



Materia: Estructura de Datos y Algoritmia.

Maestro: Víctor Manuel Navarro Bustos.

Alumno: Claudio Lara Jasso.

Registro: 22310545.

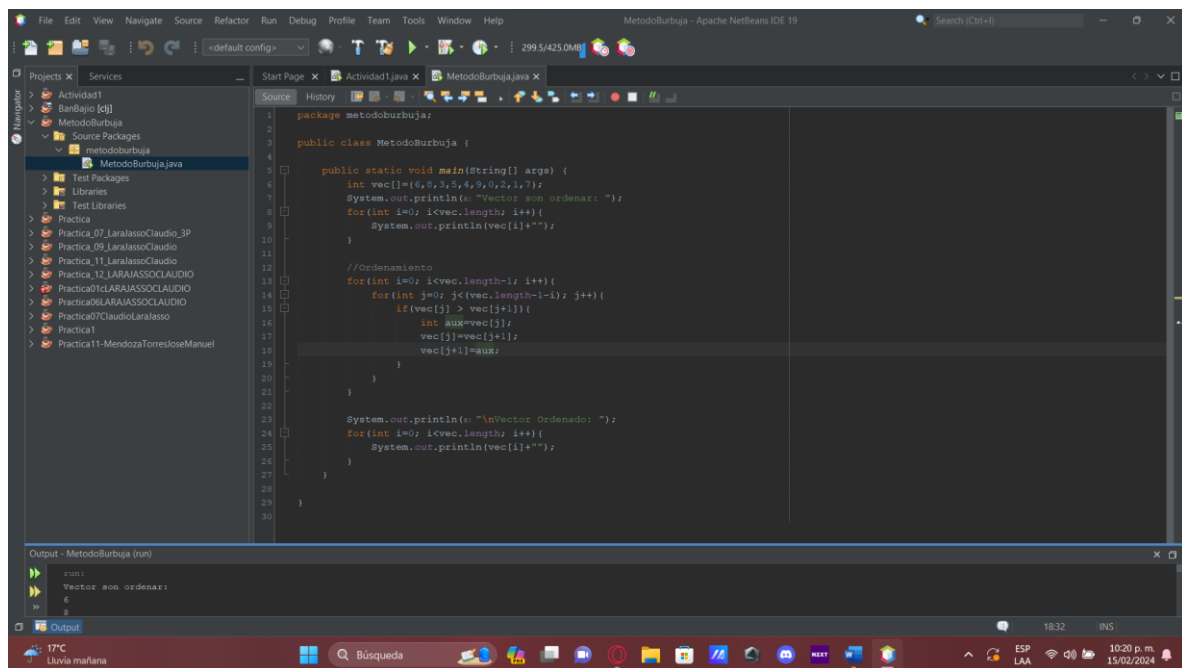
Fecha: 15/02/2024.

Tarea: Métodos de Ordenamiento.

1.- Método de la burbuja.

La ordenación por el método de la burbuja que en ingles se denomina Bubble sort, es un algoritmo de ordenamiento muy sencillo que funciona revisando cada elemento de la list que se está ordenando e intercambiándolo con otra posición, de tal manera que el mas grande o pequeño según sea el orden va subiendo, de ahí el nombre.

En otras palabras, es un método de ordenamiento de datos usado en vectores, funciona revisando cada elemento de un vector con el elemento siguiente cambiando de posición si están en orden equivocado. Su funcionamiento es acomodar el valor menor en la posición 0 del vector y el valor mayor en la posición final del vector comparando valor por valor de cada elemento de un vector.



```
package metodoburbuja;

public class MetodoBurbuja {

    public static void main(String[] args) {
        int vec[]={6,8,3,5,4,3,0,2,1,7};
        System.out.println("Vector non ordenar: ");
        for(int i=0; i<vec.length; i++){
            System.out.println(vec[i]+"");
        }

        //Ordenamiento
        for(int i=0; i<vec.length-1; i++){
            for(int j=0; j<(vec.length-1-i); j++){
                if(vec[j] > vec[j+1]){
                    int aux=vec[j];
                    vec[j]=vec[j+1];
                    vec[j+1]=aux;
                }
            }
        }

        System.out.println("\nVector Ordenado: ");
        for(int i=0; i<vec.length; i++){
            System.out.println(vec[i]+"");
        }
    }
}
```

Output - MetodoBurbuja (run)

```
run:
Vector non ordenar:
6
8
3
5
4
3
0
2
1
7

Vector Ordenado:
0
1
2
3
3
4
5
6
7
8
```

Imagen de código en lenguaje JAVA en NetBeans.

En el código observamos que el vector queda ordenado en la segunda iteración, sin embargo el algoritmo compara numero con numero hasta terminar los ciclos, aun cuando ya está ordenado el vector.

2.- Método de Selección.

