Código Fuente

```java
package Graphs;
/**
 * @author Victor Lavalle
 */
import javax.swing.JFrame;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;
import Patterns.observer.Observer;
import org.jfree.data.category.CategoryDataset;

public class Bars extends JFrame implements Observer{

    private CategoryDataset createDataset(String[] nominees, int[ ]votes){
       final  DefaultCategoryDataset result = new DefaultCategoryDataset();
         for(int i=0;i<nominees.length;i++){
                 result.addValue(votes[i], nominees[i],"");
         }
        return result;
    }
    public void Graph(ChartPanel chartpanel){
        chartpanel.setPreferredSize(new  java.awt.Dimension(500,500));
        this.setSize(400, 300);
        this.setLocation(105,20);
        this.setContentPane(chartpanel);
        this.setVisible(true);
    }

    @Override
    public void update(String []nominees,int[]votes) {
      JFreeChart barChart = ChartFactory.createBarChart("Command + Memento",
"Nominees","Votes",
      createDataset(nominees,votes ),PlotOrientation.VERTICAL, true, true, false);
      ChartPanel chartPanel = new ChartPanel( barChart );
      Graph(chartPanel);
    }
}
```

```java
package Graphs;
/**
 * @author Victor Lavalle
 */
import javax.swing.JFrame;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.data.general.DefaultPieDataset;
import org.jfree.data.general.PieDataset;
import Patterns.observer.Observer;
import org.jfree.chart.plot.PiePlot3D;
import org.jfree.util.Rotation;

```

```
15
16  public class Circle extends JFrame implements Observer{
17
18      private PieDataset createDataset(String[] nominees, int[] votes){
19          DefaultPieDataset result = new DefaultPieDataset();
20           for(int i=0;i<nominees.length;i++){
21                   result.setValue(nominees[i], votes[i]);
22           }
23          return result;
24      }
25
26      private JFreeChart createChart(PieDataset dataset, String title){
