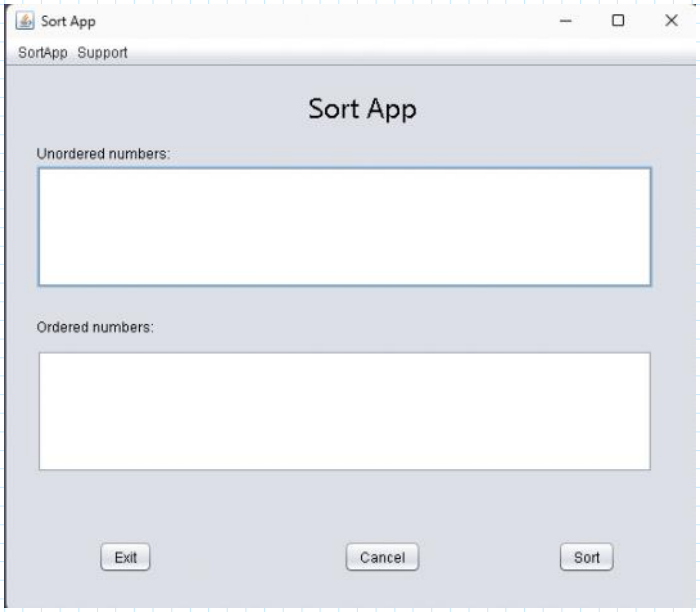
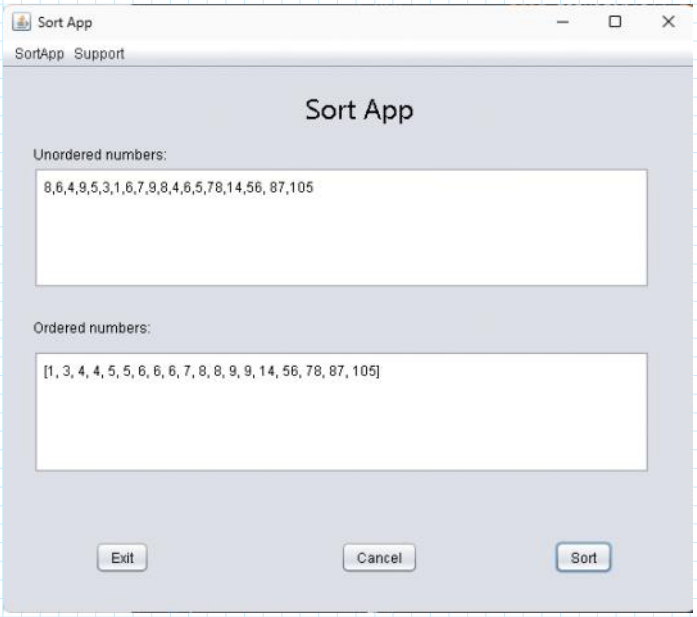
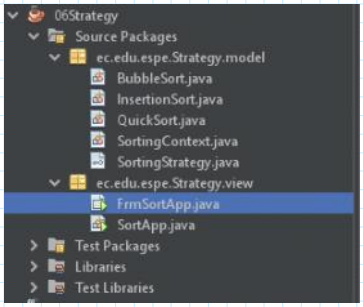
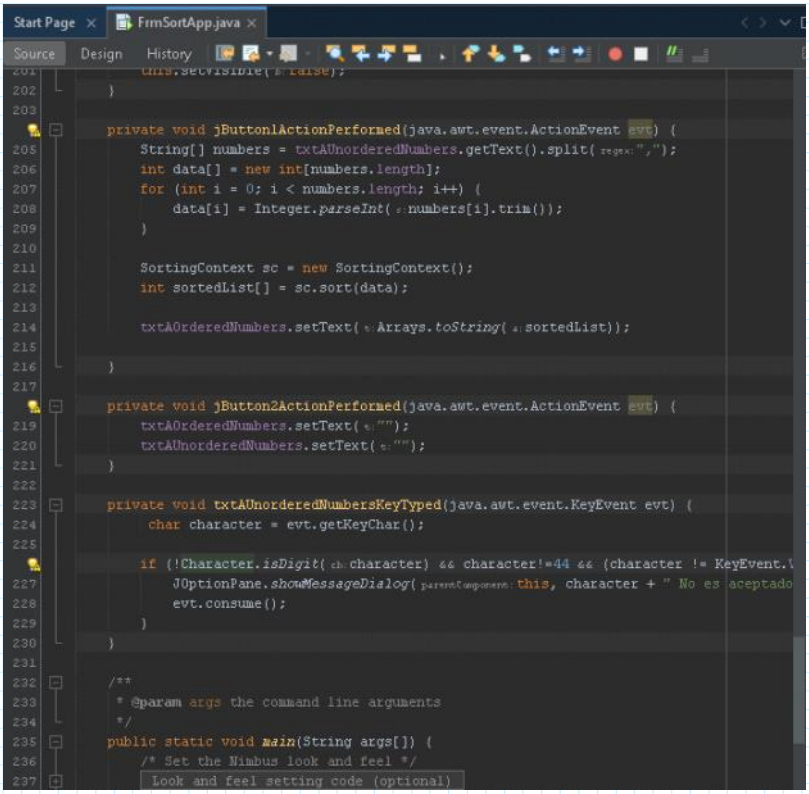
hw23StrategyGUI

domingo, 19 de febrero de 2023 17:05

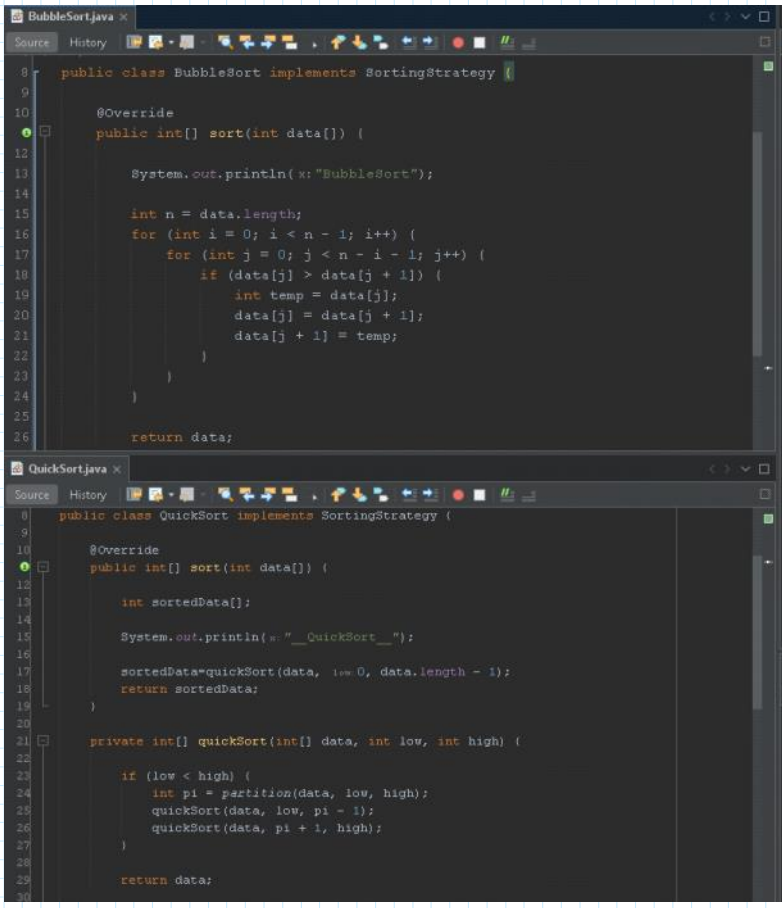






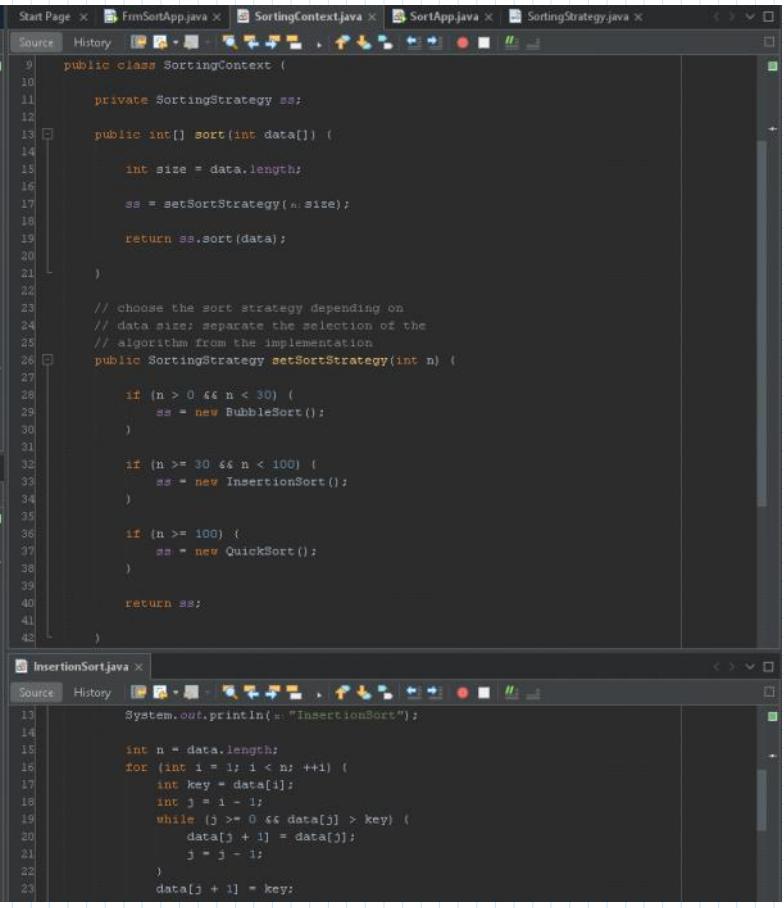


```
Start Page | FrmSortApp.java
Source | Design | History
        this.setVisible(false);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String[] numbers = txtAUnorderedNumbers.getText().split(regex:",");
        int data[] = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
            data[i] = Integer.parseInt(s:numbers[i].trim());
        }

        SortingContext sc = new SortingContext();
        int sortedList[] = sc.sort(data);

        txtAOrderedNumbers.setText(t:Arrays.toString(a:sortedList));

