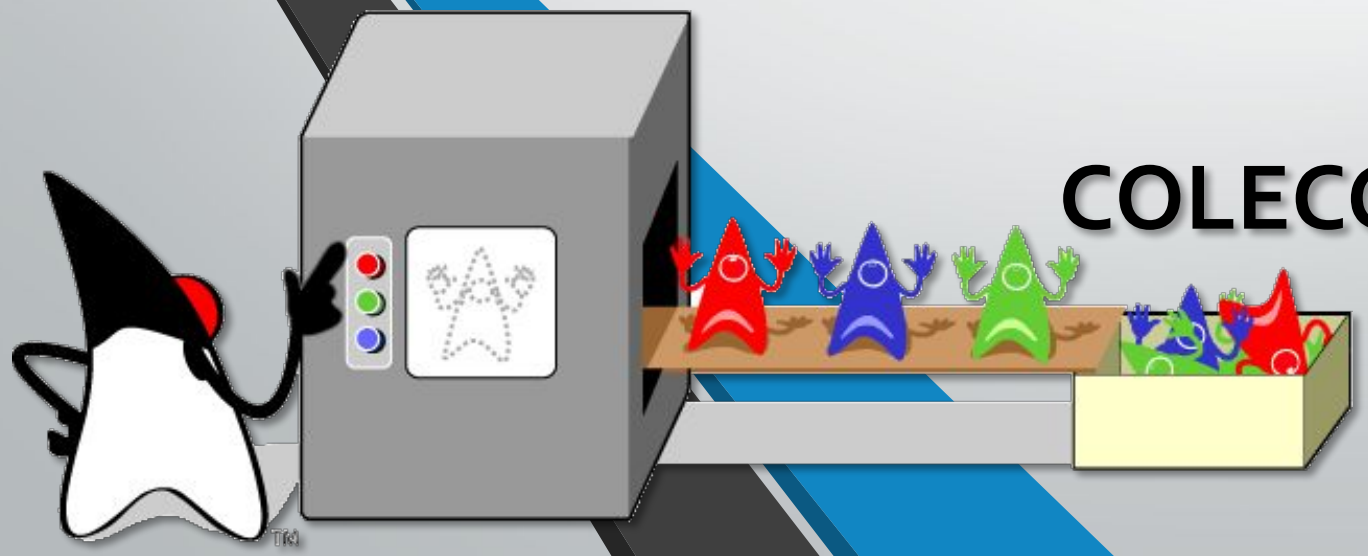
JAVA

TEMA 11:

