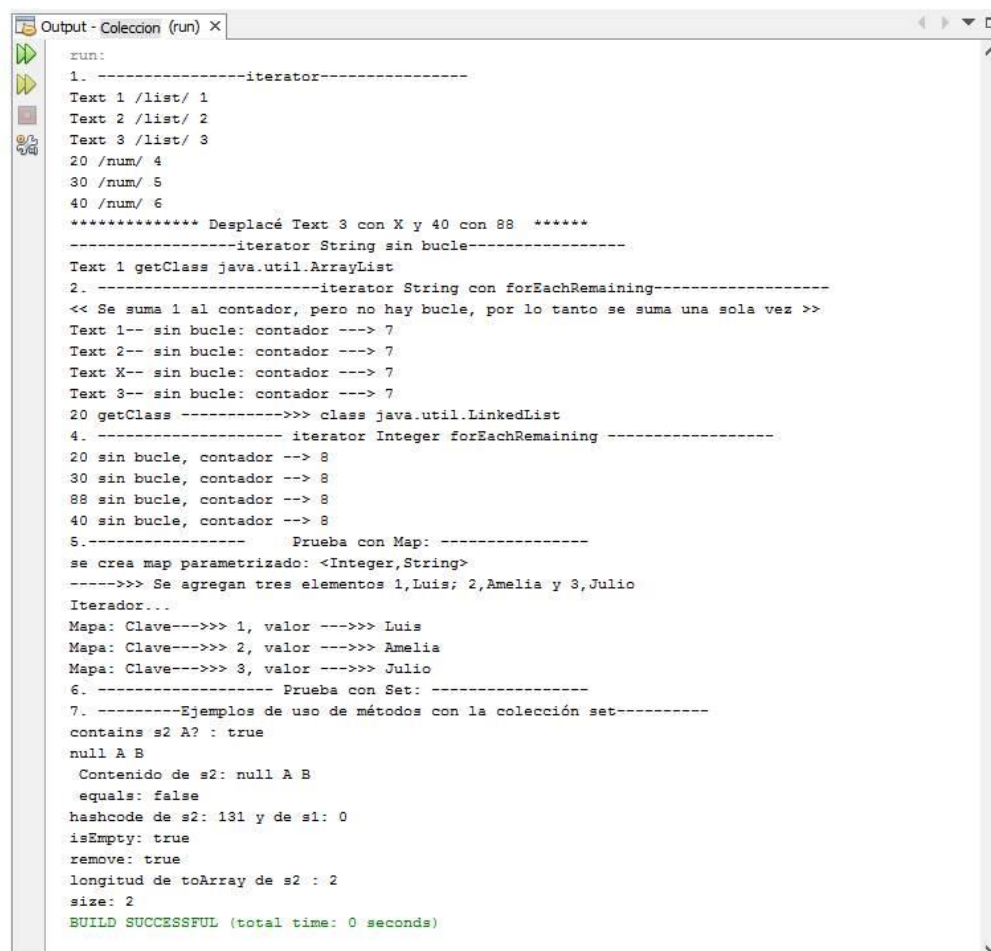
    }

}

}

```

Ejecución



```

Output - Coleccion (run) X
run:
1. -----iterator-----
Text 1 /list/ 1
Text 2 /list/ 2
Text 3 /list/ 3
20 /num/ 4
30 /num/ 5
40 /num/ 6
***** Desplacé Text 3 con X y 40 con 88 *****
-----iterator String sin bucle-----
Text 1 getClass java.util.ArrayList
2. -----iterator String con forEachRemaining-----
<< Se suma 1 al contador, pero no hay bucle, por lo tanto se suma una sola vez >>
Text 1-- sin bucle: contador --> 7
Text 2-- sin bucle: contador --> 7
Text X-- sin bucle: contador --> 7
Text 3-- sin bucle: contador --> 7
20 getClass ----->>> class java.util.LinkedList
4. ----- iterator Integer forEachRemaining -----
20 sin bucle, contador --> 8
30 sin bucle, contador --> 8
88 sin bucle, contador --> 8
40 sin bucle, contador --> 8
5.----- Prueba con Map: -----
se crea map parametrizado: <Integer,String>
----->>> Se agregan tres elementos 1,Luis; 2,Amelia y 3,Julio
Iterador...
Mapa: Clave-->>> 1, valor -->>> Luis
Mapa: Clave-->>> 2, valor -->>> Amelia
Mapa: Clave-->>> 3, valor -->>> Julio
6. ----- Prueba con Set: -----
7. -----Ejemplos de uso de métodos con la colección set-----
contains s2 A? : true
null A B
Contenido de s2: null A B
equals: false
hashcode de s2: 131 y de s1: 0
isEmpty: true
remove: true
longitud de toArray de s2 : 2
size: 2
BUILD SUCCESSFUL (total time: 0 seconds)

```