    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        txtAOrderedNumbers.setText(t:"");
        txtAUnorderedNumbers.setText(t:"");
    }

    private void txtAUnorderedNumbersKeyTyped(java.awt.event.KeyEvent evt) {
        char character = evt.getKeyChar();

        if (!Character.isDigit(ch:character) && character!=44 && (character != KeyEvent.V
            JOptionPane.showMessageDialog(parentComponent:this, character + " No es aceptado
            evt.consume();
        }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)
```



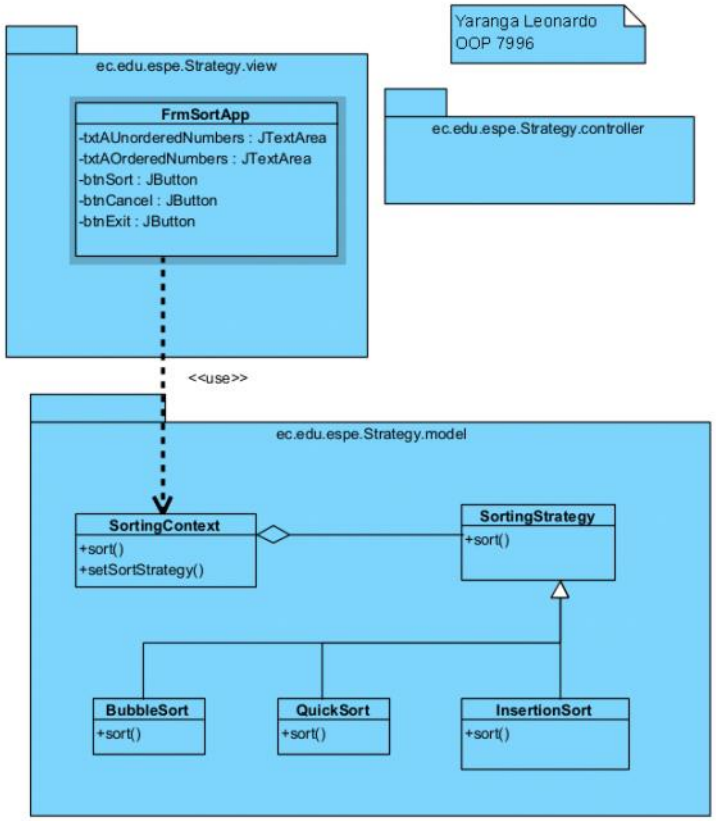
```
BubbleSort.java
Source | History
public class BubbleSort implements SortingStrategy {

    @Override
    public int[] sort(int data[]) {

        System.out.println(x:"BubbleSort");

        int n = data.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (data[j] > data[j + 1]) {