27          JFreeChart chart = ChartFactory.createPieChart3D(title,
    dataset,true,true,false);
28          PiePlot3D plot = (PiePlot3D) chart.getPlot();
29          plot.setStartAngle(290);
30          plot.setDirection(Rotation.CLOCKWISE);
31          plot.setForegroundAlpha(0.5f);
32       return chart;
33      }
34
35      public void Graph(ChartPanel chartpanel){
36          chartpanel.setPreferredSize(new  java.awt.Dimension(500,500));
37          this.setSize(400, 300);
38          this.setLocation(860,20);
39          this.setContentPane(chartpanel);
40          this.setVisible(true);
41
42      }
43
44      @Override
45      public void update( String []nominees, int[]votes) {
46      PieDataset dataset = createDataset(nominees,votes);
47      JFreeChart chart = createChart(dataset,"Voting Command + Observer");
48      ChartPanel chartPanel = new  ChartPanel(chart);
49      Graph(chartPanel);
50      }
51  }
```

```
1   package Methods;
2
3   import java.io.PrintWriter;
4   import Patterns.observer.Observer;
5   import java.io.FileNotFoundException;
6   /**
7    * @author Victor Lavalle
8    */
9   public class CSVGenerator  implements Observer {
10
11     @Override
12      public void update(String []nominees,int[]votes) {
13
14           try{
15              PrintWriter esc = new PrintWriter("Results.csv");
16              String vote="";
17              for(int i=0;i<votes.length;i++){
18                  vote+=votes[i]+",";
19              }
20              esc.println(Arrays.toString(nominees));
21              esc.println(vote);
```

```
            esc.close();
        }
         catch(FileNotFoundException e){}
    }
}
```

```
package Patterns.command;
/**
 * @author Victor Lavalle
 */
import Patterns.observer.Votes;

public interface Command {
    public void addVote(Votes nominees);
}
```

```
package Patterns.command;

import Patterns.observer.Votes;

