

Expresiones Stream más comunes Repaso:

Obtengo la suma de todos los sueldos, un double

```
empleados.stream.mapToDouble(empleado -> empleado.getSueldo)
.sum() //suma todos los sueldos
```

Aumento sueldo a todos aplicando una funcion

```
List<Empleado> empleadosConAumento =
empleados.stream.map( empleado -> empleado.aumento(monto))
.toList()
//map siempre depende del resultado de una funcion, es decir q mi
metodo empleado.aumento debera aplicar el aumento y devolverse asi
mismo con el this (el empleado)

public Empleado aumento(double monto){
    this.setSueldo(this.getSueldo+ monto)
    return this
}
```

Obtengo los 3 primeros que se llaman con el nombre ingresado

```
empleados.stream.filter( nombre -> nombre.equals(nombreIngresado)
.limit(3) //para q arroje los 3 primeros
.sorted((empleado1,empleado2) ->
empleado1.getNombre.compareTo(empleado2.getNombre)
.toList()); // recordar que el metodo devuelva List y no ArrayList
//obtiene los q se llaman como nombreIngresado, primeros 3 y
ordenados alfabéticamente
```

Algo parecido

```
empleados.stream.filter( empleado -> empleado.getNombre()
.contains(nombreIngresado))
.findFirst() //devolvera solo el primero
```

El flatmap aplana lista de listas

```
List<Cancion> canciones =
    autores.stream()
        .filter(autor -> autor.getNombre()
            .toUpperCase()
            .contains(nombreAutor.toUpperCase()))
        .flatMap(autor -> autor.buscar().stream())//Para no obtener una
lista de listas debo usar el flatmap!!!!
        .toList();
```