COLECCIONES: SETS Y MAPS



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

ÍNDICE

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

JAVA

UTILIZACIÓN AVANZADA DE CLASES

1. Introducción

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Introducción

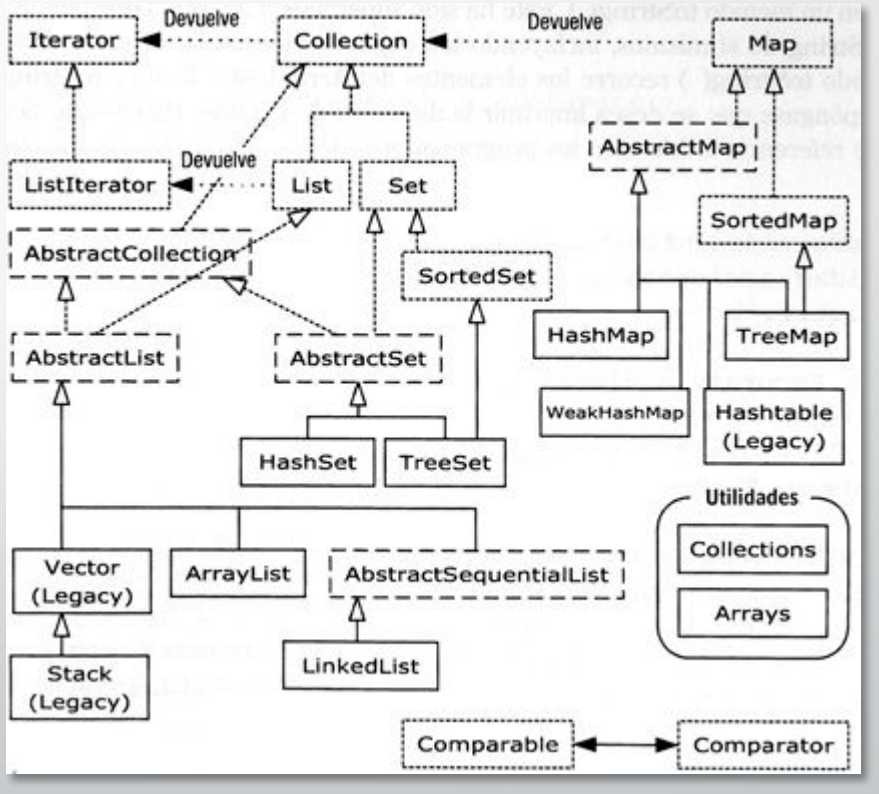
- Como vimos en el tema anterior, las colecciones son objetos que a su vez contienen objetos, y que se usan para almacenar, obtener, manipular y comunicar datos incluidos en éstas.
- También vimos que las colecciones se diferencian de los arrays en que su tamaño no es fijo, esto es, son dinámicas. Se pueden realizar operaciones de añadir, eliminar, obtener, buscar o recorrer una colección.

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Introducción

- La Java Collections Framework (JCF) está constituida por:
  - Interfaces principales: **List** (listas), **Set** (conjuntos) y **Map** (mapas).
  - Interfaces de soporte: Iterator, ListIterator, Comparable y Comparator.
  - Implementaciones de las interfaces, que sirven para almacenar y manipular grupos de datos como una sola unidad, como una colección (HashSet, TreeSet, ArrayList, LinkedList, HashMap, TreeMap, etc.).



Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Introducción

- Métodos de las interfaces:
  - Listas y Conjuntos:
    - get (*)
    - add
    - remove
    - contains
  - Mapas:
    - get
    - put
    - remove
    - containsKey

(*) Los conjuntos no tienen método *get*, se refiere a la acción de retornar el elemento (para los conjuntos, a través del *it.next();*)

Listas y conjuntos

Estructura	get	add	remove	contains
ArrayList	O(1)	O(1)	O(n)	O(n)
LinkedList	O(n)	O(1)	O(1)	O(n)
HashSet	O(1)	O(1)	O(1)	O(1)
LinkedHashSet	O(1)	O(1)	O(1)	O(1)
TreeSet	O(log n)	O(log n)	O(log n)	O(log n)

Mapas:

Estructura	get	put	remove	containsKey
HashMap	O(1)	O(1)	O(1)	O(1)
LinkedHashMap	O(1)	O(1)	O(1)	O(1)
TreeMap	O(log n)	O(log n)	O(log n)	O(log n)

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

JAVA

UTILIZACIÓN AVANZADA DE CLASES

2. Sets (Conjuntos)

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Sets (Conjuntos)

- La diferencia entre las listas y los conjuntos es que, en los conjuntos, no puede haber elementos repetidos, mientras que en las listas si.
- Veamos un ejemplo:

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Sets (Conjuntos)

```java
public class Conjuntos {

    public static void main(String[] args) {
        List<String> lista = new ArrayList<String>();
        lista.add("Victor");
        lista.add("Amaya");
        lista.add("Amaya"); // Los elementos pueden estar repetidos
        lista.add("Javier");
        System.out.println(lista);

        Set<String> conjunto = new HashSet<String>();
        conjunto.add("Victor");
        conjunto.add("Amaya");
        conjunto.add("Amaya"); // Los elementos solo pueden estar una vez
        conjunto.add("Javier");
        System.out.println(conjunto);
    }
}
```

```
- Conjuntos (run)
run:
[Victor, Amaya, Amaya, Javier]
[Victor, Javier, Amaya]
```