public class VoteJ implements Command {

    @Override
    public void addVote(Votes nominees) {
        nominees.addJuan();
    }
}
```

```
package Patterns.command;

import Patterns.observer.Votes;

public class VoteL implements Command{

    @Override
    public void addVote(Votes nominees) {
        nominees.addLuis();
    }
}
```

```
package Patterns.command;

import Patterns.observer.Votes;

public class VoteP implements Command {

    @Override
    public void addVote(Votes nominees) {
        nominees.addPedro();
    }
```

```java
package Patterns.memento;

import java.util.LinkedList;
/**
 * @author Victor Lavalle
 */
public class CareTaker {

    private final LinkedList<Memento>mementos= new LinkedList<>();

    public void addMemento(Memento memento){
        mementos.addLast(memento);
    }

    public Memento getMemento(){
       return mementos.getLast();
    }
}
```

```java
package Patterns.memento;

import Patterns.observer.Votes;

public class Memento {
    private final Votes votes;

    public Memento(Votes votes) { this.votes = votes;}
    public Votes getVotes() {return votes;}
}
```

```java
package Patterns.memento;

import Patterns.observer.Votes;

public class Originator {

    private Votes votes;

    public Votes getVotes() { return votes;}

    public void setVotes(Votes votos) {
        Votes _votos = new Votes();
        _votos.setJuan(votos.getJuan());
        _votos.setLuis(votos.getLuis());
        _votos.setPedro(votos.getPedro());
        this.votes = _votos;
    }

    public Memento Save(){
        return new Memento(votes);
    }

    public void Undo(Memento m){
        this.votes = m.getVotes();
    }

}
```

```java
package Patterns.observer;

import java.util.ArrayList;
/**
 * @author Victor Lavalle
 */
public abstract class Observed {

    ArrayList<Observer> observers= new ArrayList();

    public  void addObserver(Observer observer){
        observers.add(observer);
    }

