Ejercicios 03

1º) Grupo de alumnos con n notas. Pasan a siguiente curso los que tengan menos de dos suspensas. Debes mostrar, al final, los nombres de los alumnos que hayan aprobado y el nombre de las materias que han suspendido, cada uno de ellos.

2º) Calcular e imprimir el promedio de 50 valores aleatorio almacenados en un Lista. Determinar cuántos son mayores que el promedio e imprimirlos.

3º) Almacenar 300 números en un Lista, imprimir cuántos son ceros, cuántos son negativos, cuántos positivos. Imprimir además la suma de los negativos y la suma de los positivos.

4º) Se pide implementar una lista que gestione la lista de espera de un taller mecánico. El taller tendrá una lista de fichas de vehículos que serán dados de alta en el momento de su llegada al taller, donde se le asignará la hora de entrada. Una vez diagnosticado el problema del coche se le asignará una fecha de resolución de la avería y sólo cuando el técnico venga a buscarlo, se le dará una fecha de salida y el borrado de la lista de espera. Es recomendable crear tres listas, una para los coches en el momento de llegada, otra para los no arreglados diagnosticados, otra para los arreglados y están pendientes de retirar.

5º Listar el nombre de los alumnos con una materia x suspensa.

**Ejercicio 1 — Filtrado y ordenación**

Devuelve una lista con los alumnos cuya nota media sea superior a 7.5, ordenados de mayor a menor nota.

* Usa **stream()**, **filter()** y **sorted()**.
* La lista original no debe modificarse.

**Ejercicio 3 — Búsqueda y transformación**

Obtén una lista con los nombres de los tres alumnos con mayor nota media.

* Usa **map()** para quedarte solo con el nombre.
* Limita el resultado a tres elementos con **limit()**.
* **Ejercicio 4 — Búsqueda parcial**

Devuelve una lista con los alumnos cuyo nombre contenga una cadena dada (sin distinguir mayúsculas/minúsculas).

Usa **filter()** y **toLowerCase()**.

El método recibirá el texto a buscar como parámetro.

* **Ejercicio 5 — Estadística simple**

Calcula la nota media global de todos los alumnos del repositorio.

Usa **mapToDouble(Alumno::getMedia)** y **average()**.

Devuelve 0.0 si la lista está vacía.