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Sets (Conjuntos)

- Para recorrer un conjunto, tenemos dos opciones:
  - Con un *for-each*.
  - Con un iterador.

```
//Recorrido de un conjunto
public static void recorrerConjunto(Set<String> conjunto){
    Iterator<String> it = conjunto.iterator();
    String valor;
    //Recorrido con iterador
    while(it.hasNext()){
        valor = it.next();
        System.out.println(valor);
    }
    //Recorrido con for-each
    for(String elem : conjunto){
        System.out.println(elem);
    }
}
```

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Sets (Conjuntos)

- Los tipos de conjuntos más utilizados son los HashSet y los LinkedHashSet.
- La diferencia es que en los LinkedHashSet los elementos se guardan de manera ordenada.
- Veamos un ejemplo:

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Sets (Conjuntos)

```java
public class Conjuntos {

    public static void main(String[] args) {
        List<String> lista = new ArrayList<String>();
        lista.add("Victor");
        lista.add("Amaya");
        lista.add("Amaya"); // Los elementos pueden estar repetidos
        lista.add("Javier");
        System.out.println(lista);

        Set<String> conjunto = new HashSet<String>();
        conjunto.add("Victor");
        conjunto.add("Amaya");
        conjunto.add("Amaya"); // Los elementos solo pueden estar una vez
        conjunto.add("Javier");
        System.out.println(conjunto);

        Set<String> lhs = new LinkedHashSet<String>();
        lhs.add("Victor");
        lhs.add("Amaya");
        lhs.add("Amaya"); // Los elementos solo pueden estar una vez
        lhs.add("Javier");
        System.out.println(lhs);
    }
}
```

```
- Conjuntos (run) ×
run:
[Victor, Amaya, Amaya, Javier]
[Victor, Javier, Amaya]
[Victor, Amaya, Javier]
```

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

JAVA

ENTRADA / SALIDA EN JAVA

3. Maps (Mapas)

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Maps (Mapas)

- Los mapas son un tipo de colección basada en claves, donde los objetos almacenados en la misma no tienen un índice numérico basado en su posición, sino una clave que lo identifica de forma única dentro de la colección. Una clave puede ser cualquier tipo de objeto.

Array

Value
New York
Boston
Mexico
Kansas
Detroit
California

Hash Table

Key	Value
1	New York
2	Boston
3	Mexico
4	Kansas
5	Detroit
6	California

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Maps (Mapas)

- El uso de Maps es útil para realizar búsquedas de objetos a partir de un dato que lo identifica. Por ejemplo, un conjunto de objetos de tipo Empleado es más práctico almacenarlos en un mapa asociándoles como clave el "dni", que guardarlos en una lista en que a cada empleado se le asigna un índice según el orden de almacenamiento.

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Maps (Mapas)

- La sintaxis de creación de un HashMap es:

  HashMap<Object, Object> tabla = new HashMap<>();

- Por ejemplo:

  HashMap<Integer, String> tabla = new HashMap<>();

  HashMap<String, Cliente> tabla = new HashMap<>();

- También sería válido:

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

  //porque Map es la interfaz y HashMap la clase que la implementa

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

3.- Maps (Mapas)

- Algunos de los métodos de la interface Map son los siguientes:
  - clear(): elimina todos los mapeos del Map.
  - containsKey(Object clave): devuelve true si el Map contiene un mapeo igual que el objeto pasado por parámetro: boolean existe = productos.containsKey(producto);
  - containsValue(Object valor): devuelve true si el Map mapea a uno o más claves del objeto valor pasado por parámetro.
  - get(Object clave): devuelve el objeto valor de la clave pasada por parámetro; o null si el Map no encuentra ningún mapeo con esta clave.
  - isEmpty(): devuelve true si el Map está vacío.
  - put(Clave clave, Valor valor): asigna el valor pasado con la clave especificada a otro objeto valor.
  - remove(Object clave): elimina el mapeo que tenga la clave pasada por parámetro si existe, devolviendo el valor de dicho mapeo.
  - size(): devuelve un entero con el número de mapeos del Map.