    public void deleteObserver(int index){
        observers.remove(index);
    }

    public void notifyAll(String []nominees,int[]votos){
        for(int i=0;i<observers.size();i++){
            observers.get(i).update(nominees,votos);
         }
    }

}
```

```java
package Patterns.observer;

public interface Observer {
    public void update(String []Candidatos,int[]votos);
}
```

```java
package Patterns.observer;
/**
 * @author Victor Lavalle
 */
public class Votes {
    private int Juan, Pedro, Luis;

//Methods Get & Set
    public int getJuan() {return Juan; }
    public int getPedro() {return Pedro; }
    public int getLuis() {return Luis;}

    public void setJuan(int Juan) {this.Juan = Juan;}
    public void setPedro(int Pedro) {this.Pedro = Pedro;}
    public void setLuis(int Luis) { this.Luis = Luis; }

//Votes Counter
    public void addPedro(){Pedro++; }
    public void addJuan(){Juan++;}
    public void addLuis(){ Luis++; }
}
```

```java
package View;
```

```
2
3  import Patterns.memento.*;
4  import Patterns.command.*;
5  import Patterns.observer.*;
6  /**
7   * @author Victor Lavalle
8   */
9  public class UIvotes extends javax.swing.JFrame {
10
11     private Observed ob;
12     int[]votes= new int [3];
13     String [] nominees={"Juan","Luis","Pedro"};
14     VoteJ vj = new VoteJ();
15     VoteL vl = new VoteL();
16     VoteP vp = new VoteP();
17     Votes _votes= new Votes();
18     Originator or = new Originator();
19     CareTaker Ct = new CareTaker();
20
21     public void setOb(Observed ob) {this.ob = ob;}
22
23     public UIvotes() {
24         initComponents();
25         this.setLocationRelativeTo(null);
26         _votes.setJuan(0);
27         _votes.setLuis(0);
28         _votes.setPedro(0);
29         or.setVotes(_votes);
30         Ct.addMemento(or.Save());
31     }
32
33     @SuppressWarnings("unchecked")
34     // <editor-fold defaultstate="collapsed" desc="Generated Code">

