

Ejercicios con Map.

Ejercicio 1.

Escribe un método *static* para calcular la frecuencia absoluta de aparición de caracteres en un *String*, número de veces que aparecen los caracteres en un *String*. Para ello define un *Map* cuyas claves son de tipo *Character* y cuyos valores son de tipo *Integer*. El método recibirá un *String* y devolverá un objeto de tipo *Map<Character, Integer>*.

Ejercicio 2.

Supongamos que tenemos una lista de *String* que denominamos palabras y se quiere obtener un índice de las posiciones que ocupan las cadenas de la lista palabras mediante una lista de enteros. El argumento de entrada será por tanto de tipo *List<String>* y el argumento de salida un *Map<String, List<Integer>>*. Por ejemplo, para una lista que contuviera las palabras de la cadena “la palabra que más aparece en este texto es la palabra palabra”, la salida sería

```
{más=[3], texto=[7], aparece=[4], palabra=[1, 10, 11], la=[0, 9], en=[5], que=[2], este=[6], es=[8]}
```

Indicando que ‘la’ está en las posiciones 0 y 9, ‘palabra’ en la 1, 10 y 11, etc.

Ejercicio 3.

Vamos a añadir al tipo **Aeropuerto** del ejercicio de Set, las siguientes funcionalidades:

- Un método que devuelva una estructura que relacione cada fecha con todos los vuelos programados para ese día.
- Un método que devuelva una estructura que cuente el número de pasajeros para cada día.
- Un método que devuelva una estructura que acumule la recaudación por destino.
- Un método que devuelva el día (una lista por si hubiera empates) que más pasajeros hubo. Invertir el Map y usar SortedMap.
- Un método que devuelva el destino (una lista por si hubiera empates) con más recaudación.

Ejercicio 4.

Vamos a ampliar el tipo **ListaPalabras** del ejercicio 2 con los siguientes métodos:

- Map<Character, Double> getFrecuenciaRelativaCaracteres();*
- SortedMap<Double, List<Character>> getCaracteresMasFrecuentes(Double p);*
- List<Character> getCaracterMasRepetido();*
- List<Character> getCaracterMasVecesInicial();*

Cuyas funcionalidades son:

- Devuelve un *Map* donde a cada carácter le corresponde su frecuencia relativa (número de veces que aparece dividido el número total de caracteres del texto)
- Dado un número real p devuelve un *SortedMap* con aquellos caracteres cuya frecuencia relativa es mayor que p.
- ¿Cuál es el carácter más repetido?
- ¿Cuál es el carácter por el que empiezan más palabras?

Proporciona una clase a **ListaPalabra** que tenga estos atributos para gestionar la información:

private List<String> texto;

// Map donde a cada carácter le corresponde el conjunto de palabras que empiezan por esa inicial.

private Map<Character, Set<String>> mapIniciales;

// Map donde a cada carácter le corresponde el número de veces que aparece en el texto.

private Map<Character, Integer> mapContador;