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Maps (Mapas)

- Otro elemento muy importante a la hora de trabajar con los Maps son los "*Iteradores*" (Iterator), ya que sirven para recorrer los Map (no podemos utilizar for como en las listas).
- Recuerda que los Iteradores tienen los métodos *hashNext()* y *next()* para recorrer una estructura de datos.
- *Veamos un ejemplo:*

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Maps (Mapas)

```java
//Muestra la información del mapa
public static void mostrarMapa(Map<Integer, String> mapa){
    //Declaramos e inicializamos el iterador para el mapa
    Iterator<Integer> it = mapa.keySet().iterator();
    int clave;
    while(it.hasNext()){
        clave = it.next();
        System.out.println("Clave: "+clave+" - Valor: "+mapa.get(clave));
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    Map<Integer, String> mapa = new HashMap<>();

    mapa.put(1, "Courtois");
    mapa.put(2, "Carvajal");
    mapa.put(3, "Nacho");
    mapa.put(4, "Vinicius");
    mapa.put(5, "Benzemá");
    mapa.put(6, "Mariano");
    mapa.put(7, "Asensio");
    mostrarMapa(mapa);


}
```

```
Output - PruebaHashSet (run)
run:
Clave: 1 - Valor: Courtois
Clave: 2 - Valor: Carvajal
Clave: 3 - Valor: Nacho
Clave: 4 - Valor: Vinicius
Clave: 5 - Valor: Benzemá
Clave: 6 - Valor: Mariano
Clave: 7 - Valor: Asensio
BUILD SUCCESSFUL (total time: 0 seconds)
```

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

3.- Maps (Mapas)

```
//Muestra la información del mapa
public static void mostrarMapa(Map<Integer, String> mapa){
    //Declaramos e inicializamos el iterador para el mapa
    Iterator<Integer> it = mapa.keySet().iterator();
    int clave;
    while(it.hasNext()){
        clave = it.next();
        System.out.println("Clave: "+clave+" - Valor: "+mapa.get(clave));
    }
}

public static void insertarMapa(Map<Integer, String> mapa){
    Scanner teclado = new Scanner(System.in);
    String nombre, resp;
    do{
        System.out.print("Nombre: ");
        nombre = teclado.nextLine();
        mapa.put(mapa.size(), nombre);
        System.out.println("¿Desea añadir más nombres? (si/no)");
        resp = teclado.nextLine();
    }while(resp.equalsIgnoreCase("SI"));
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    Map<Integer, String> mapa = new HashMap<>();

    insertarMapa(mapa);
    mostrarMapa(mapa);
```

```
Output - PruebaHashSet (run)
run:
Nombre: Carvajal
¿Desea añadir más nombres? (si/no)
si
Nombre: Benzemá
¿Desea añadir más nombres? (si/no)
si
Nombre: Raúl
¿Desea añadir más nombres? (si/no)
si
Nombre: Roberto Carlos
¿Desea añadir más nombres? (si/no)
si
Nombre: Ronaldo
¿Desea añadir más nombres? (si/no)
no
Clave: 0 - Valor: Carvajal
Clave: 1 - Valor: Benzem□
Clave: 2 - Valor: Ra□l
Clave: 3 - Valor: Roberto Carlos
Clave: 4 - Valor: Ronaldo
BUILD SUCCESSFUL (total time: 35 seconds)
```

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Maps (Mapas)