El ordenamiento por selección que en ingles se dice selection Sort es un algoritmo de ordenamiento que busca el mínimo elemento de la lista, lo intercambia con el primero, buscar el siguiente mínimo en el resto de la lista y lo intercambia con el segundo hasta llegar al final de la lista.

El algoritmo de ordenación por selección recorre todo el arreglo desde la primera posición de buscando el menor elemento de todos. Cuando termina, lo reemplaza por el elemento de la primera posición y repite todo el proceso con el segundo elemento y así sucesivamente.

Ejemplo:

Tenemos el vector original con los siguientes valores:

| | | | | |
|----|----|----|----|----|
| 45 | 17 | 23 | 67 | 21 |
|----|----|----|----|----|

Iteración 1:

| | | | | |
|----|----|----|----|----|
| 45 | 17 | 23 | 67 | 21 |
|----|----|----|----|----|

Se selecciona el numero 17 y se intercambia por el 45.

Iteración 2:

| | | | | |
|----|----|----|----|----|
| 17 | 45 | 23 | 67 | 21 |
|----|----|----|----|----|

Se selecciona el 21 y se intercambia por el 45.

Iteración 3:

| | | | | |
|----|----|----|----|----|
| 17 | 21 | 23 | 67 | 45 |
|----|----|----|----|----|

Se selecciona el 23, no hay cambio.

Iteración 4:

| | | | | |
|----|----|----|----|----|
| 17 | 21 | 23 | 67 | 45 |
|----|----|----|----|----|

Se selecciona el 45 y se intercambia por el 67

Finalmente, vector ordenado:

| | | | | |
|----|----|----|----|----|
| 17 | 21 | 23 | 45 | 67 |
|----|----|----|----|----|

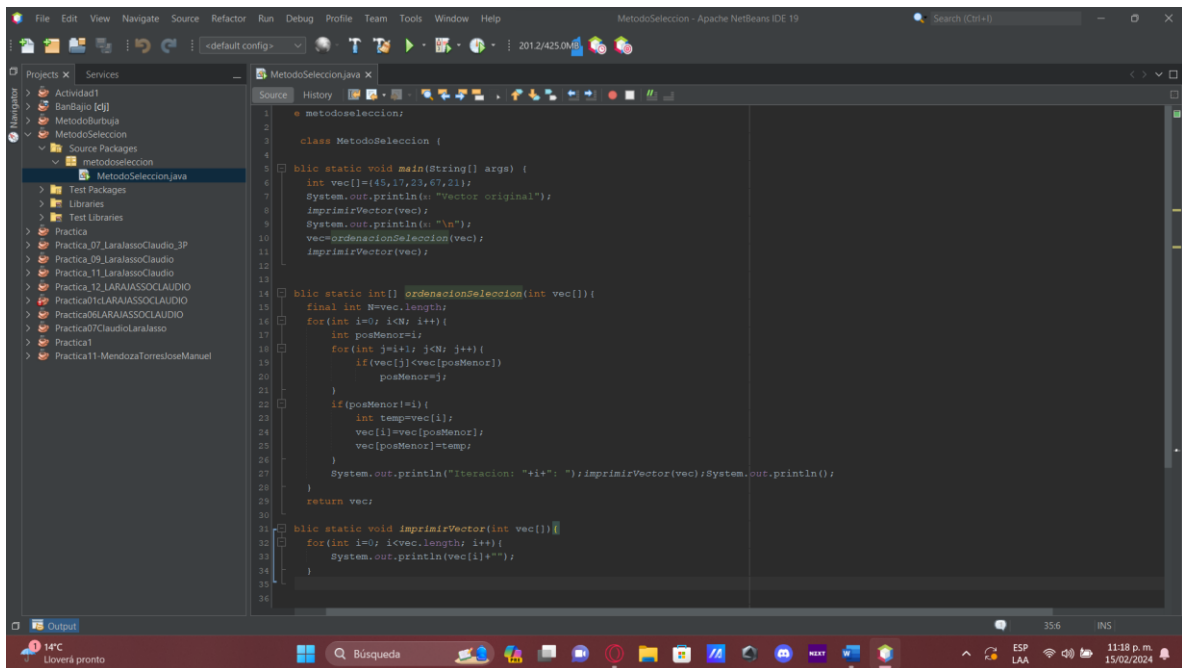


Imagen de código en lenguaje JAVA en Netbeans.

3.- Método por inserción.

El ordenamiento por inserción o insertionSort en ingles empieza por un elemento, el cual hace la suposición de que la lista está ordenada, se toma los elementos $k+1$ y se compara con todos los elementos ya ordenados, se detiene cuando se encuentra un elemento menor (todos los elementos mayores han sido desplazados una posición a la derecha) o cuando ya no se encuentran elementos.

Este algoritmo ordena los dos primeros dos elementos del vector, a continuación, toma el tercer elemento y lo ordena en la posición correspondiente. Cuando termina de ubicar un elemento en su posición correcta, continúa con el siguiente elemento, hasta hacer este proceso con todos los elementos del vector.