                    int temp = data[j];
                    data[j] = data[j + 1];
                    data[j + 1] = temp;
                }
            }
        }

        return data;
```

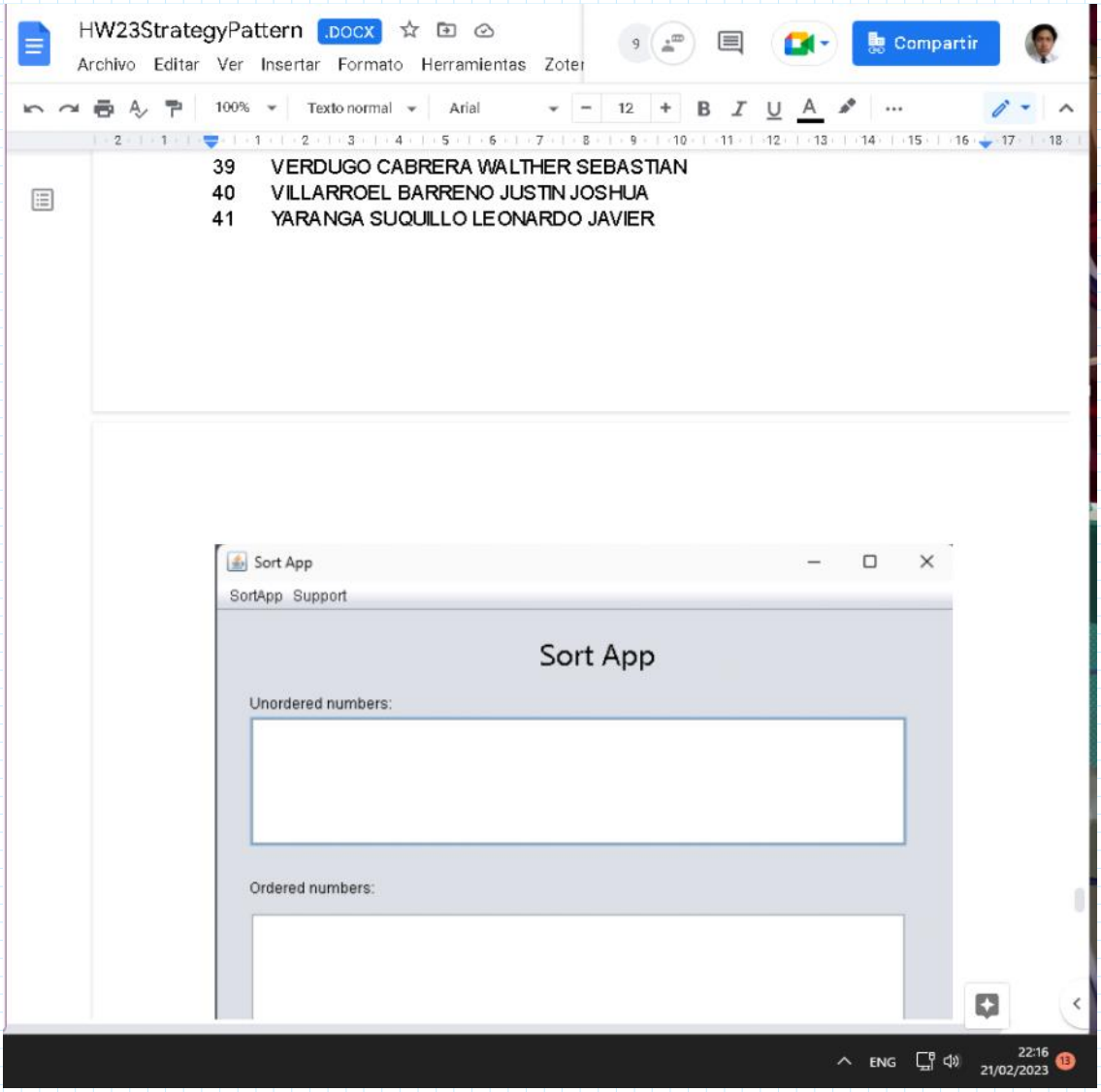
```
QuickSort.java
Source | History
public class QuickSort implements SortingStrategy {

    @Override
    public int[] sort(int data[]) {

        int sortedData[];

        System.out.println(x:"__QuickSort__");

        sortedData=quickSort(data, low:0, data.length - 1);
        return sortedData;
    }

    private int[] quickSort(int[] data, int low, int high) {

        if (low < high) {
            int pi = partition(data, low, high);
            quickSort(data, low, pi - 1);
            quickSort(data, pi + 1, high);
        }

        return data;
```



```
Start Page | FrmSortApp.java | SortingContext.java | SortApp.java | SortingStrategy.java