- Fíjate como lo único que cambia entre los diferentes tipos de Mapas (HashMap, TreeMap y LinkedHashMap) es la forma que tienen de almacenar la información:

```java
import java.util.Iterator;
import java.util.Map;
import java.util.TreeMap;
/**
 * @author OLG
 */
public class Test {
    public static void mostrarMapa(Map mapa){
        Iterator it = mapa.keySet().iterator(); //key
        while (it.hasNext()) {
            Integer clave = (Integer) it.next();
            System.out.println("Clave: " + clave + " -
        }
    }

    public static void main(String[] args) {
        Map<Integer, String> mapa = new TreeMap<>();

        mapa.put(1, "De Gea");
        mapa.put(15, "Ramos");
        mapa.put(3, "Piqué");
        mapa.put(5, "Carbajal");
        mapa.put(11, "Alba");
```

```
put - tablas (run) ×
run:
Clave: 1 -> Valor: De Gea
Clave: 3 -> Valor: Piqué
Clave: 5 -> Valor: Carbajal
Clave: 6 -> Valor: Iniesta
Clave: 7 -> Valor: Aduriz
Clave: 8 -> Valor: Koke
Clave: 11 -> Valor: Alba
Clave: 14 -> Valor: Saul
Clave: 15 -> Valor: Ramos
Clave: 16 -> Valor: Busquets
Clave: 18 -> Valor: Nolito
```

```java
package tablas;
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Map;
/**
 * @author OLG
 */
public class Test {
    public static void mostrarMapa(Map mapa){
        Iterator it = mapa.keySet().iterator(); //keySet m
        while (it.hasNext()) {
            Integer clave = (Integer) it.next();
            System.out.println("Clave: " + clave + " -> Va
        }
    }

    public static void main(String[] args) {
        Map<Integer, String> mapa = new LinkedHashMap<>();

        mapa.put(1, "De Gea");
        mapa.put(15, "Ramos");
        mapa.put(3, "Piqué");
        mapa.put(5, "Carbajal");
        mapa.put(11, "Alba");
        mapa.put(16, "Busquets");
```

```
ut - tablas (run) ×
run:
Clave: 1 -> Valor: De Gea
Clave: 15 -> Valor: Ramos
Clave: 3 -> Valor: Piqué
Clave: 5 -> Valor: Carbajal
Clave: 11 -> Valor: Alba
Clave: 16 -> Valor: Busquets
Clave: 14 -> Valor: Saul
Clave: 8 -> Valor: Koke
Clave: 6 -> Valor: Iniesta
Clave: 18 -> Valor: Nolito
Clave: 7 -> Valor: Aduriz
```

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Maps (Mapas)

- *Veamos otro ejemplo:*

Test.java | Empleado.java

Source | History

```java
package mapas;
/**
 * @author OLG
 */
public class Empleado {

    private String dniEmp;
    private String nombre;
    private int edad;

    public Empleado() {
    }

    public Empleado(String dniEmp, String nombre, int edad) {
        this.dniEmp = dniEmp;
        this.nombre = nombre;
        this.edad = edad;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
```

Test.java | Empleado.java

Source | History

```java
    public static boolean introducirMas(){
        boolean mas = false;
        Scanner entrada = new Scanner (System.in);
        System.out.println("¿Desea introducir más empleados? (si/no)");
        if (entrada.nextLine().equalsIgnoreCase("si")){
            mas = true;
        }
        return mas;
    }

    public static void mostrarMapa(Map mapa){
        Empleado empleadoAux;
        String clave;
        Iterator it = mapa.keySet().iterator(); //keySet me devuelve todas las claves
        while (it.hasNext()) {
            clave = (String) it.next();
            System.out.print("Clave: " + clave);
            empleadoAux = (Empleado) mapa.get(clave);
            System.out.print(" -> Nombre: " + empleadoAux.getNombre());
            System.out.println(" -> Edad: " + empleadoAux.getEdad());
        }
    }

    public static void rellenarMapa(Map mapa){
        do{
            Empleado empleadoAux = new Empleado(pedirDni(), pedirNombre(), pedirEdad());
            mapa.put(empleadoAux.getDniEmp(), empleadoAux);
        }while(introducirMas());
    }

    public static void main(String[] args) {

        Map<String, Empleado> mapa = new HashMap<>();
        rellenarMapa(mapa);
        mostrarMapa (mapa);
    }
}
```

