

## TP - Colecciones en JAVA

1. Cree un `ArrayList<Integer>`, que permita asignar los valores numéricos manualmente (los que tú quieras) hasta ingresar un valor menor a 0, mostrar los números por pantalla. Solicite los números mediante un bucle.
2. Codifique una aplicación que:
  - a) lea 20 números decimales ingresados por teclado por el usuario y los coloque en un `ArrayList<Double>`.
  - b) determine y muestre el mayor de los números en el arreglo
  - c) determine y muestre el menor de los números del arreglo calcule y muestre el rango (diferencia entre el mayor y el menor) de los elementos en el arreglo
3. Codifique una aplicación para almacenar en un `ArrayList<Integer>` aleatoriamente 20 números enteros positivos pares del 1 al 100, de los cuales se desea saber:
  - a) su promedio aritmético
  - b) cuántos de los números son iguales al promedio aritmético
  - c) cuántos de los números son mayores que el promedio aritmético
  - d) cuántos de los números son menores que el promedio aritmético
4. Codifique una aplicación que permita almacenar palabras en un `ArrayList<String>` hasta que se ingrese la palabra "salir", al finalizar muestre por pantalla:
  - a) Las palabras ingresadas
  - b) Indicar cuál de las palabras ingresadas posee más caracteres
  - c) Indicar si se ingresaron o no palabras iguales o repetidas
  - d) Mostrar las palabras repetidas ingresadas y cuantas veces se repitió esa palabra
5. Realice una búsqueda secuencial en un `ArrayList<Integer>` que contenga 50 elementos generados aleatoriamente mediante la función java **random**.

Realice una búsqueda secuencial de la siguiente forma:

- se lee el valor que se desea buscar,
- se compara la primera posición;
- si son iguales,
- fin de la búsqueda.
- De lo contrario, compararlo con la segunda posición, y así sucesivamente.
- Si se llega al final del arreglo y no se encontró el valor, debe indicarlo con un mensaje apropiado.
- Si se encuentra, se debe especificar la posición que ese valor ocupa en la lista por primera vez.

6. Codifique una aplicación que solicite y cargue en una `ArrayList<Integer>` 10 dígitos enteros, luego cree dos nuevas `ArrayList<Integer>` y asigne a la primera los números ingresados por el usuario de forma ascendente y en la segunda de forma descendente. Muestre las 2 listas por pantalla. Investigue como ordenar una Lista(`List<>`)

7. Codifique la clase celda con los atributos:

```
int fila;  
int column;  
String valor;
```

- Crea una clase Matriz que contenga una variable global `ArrayList<Celda>`

```
ArrayList<Celda> matrizCuadrada = new ArrayList<Celda>();
```

- Codifique un programa que solicite al usuario un valor para la celda y que solicite la posición donde se desea almacenar el valor, cree una instancia de la clase Celda, asigne los valores cargados por el usuario y agregue la instancia a la lista `matrizCuadrada`; repita este proceso hasta que el usuario ingrese como valor la cadena "FIN".
- Muestre por pantalla los valores cargados en `matrizCuadrada`.
- Codifique un método que reciba como parámetro los valores fila y columna y retorne el valor almacenado, en caso de que la fila y la columna no exista retorne el mensaje "La fila y columna indicada no ha sido asignada"