Source | History
public class SortingContext {

    private SortingStrategy ss;

    public int[] sort(int data[]) {

        int size = data.length;

        ss = setSortStrategy(n:size);

        return ss.sort(data);

    }

    // choose the sort strategy depending on
    // data size; separate the selection of the
    // algorithm from the implementation
    public SortingStrategy setSortStrategy(int n) {

        if (n > 0 && n < 30) {
            ss = new BubbleSort();
        }

        if (n >= 30 && n < 100) {
            ss = new InsertionSort();
        }

        if (n >= 100) {
            ss = new QuickSort();
        }

        return ss;

    }
```

```
InsertionSort.java
Source | History
        System.out.println(x:"InsertionSort");

        int n = data.length;
        for (int i = 1; i < n; ++i) {
            int key = data[i];
            int j = i - 1;
            while (j >= 0 && data[j] > key) {
                data[j + 1] = data[j];
                j = j - 1;
            }
            data[j + 1] = key;
```


Yaranga Leonardo
OOP 7996
ec.edu.espe.Strategy.view
FrmSortApp
-txtAUnorderedNumbers : JTextArea
-txtAOrderedNumbers : JTextArea
-btnSort : JButton
-btnCancel : JButton
-btnExit : JButton
ec.edu.espe.Strategy.controller
<<use>>
ec.edu.espe.Strategy.model
SortingContext
+sort()
+setSortStrategy()
SortingStrategy
+sort()
BubbleSort
+sort()
QuickSort
+sort()
InsertionSort
+sort()


HW23StrategyPattern .DOCX
Archivo Editar Ver Insertar Formato Herramientas
Compartir
39 VERDUGO CABRERA WALTHER SEBASTIAN
40 VILLARROEL BARRENO JUSTIN JOSHUA
41 YARANGA SUQUILLO LEONARDO JAVIER
Sort App
SortApp Support
Sort App
Unordered numbers:
Ordered numbers:
ENG
22:16
21/02/2023