

Código Fuente

```
1 package Pattern;
2 /**
3  * @author Victor Lavalle
4  */
5 public interface Observer {
6
7     public void update(int[] votes, String[] nominees);
8
9 }
```

```
1 package Pattern;
2
3 import java.util.ArrayList;
4 /**
5  * @author Victor Lavalle
6  */
7 public class Observed {
8
9     protected ArrayList<Observer> observers;
10
11     //Constructor
12     public Observed(){
13         this.observers = new ArrayList<>();
14     }
15
16     //Functions to apply the pattern design
17     public void notifyAll(int[] votes, String[] nominees){
18
19         for(int i = 0; i < observers.size(); i++){
20             observers.get(i).update(votes, nominees);
21         }
22     }
23
24
25     // _____
26     public void addObserver(Observer o){
27         observers.add(o);
28     }
29
30     // _____
31     public void removeObserver(Observer o){
32         observers.remove(o);
33     }
34
35 }
```

```

1 package Pattern;
2
3 import java.io.PrintWriter;
4 import java.util.Arrays;
5 /**
6  * @author Victor Lavalle
7  */
8 public class ExcelGenerator implements Observer{
9
10     @Override
11     public void update(int[] votes, String[] nominees) {
12
13         try (PrintWriter esc = new PrintWriter("ObserverVotes.csv")) {
14             String vote="";
15             for(int i=0; i<votes.length;i++){
16                 vote+=votes[i]+",";
17             }
18             esc.println(Arrays.toString(nominees));
19             esc.println(vote);
20             esc.close();
21         }
22         catch(Exception e){
23         }
24     }
25 }

```

```

1 package Pattern;
2 /**
3  * @author Victor Lavalle
4  */
5 import View.UIVotes;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8
9 public class CaptureVotes extends Observed implements ActionListener {
10 private final UIVotes voting;
11 private final int[] votes;
12 private final String[] nominees;
13
14 public CaptureVotes(String[] nominees){
15     this.nominees = nominees;
16     int[] votes = {0, 0, 0};
17     this.votes = votes;
18     this.voting = new VotingFRM();
19     notifyAll(this.votes, this.nominees);
20     voting.getVote().addActionListener(this);
21     voting.getVote2().addActionListener(this);
22     voting.getVote3().addActionListener(this);
23     voting.setVisible(true);
24 }
25
26 @Override
27 public void actionPerformed(ActionEvent e) {
28
29     if(e.getSource().equals(voting.getVote())) {
30         votes[0]++;
31         notifyAll(votes, nominees);
32     }

```

```

33
34     if(e.getSource().equals(voting.getVote2())){
35         votes[1]++;
36         notifyAll(votes, nominees);
37     }
38
39     if(e.getSource().equals(voting.getVote3()) ){
40         votes[2] ++;
41         notifyAll(votes, nominees);
42     }
43
44 }
45 }

```

```

1 package Launcher;
2 /**
3  * @author Victor Lavalle
4  */
5 import Pattern.*;
6 import View.*;
7
8 public class Client {
9
10     public static void main(String[] args) {
11         String[] part = {"Discord", "Teams", "Zoom"};
12         CaptureVotes demo = new CaptureVotes(part);
13         BarsFRM b = new BarsFRM();
14         CircleFRM p = new CircleFRM();
15         ExcelGenerator e = new ExcelGenerator();
16         demo.addObserver(b);
17         demo.addObserver(p);
18         demo.addObserver(e);
19     }
20 }

```

```

1 package View;
2 /**
3  * @author Victor Lavalle
4  */
5 import Pattern.Observer;
6 import org.jfree.chart.ChartFactory;
7 import org.jfree.chart.ChartPanel;
8 import org.jfree.chart.JFreeChart;
9 import org.jfree.chart.plot.PlotOrientation;
10 import org.jfree.data.category.CategoryDataset;
11 import org.jfree.data.category.DefaultCategoryDataset;
12
13 public class BarsFRM extends javax.swing.JFrame implements Observer {
14
15     /**
16      * Creates new form BarsFRM
17      */
18     public BarsFRM() {
19         initComponents();
20         //Set the location of the windows that contains the graphs
21         this.setLocation(700,50);
22     }
23
24     private CategoryDataset createDataset(int[] votes, String[] nominees) {