35     private void initComponents() {
36
37         jPanel1 = new javax.swing.JPanel();
38         Vote2 = new javax.swing.JButton();
39         Vote1 = new javax.swing.JButton();
40         Vote3 = new javax.swing.JButton();
41         Undo = new javax.swing.JButton();
42         Name1 = new javax.swing.JLabel();
43         Name2 = new javax.swing.JLabel();
44         Name3 = new javax.swing.JLabel();
45         Title = new javax.swing.JLabel();
46         Credits = new javax.swing.JLabel();
47
48         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
49         getContentPane().setLayout(new
   org.netbeans.lib.awtextra.AbsoluteLayout());
50
51         jPanel1.setBackground(new java.awt.Color(32, 33, 36));
52         jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
53
54         Vote2.setBackground(new java.awt.Color(187, 134, 252));
55         Vote2.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N
56         Vote2.setForeground(new java.awt.Color(255, 255, 255));
57         Vote2.setText("VOTE");
58         Vote2.addActionListener(new java.awt.event.ActionListener() {
59             public void actionPerformed(java.awt.event.ActionEvent evt) {
60                 Vote2ActionPerformed(evt);
```

```
            }
        });
        jPanel1.add(Vote2, new org.netbeans.lib.awtextra.AbsoluteConstraints(230,
40, -1, -1));

        Vote1.setBackground(new java.awt.Color(187, 134, 252));
        Vote1.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N
        Vote1.setForeground(new java.awt.Color(255, 255, 255));
        Vote1.setText("VOTE");
        Vote1.setActionCommand("Vote1");
        Vote1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                Vote1ActionPerformed(evt);
            }
        });
        jPanel1.add(Vote1, new org.netbeans.lib.awtextra.AbsoluteConstraints(230,
90, -1, -1));

        Vote3.setBackground(new java.awt.Color(187, 134, 252));
