

# Introducción a Sistemas Complejos, JAVA, MVN y GIT

Nombre estudiante: *Michael Jefferson Ballesteros Coca*

Curso: *Arquitecturas Empresariales* – Docente: *Luis Daniel Benavides Navarro*

Fecha de entrega: *Agosto 13, 2020*

## Requerimientos del Programa.

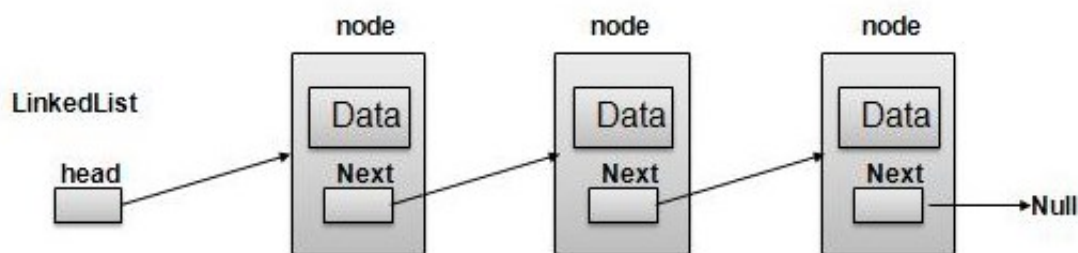
- (a) El programa lee  $n$  numeros reales desde un archivo
- (b) Use una LinkedList (Lista encadenada) para guardar los  $n$  numeros para los cálculos. **Use su propia implementación**

## Generalidades.

**LinkedList.** Las LinkedLists son estructuras de datos comunes donde se guardan datos, se implementan con apuntadores, que permiten encadenar a partir de direcciones de memoria los datos que se usan. Las LinkedLists poseen 2 componentes principales:

- Cabeza de la lista
- Nodos

Existen varias opciones para crear esta estructura, para este repositorio, la LinkedList posee una cabeza, donde tiene 2 apuntadores, uno hacia el inicio de la LinkedList y otro hacia el ultimo nodo de la estructura.



También tiene diferentes funcionalidades que fueron implementadas, entre ellas:

- add()
- remove()
- size()
- toArray()

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + a_2 + a_3 + \dots + a_n}{n}$$

### Media.

Cuando buscamos la media de un conjunto de datos, ubicamos la posición del centro de estos a través del promedio de estos.

La clase Calculator en el repositorio posee un método ( calculateMean() )

```
public static Double calculateMean(Double[] array){
    Double sum = 0d;
    int n = array.length;
    for(Double x: array) {
        sum += x;
    }
    return sum/n;
}
```

Donde, se usó un ciclo For para recorrer los datos y se lleva una variable (sum) para realizar la suma de los datos; se concluye después de recorrer los datos, la media, calculando con el valor n de la cantidad de datos que existen y la suma de todos los datos.

$$S = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}}$$

### Desviación Estándar.

La desviación estándar cuantifica la variación de la población, aunque parecería más robusta con respecto a el promedio si se calculara a papel y lápiz, podría ser un procedimiento complejo.

Se implemento un método en la clase Calculator para calcular la desviación estándar ( calculateDeviation() )

```
public static Double calculateDeviation(Double[] array, double mean){
    Double sumax = 0d;
    int n = array.length;
    for(Double x: array){
        sumax+=Math.pow(x-mean, 2);
    }
    return Math.sqrt((sumax/(n-1)));
}
```

Esta vez, usamos el ciclo For para determinar la parte interna de la raíz, donde se llevó una variable (sumax) donde se hacia el cálculo entre el promedio y cada dato del conjunto de datos. El método finaliza retornando la raíz cuadrada de la parte interna de la raíz, usando la librería Math de Java.

$$\text{Absolute Error} = |v_A - v_E|$$

**Error Absoluto.**

Para poder comprobar nuestros cálculos, necesitamos usar el error absoluto y tener un margen de tolerancia, ya que con este podemos decir si el dato resultante es confiable o no.

```
private static Double TOLERANCE=0.1d;

@Test
public void shouldCalculateMean(){
    String file = "src\\test\\resources\\data\\data1.txt";
    LinkedList list = new LinkedList();
    Calculator.readFile(file, list);
    Double errorAbsoluto=Math.abs(Calculator.calculateMean(list.toArray())-
                                550.6);
    assertTrue(errorAbsoluto < TOLERANCE);
}
```

Para este caso de prueba, previamente se ha declarado la variable ( TOLERANCE ) para después compararla con el error absoluto. Usamos el archivo data1.txt que posee un conjunto de datos a operar, guardamos estos datos en la LinkedList para luego determinar la media de estos datos. En la variable ( errorAbsoluto ) la librería Math se encarga de darnos el valor absoluto entre el valor que se calculó a través de la aplicación y el número que se supone previamente debería dar , después de hacer el cálculo. Con una tolerancia de 0.1, usamos la variable ( errorAbsoluto ) para determinar si el error es menor a la tolerancia que tenemos.

**Conclusión.** Aunque desde el principio, el taller parecía bastante sencillo, fue un verdadero reto cuando se tuvo que enfatizar en que lo que importaba era la arquitectura del repositorio; y sin dejar de lado la ejecución del programa, se dio a conocer de manera clara y concisa la estructura del repositorio y sus respectivas componentes.