Ejemplo:

Primeramente, tenemos el vector desordenado:

| | | | | |
|----|----|----|----|----|
| 45 | 17 | 23 | 67 | 21 |
|----|----|----|----|----|

Iteración 1:

Se ordenan los primeros dos elementos.

| | | | | |
|----|----|----|----|----|
| 17 | 45 | 23 | 67 | 21 |
|----|----|----|----|----|

Iteración 2:

Se ordenan los elementos 3 y 4.

| | | | | |
|----|----|----|----|----|
| 17 | 23 | 45 | 67 | 21 |
|----|----|----|----|----|

Se toma el tercer elemento 23 y se ordena el primer elemento y el segundo recorriéndose el 45 a la tercera posición

Iteración 3:

| | | | | |
|----|----|----|----|----|
| 17 | 23 | 45 | 67 | 21 |
|----|----|----|----|----|

No hay cambio ya que el 67 es mayor al 45.

Iteración 4:

| | | | | |
|----|----|----|----|----|
| 17 | 23 | 45 | 67 | 21 |
|----|----|----|----|----|

Se toma el número 21, y se mete en la posición correspondiente del vector, recorriendo los números mayores.

| | | | | |
|----|----|----|----|----|
| 17 | 21 | 23 | 45 | 67 |
|----|----|----|----|----|

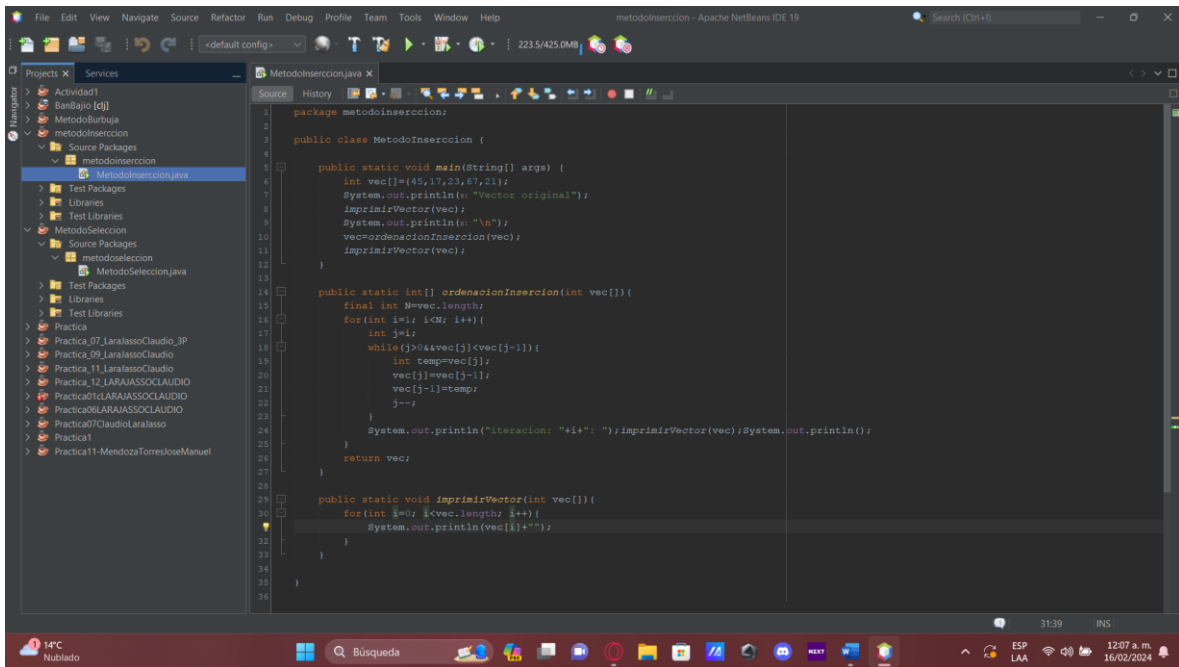


Imagen de código en lenguaje JAVA en Netbeans.

Conclusión:

Estas tres metodologías de ordenamiento, burbuja, selección e inserción son algoritmos simples y fáciles de entender. Sin embargo, su eficiencia es limitada en comparación con algoritmos más avanzados como QuickSort o MergeSort, especialmente cuando se trata de conjuntos de datos grandes.

La metodología de ordenamiento burbuja, aunque fácil de implementar, tiene un rendimiento pobre en términos de tiempo de ejecución, especialmente en conjuntos de datos grandes debido a su complejidad cuadrática $O(n^2)$. La metodología de selección mejora un poco la eficiencia al reducir el número de intercambios, pero aun así tiene un rendimiento $O(n^2)$ en el peor de los casos. Por otro lado, la metodología de inserción, aunque también cuadrática en el peor de los casos, puede ser más eficiente en conjuntos de datos pequeños o casi ordenados.