        Vote3.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N
        Vote3.setForeground(new java.awt.Color(255, 255, 255));
        Vote3.setText("VOTE");
        Vote3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                Vote3ActionPerformed(evt);
            }
        });
        jPanel1.add(Vote3, new org.netbeans.lib.awtextra.AbsoluteConstraints(230,
140, -1, -1));

        Undo.setBackground(new java.awt.Color(57, 255, 20));
        Undo.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
        Undo.setForeground(new java.awt.Color(255, 255, 255));
        Undo.setText("UNDO");
        Undo.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                UndoActionPerformed(evt);
            }
        });
        jPanel1.add(Undo, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
180, -1, 30));

        Name1.setForeground(new java.awt.Color(255, 255, 255));
        Name1.setText("b) Luis");
        jPanel1.add(Name1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
100, -1, -1));

        Name2.setForeground(new java.awt.Color(255, 255, 255));
        Name2.setText("a) Juan");
        jPanel1.add(Name2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
50, -1, -1));

        Name3.setForeground(new java.awt.Color(255, 255, 255));
        Name3.setText("c) Pedro");
        jPanel1.add(Name3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
150, -1, -1));

        Title.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        Title.setForeground(new java.awt.Color(255, 255, 255));
        Title.setText("VOTING SYSTEM");
```

```
114         jPanel1.add(Title, new org.netbeans.lib.awtextra.AbsoluteConstraints(100,
    0, 110, 20));
115 
116         Credits.setFont(new java.awt.Font("Dialog", 2, 12)); // NOI18N
117         Credits.setForeground(new java.awt.Color(51, 51, 51));
118         Credits.setText("By Victor Lavalle");