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

2.- Sets (Conjuntos)

- Podemos recorrer un mapa a través de:
  - Un *for-each*.
  - Un iterador.

```
//Recorrido de un mapa
public static void recorrerMapa(Map<Integer, String> mapa){
    Iterator<Integer> it = mapa.keySet().iterator();
    Integer valor;
    //Recorrido con iterador
    while(it.hasNext()){
        valor = it.next();
        System.out.println(mapa.get(valor));
    }

    //Recorrido con for-each
    for(Integer i : mapa.keySet()){
        System.out.println("Clave: "+i+" - Valor: "+mapa.get(i));
    }
}
```

JAVA

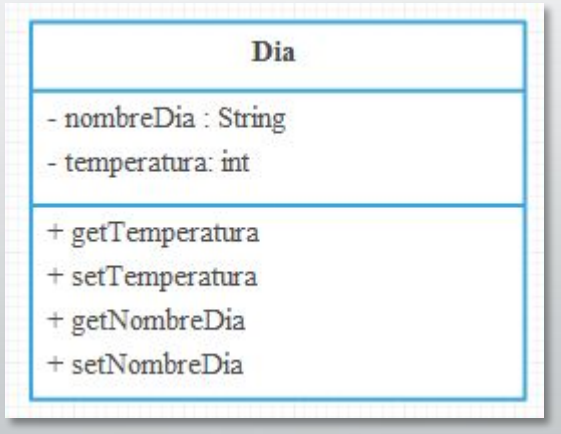
ENTRADA / SALIDA EN JAVA

4. Ejercicios de Consolidación

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 01.- (<u>OBLIGATORIO</u>)** Diseña programa que almacene las temperaturas medias de un mes. Para ello crearemos una HashMap de 31 posiciones relleno, el campo **<u>Clave</u>**, con el número del día, y el campo **<u>Valor</u>** con objetos de la clase Día. La clase Día contiene los siguientes atributos:



Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- Hasta que el usuario pulse 5, mostrar un menú que nos permita:
  1. Rellenar de forma aleatoria las temperaturas. Además el día 1 del mes no tiene porqué ser un Lunes, será un día aleatorio de la semana.
  2. Mostrar las temperaturas. Ejemplo*: Jueves día 1: 40 grados, Viernes día 2: 35 grados, Sábado día 3: 38 grados…*
  3. Visualizar la temperatura media del mes.
  4. Día o días más calurosos del mes. Ejemplo: *El día o días más calurosos fueron:*
     - *El Jueves día 1 con 40 grados.*
     - *El Sábado día 18 con 40 grados.*
  5. *Salir del programa.*
- *Fíjate que necesitarás un array con el nombre de los días de la semana.*

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 02.- (OPTATIVO)** Realizar un programa en JAVA en el realices la gestión de una pequeña tienda de deportes. Ayudate de un HashMap, donde el campo ***Clave*** será un código de producto (Integer o String, como prefieras), y el campo ***Value*** será un objeto de la clase Producto.
- La clase Producto tendrá 3 atributos:
  - Nombre del producto: de tipo String.
  - Precio: de tipo float.
  - Stock: de tipo int.
- Se le mostrarán al usuario un menú con 3 opciones:
  1. ***MENU DE ADMINISTRACIÓN***
  2. ***MENÚ DE COMPRA***
  3. ***SALIR***

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

EJERCICIOS

- **MENU DE ADMINISTRACIÓN:** Se visualizará el siguiente submenú:
  1. Introducir productos en la lista: Pediremos los datos de un producto al usuario y lo introducimos en el HashMap.
  2. Visualizar todos los productos.
  3. Eliminar productos de la lista: Pediremos el código del producto y lo eliminaremos.
  4. Volver al menú principal.