119         jPanel1.add(Credits, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(110, 230, -1, -1));
120 
121         getContentPane().add(jPanel1, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 310, 260));
122 
123         pack();
124     }// </editor-fold>
125 
126     private void Vote2ActionPerformed(java.awt.event.ActionEvent evt) {
127       or.setVotes(_votes);
128       Ct.addMemento(or.Save());
129       vj.addVote(_votes);
130        votes[0]=_votes.getJuan();
131        votes[1]=_votes.getLuis();
132        votes[2]=_votes.getPedro();
133        ob.notifyAll(nominees, votes);
134     }
135 
136     private void Vote1ActionPerformed(java.awt.event.ActionEvent evt) {
137       or.setVotes(_votes);
138       Ct.addMemento(or.Save());
139        vl.addVote(_votes);
140        votes[0]=_votes.getJuan();
141        votes[1]=_votes.getLuis();
142        votes[2]=_votes.getPedro();
143        ob.notifyAll(nominees, votes);
144     }
145 
146     private void Vote3ActionPerformed(java.awt.event.ActionEvent evt) {
147        or.setVotes(_votes);
148        Ct.addMemento(or.Save());
149        vp.addVote(_votes);
150        votes[0]=_votes.getJuan();
151        votes[1]=_votes.getLuis();
152        votes[2]=_votes.getPedro();
153        ob.notifyAll(nominees, votes);
154     }
155 
156     private void UndoActionPerformed(java.awt.event.ActionEvent evt) {
157         or.Undo(Ct.getMemento());
158         _votes.setJuan(or.getVotes().getJuan());
159         _votes.setLuis(or.getVotes().getLuis());
160         _votes.setPedro(or.getVotes().getPedro());
161         votes[0]= or.getVotes().getJuan();
162         votes[1]= or.getVotes().getLuis();
163         votes[2]= or.getVotes().getPedro();
164        ob.notifyAll(nominees, votes);
165     }
166 
167     /**
```

```
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(UIvotes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(UIvotes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(UIvotes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(UIvotes.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new UIvotes().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel Credits;
    private javax.swing.JLabel Name1;
    private javax.swing.JLabel Name2;
    private javax.swing.JLabel Name3;
    private javax.swing.JLabel Title;
    private javax.swing.JButton Undo;
    private javax.swing.JButton Vote1;
    private javax.swing.JButton Vote2;
    private javax.swing.JButton Vote3;
    private javax.swing.JPanel jPanel1;
    // End of variables declaration
```

```
217 }
```

```java
package Launcher;

import Methods.CSVGenerator;
import Graphs.Bars;
import Graphs.Circle;
import Patterns.observer.Observed;
import View.UIvotes;
/**
 * @author Victor Lavalle
 */
public class Client {

    public static void main(String[] args) {

       Observed observer = new Observed() {};
       CSVGenerator csv = new CSVGenerator();
       Bars b =  new Bars();
       Circle c= new Circle();
       observer.addObserver(csv);
       observer.addObserver(b);
       observer.addObserver(c);
       UIvotes  ui= new UIvotes ();
       ui.setOb(observer);
       ui.setVisible(true);
    }
}
```

Diagrama de Clases

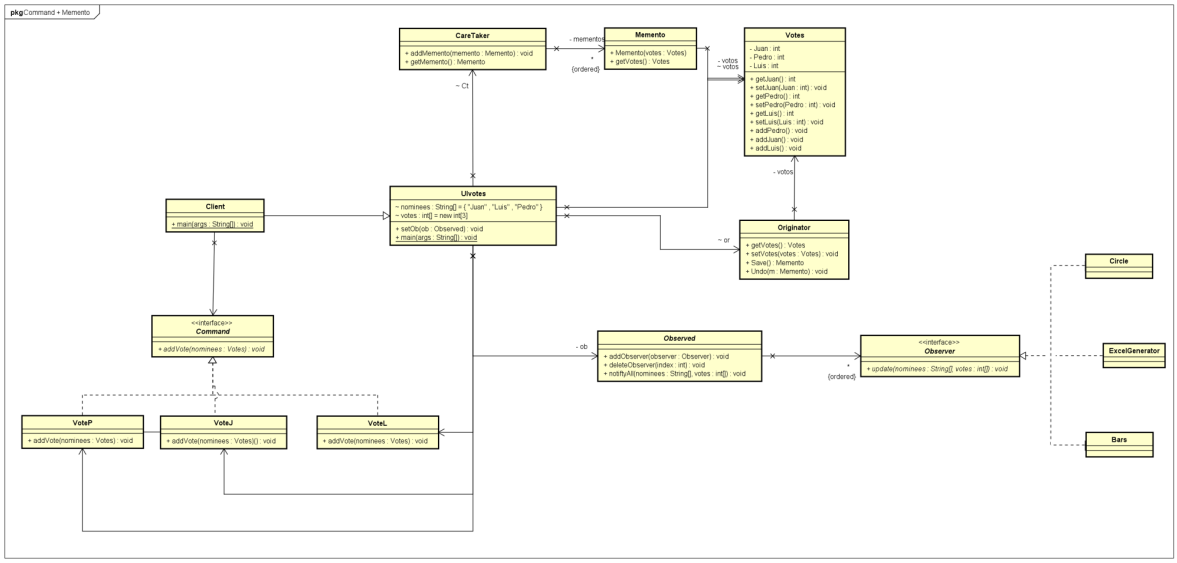
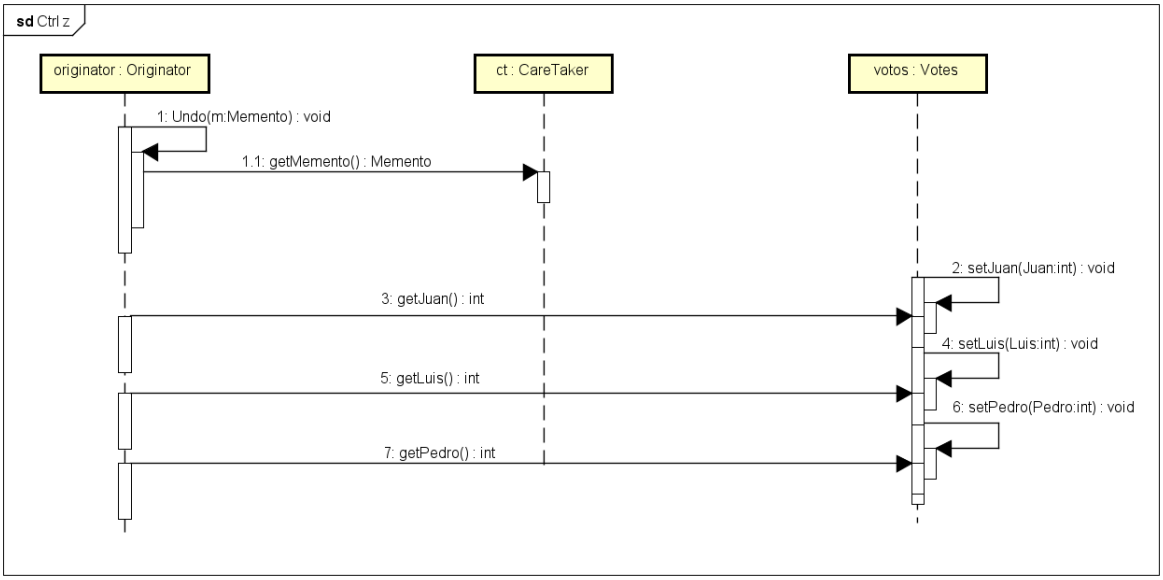
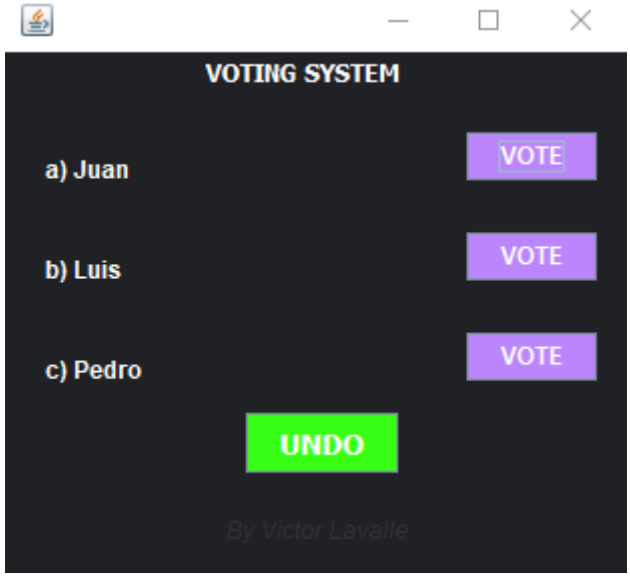


Diagrama de Secuencia

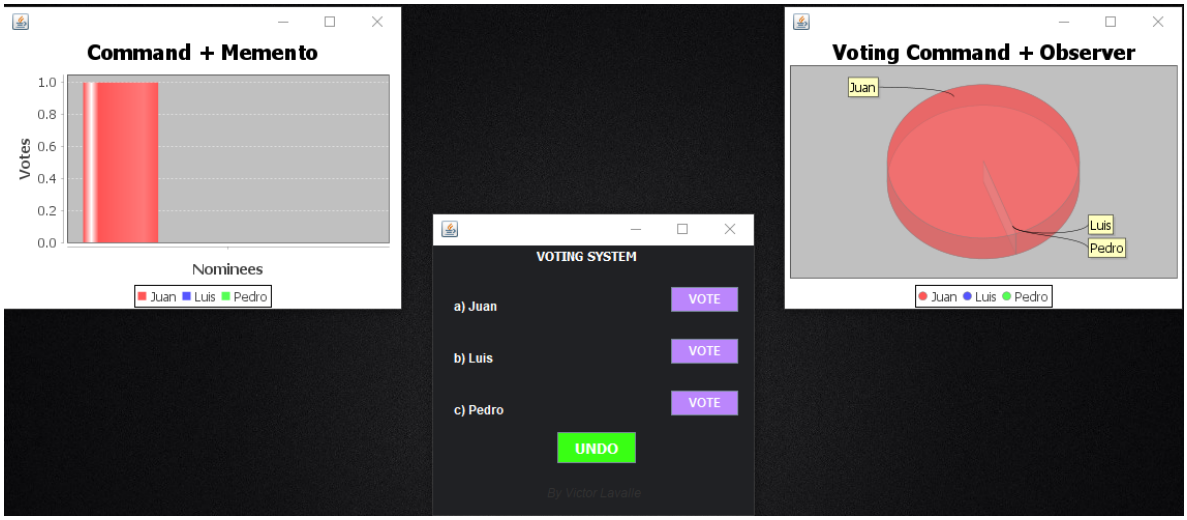


Ejecución

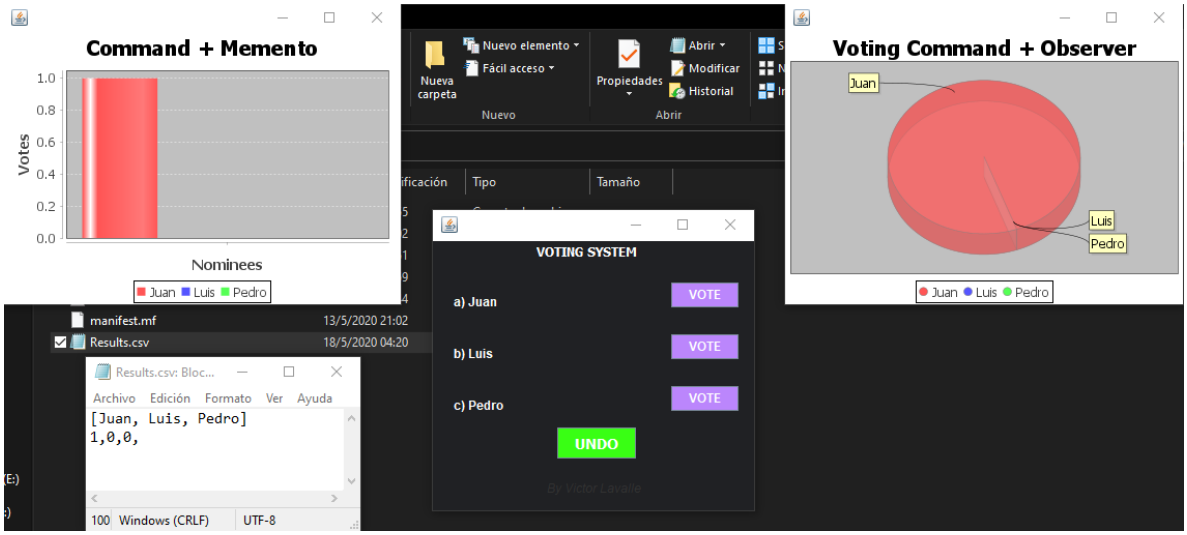
Al ejecutarse se muestra la ventan principal de votos



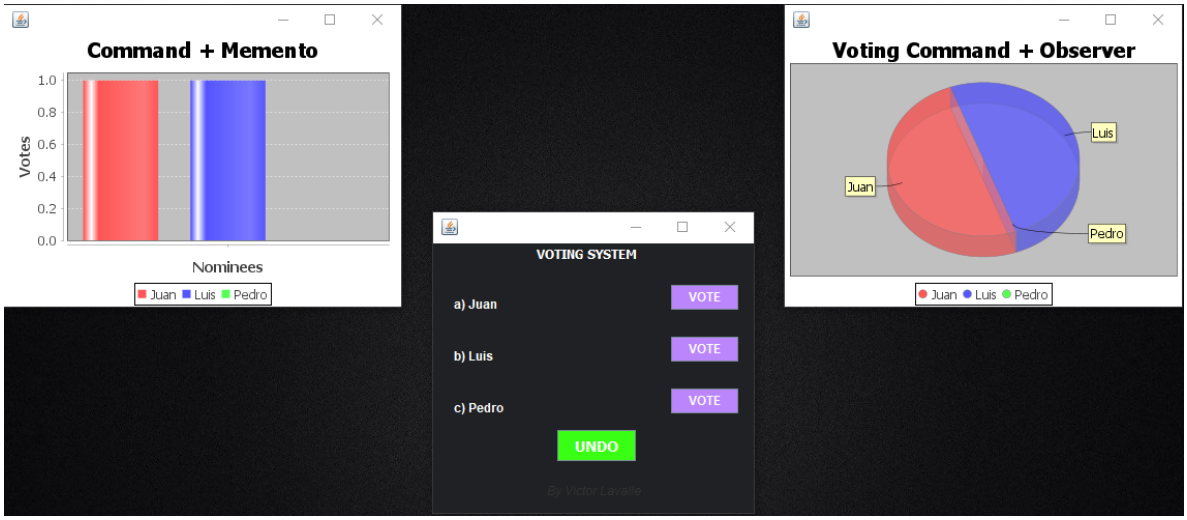
Se realiza un voto a favor de Juan



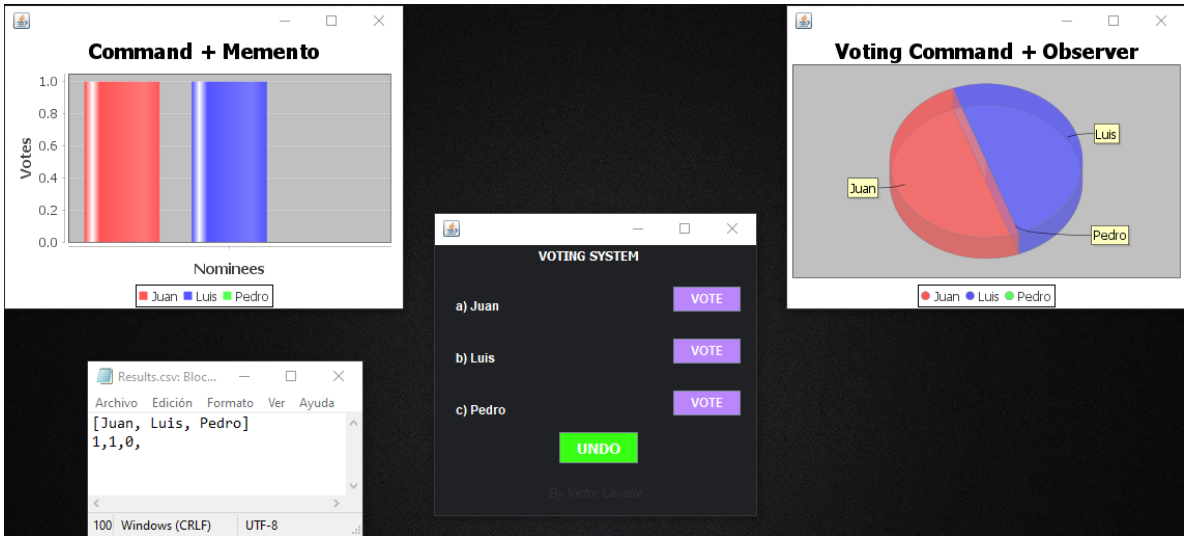
Se Crea el archivo CSV con el registro de votos



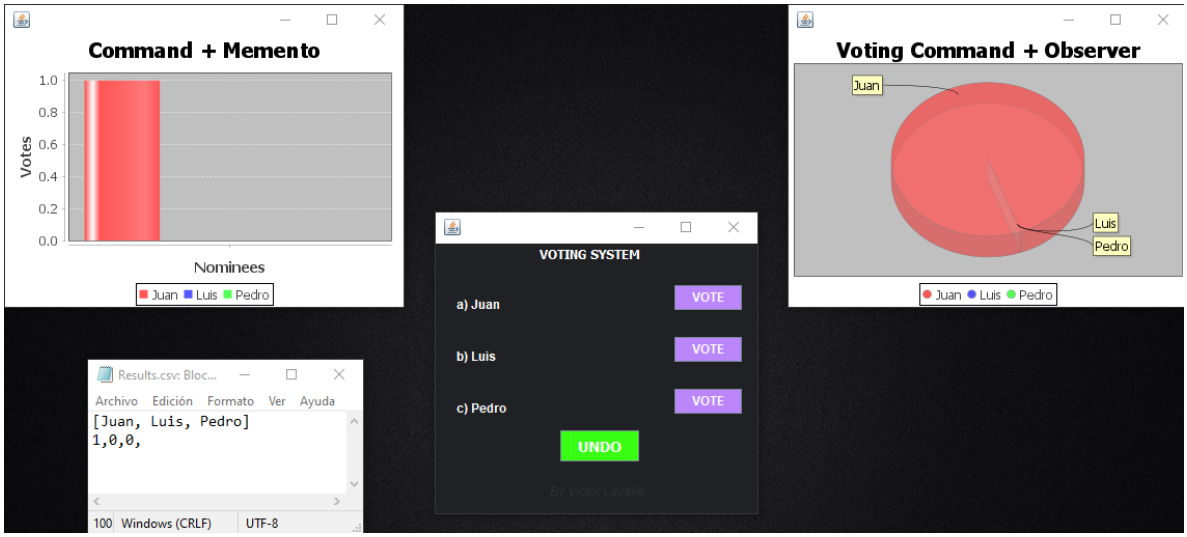
Se vota a favor de Luis



Se Actualiza el CSV



Hacemos CTRL + Z, haciendo click en el botón"UNDO"



Link de Video de Demostración:

https://drive.google.com/open?id=1Mnli-InepQjZsNL8j0zJxW0sL-dSFbf2