EJERCICIOS

- **MENU DE COMPRA:**
  1. Comprar productos:
     - Mostramos una lista con los productos a comprar.
     - El usuario elegirá que producto comprar (el código) y luego le preguntaremos cuantas unidades desea de él. Luego se le preguntará si desea comprar otro producto o salir.
     - Por último, se le mostrará el importe total de la compra.
     - Date cuenta de que necesitarás actualizar el valor del stock de un producto cuando el usuario lo compre. En caso de que el usuario pida más unidades de las que quedan se le avisará por pantalla del error, se le comunicarán las unidades restantes y le preguntará si desea comprar otro producto.
  2. Volver al menú principal.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 03.- (OBLIGATORIO):** Diseña un programa en Java para gestionar los autobuses de la estación de Plasencia.
- Nuestro programa dispondrá de un vector de 6 celdas dónde "aparcar" los autobuses.
- De cada autobús, almacenaremos su matrícula y los conductores que tiene asignados (los cuales se almacenarán en un HashMap).
- De cada conductor almacenaremos su DNI (que hará las veces de clave) y su nombre.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- El programa dispondrá del siguiente menú:
  - Aparcar (pedirá un número, que será la posición del vector donde deberemos aparcar el autobús. Si la posición está ocupada, se volverá a pedir hasta encontrar una libre).
  - Mostrar dársenas libres.
  - Buscar autobús (método que muestre toda la información del autobús a partir de su matrícula).
  - Buscar conductor (Mostrará la matrícula del autobús que tiene asignado).
  - Método que retorne la posición del vector donde se encuentra el autobús con mayor número de conductores asignados.

EJERCICIOS

- **Ejercicio 04.- (<u>OPTATIVO</u>):** Crea un programa con dos conjuntos distintos: HashSet y LinkedHashSet. Inserta en ambos los meses de un año. Posteriormente recorre los conjuntos mostrando su contenido. Estudia sus diferencias.
- Pista: Necesitarás un vector de String para almacenar los meses.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 05.- (OPTATIVO):** Crea un programa con dos mapas distintos: TreeMap y LinkedHashMap. Inserta en ambos los meses de un año del revés asociándoles a cada uno su correspondiente número entero ([12, Diciembre], [11, Noviembre], [10, Octubre], … [1, Enero]) . Posteriormente recorre los conjuntos mostrando su contenido. Estudia sus diferencias.
- Pista: Necesitarás un vector de String para almacenar los meses.

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 06.- (OBLIGATORIO):** Debido a la avalancha de campañas de donaciones para la compra de material para hospitales y residencias contra el coronavirus, el Ministerio de Sanidad ha decidido crear un software que gestione todo el dinero donado. En la versión alfa del software debemos gestionar las donaciones de una única campaña. Para ello, debemos implementar las siguientes clases:
  - Clase *Donacion*, que almacenará el nombre de la persona que dona y la cantidad donada.
  - Clase *Campania*, que almacenará un conjunto de donaciones y el nombre de la campaña.

EJERCICIOS

- El programa mostrará un menú con las siguientes opciones:

1. Añadir donación.
2. Mostrar donaciones.
3. Mostrar donaciones por nombre de donante (Pediremos un nombre y mostraremos las donaciones cuyo donante coincida con el nombre dado).
4. Mostrar número de donaciones.
5. Mostrar total dinero recaudado.
6. Ordenar donaciones (de mayor a menor importe de las donaciones)
7. Salir.

NOTA: No se puede utilizar el método de la Burbuja para realizar la ordenación de las donaciones.
PISTA: Podéis utilizar un vector de Donaciones donde ir insertándolas de manera ordenada.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 07.- (OBLIGATORIO):** Vamos a crear ahora la versión beta del software anterior. En esta versión, debemos gestionar las donaciones de varias campañas (no sabemos cuántas habrá). Opciones de menú:

1. Añadir campaña (En esta opción SÓLO pediremos el nombre de la campaña).
2. Añadir donación (Las insertará de una en una y preguntaremos al usuario el nombre de la campaña donde insertar la donación).
3. Mostrar campañas junto con donaciones.
4. Mostrar campaña por nombre (Pediremos un nombre y mostraremos la campaña junto con las donaciones).
5. Mostrar total dinero recaudado.
6. Mostrar mayor donación.
7. Salir.

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas

EJERCICIOS

- **Ejercicio 08.- (OBLIGATORIO):** Una importante empresa multinacional nos ha solicitado un programa para gestionar las distintas sedes que tiene repartidas a lo largo del mundo. Para ello, dispondremos de un arrayList de ciudades. De cada ciudad, almacenaremos su nombre y un conjunto de sedes (elige el tipo de conjunto que prefieras).
- De cada sede almacenaremos el nombre de la sede y sus ingresos anuales.

EJERCICIOS

- Implementa las siguientes opciones para el software:
  - Añadir una ciudad (al menos pediremos los datos de una sede. Después de añadir cada sede, preguntaremos al usuario si desea seguir añadiendo sedes).
  - Mostrar todas las ciudades junto con sus sedes.
  - Método que muestre el nombre de las sedes cuyos ingresos anuales son superiores a la media.
  - Buscar por nombre de sede. (El método retornará un booleano).
  - Añadir sede (pediremos el nombre de la ciudad y, si ésta existe en el arrayList, pediremos los datos de la nueva sede y los insertaremos).
  - Mostrar todas las sedes ordenadas de mayor a menor número de ingresos anuales (PISTA: Utiliza una estructura de datos donde ir añadiendo todas las sedes de manera ordenada).

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 09.- (<u>OBLIGATORIO</u>):** El operador de loterías "*TuLotero*" desea gestionar los sorteos que realiza a lo largo del año.
- Para tratar esta información, necesitaremos una clase *Sorteo*, cuyos atributos serán la fecha del sorteo (será un LocalDate) y un vector de 4 celdas con los números ganadores (aleatorios de 1 a 100).
- Los sorteos se organizarán en un .TreeMap, cuyas claves serán las fechas del sorteo.

EJERCICIOS

- Facilitaremos las siguientes opciones al operador de loterías:
  - Realizar sorteo (La fecha será la de hoy y los números ganadores serán aleatorios entre 1 y 100).
  - Repetir sorteo. Se eliminará el sorteo de hoy y se volverá a realizar el sorteo.
  - Mostrar sorteos del mes actual.
  - Mostrar sorteo dada una fecha.
  - Realizar sorteo por fecha (Pediremos una fecha y se realizará el sorteo de esa fecha en concreto - comprobando, previamente, que no hay sorteo en esa fecha).
  - Mostrar todos los sorteos.

EJERCICIOS

- **Ejercicio 10.- (<u>OBLIGATORIO</u>):** Implementa un programa en el que crees la clase Persona con los atributos nombre y edad.
- Dicha clase deberá implementar la interface Comparable, de manera que los elementos se ordenen de menor a mayor edad.
- Utiliza el main que se te adjunta como recurso y compara los resultados.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 11.- (OBLIGATORIO):** Implementa un programa en el que crees la clase Alumno con los atributos dni (String), número de expediente (int) y nota media (float).
- Tu programa deberá mostrar las siguientes opciones:
  - Añadir alumno.
  - Mostrar alumnos (ordenados de menor a mayor número de expediente).
  - Buscar por número de expediente.
  - Mostrar alumnos por nota.
- Utiliza como ED un conjunto.

Oscar Laguna García
Ana M. Arribas Arjona
Última versión: Á. Enrique Pineda Navas