```

```

25     final DefaultCategoryDataset dataset =
26     new DefaultCategoryDataset( );
27     dataset.setValue(votes[0], nominees[0], "");
28     dataset.setValue(votes[1], nominees[1], "");
29     dataset.setValue(votes[2], nominees[2], "");
30     return dataset;
31 }
32
33 @Override
34 public void update(int[] votes, String[] nominees){
35     JFreeChart barChart = ChartFactory.createBarChart("Voting with Observer",
36 "Nominees", "Votes",
37     createDataset(votes, nominees), PlotOrientation.VERTICAL, true, true, false);
38
39     ChartPanel chartPanel = new ChartPanel( barChart );
40     chartPanel.setPreferredSize(new java.awt.Dimension( 560 , 367 ) );
41     setContentPane( chartPanel );
42     this.setVisible(true);
43 }
44
45 /**
46  * This method is called from within the constructor to initialize the form.
47  * WARNING: Do NOT modify this code. The content of this method is always
48  * regenerated by the Form Editor.
49  */
50 @SuppressWarnings("unchecked")
51 // <editor-fold defaultstate="collapsed" desc="Generated Code">
52
53 private void initComponents() {
54
55     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
56
57     javax.swing.GroupLayout layout = new
58     javax.swing.GroupLayout(getContentPane());
59     getContentPane().setLayout(layout);
60     layout.setHorizontalGroup(
61         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
62         .addGroup(layout.createSequentialGroup()
63             .addGap(0, 400, Short.MAX_VALUE)
64             .addContainerGap())
65         .addGroup(layout.createSequentialGroup()
66             .addGap(0, 300, Short.MAX_VALUE)
67             .addContainerGap())
68     );
69     pack();
70 } // </editor-fold>
71
72 /**
73  * @param args the command line arguments
74  */
75 public static void main(String args[]) {
76     /* Set the Nimbus look and feel */
77     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
78     (optional) ">
79     /* If Nimbus (introduced in Java SE 6) is not available, stay with the
80     default look and feel.
81     * For details see
82     http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
83     */
84     try {

```

```

79         for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
80             if ("Nimbus".equals(info.getName())) {
81                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
82                 break;
83             }
84         }
85     } catch (ClassNotFoundException ex) {
86
87         java.util.logging.Logger.getLogger(BarsFRM.class.getName()).log(java.util.logging
88         g.Level.SEVERE, null, ex);
89     } catch (InstantiationException ex) {
90
91         java.util.logging.Logger.getLogger(BarsFRM.class.getName()).log(java.util.logging
92         g.Level.SEVERE, null, ex);
93     } catch (IllegalAccessException ex) {
94
95         java.util.logging.Logger.getLogger(BarsFRM.class.getName()).log(java.util.logging
96         g.Level.SEVERE, null, ex);
97     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
98
99         java.util.logging.Logger.getLogger(BarsFRM.class.getName()).log(java.util.logging
100        g.Level.SEVERE, null, ex);
101    }
102    }
103
104    // Variables declaration - do not modify
105    // End of variables declaration
106 }

```

```

1 package View;
2 /**
3  * @author Victor Lavalle
4  */
5 import Pattern.Observer;
6 import org.jfree.chart.ChartFactory;
7 import org.jfree.chart.ChartPanel;
8 import org.jfree.chart.JFreeChart;
9 import org.jfree.chart.plot.PiePlot3D;
10 import org.jfree.data.general.DefaultPieDataset;
11 import org.jfree.data.general.PieDataset;
12 import org.jfree.util.Rotation;
13
14 public class CircleFRM extends javax.swing.JFrame implements Observer {
15     /**
16      * Creates new form Circle
17      */
18     public CircleFRM() {
19         initComponents();
20         this.setLocation(200,50);
21     }
22

```

```

23 private PieDataset createDataset(int[] votes, String[] nominees) {
24     DefaultPieDataset result = new DefaultPieDataset();
25     result.setValue(nominees[0], votes[0]);
26     result.setValue(nominees[1], votes[1]);
27     result.setValue(nominees[2], votes[2]);
28     return result;
29 }
30
31 private JFreeChart createChart(PieDataset dataset, String title) {
32     JFreeChart chart = ChartFactory.createPieChart3D(title,
dataset,true,true,false);
33     PiePlot3D plot = (PiePlot3D) chart.getPlot();
34     plot.setStartAngle(290);
35     plot.setDirection(Rotation.CLOCKWISE);
36     plot.setForegroundAlpha(0.5f);
37     return chart;
38 }
39
40 @Override
41 public void update(int[] votos, String[] nominees){
42     PieDataset dataset = createDataset(votos, nominees);
43     JFreeChart chart = createChart(dataset, "Voting with Observer");
44     ChartPanel chartPanel = new ChartPanel(chart);
45     chartPanel.setPreferredSize(new java.awt.Dimension(500, 270));
46     setContentPane(chartPanel);
47     this.setVisible(true);
48 }
49
50
51 /**
52  * This method is called from within the constructor to initialize the form.
53  * WARNING: Do NOT modify this code. The content of this method is always
54  * regenerated by the Form Editor.
55  */
56 @SuppressWarnings("unchecked")
57 // <editor-fold defaultstate="collapsed" desc="Generated Code">
58
59 private void initComponents() {
60
61     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
62
63     javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
64     getContentPane().setLayout(layout);
65     layout.setHorizontalGroup(
66         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
67         .addGap(0, 400, Short.MAX_VALUE)
68     );
69     layout.setVerticalGroup(
70         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
71         .addGap(0, 300, Short.MAX_VALUE)
72     );
73
74     pack();
75 }// </editor-fold>
76
77 /**
78  * @param args the command line arguments
79  */
80 public static void main(String args[]) {
    /* Set the Nimbus look and feel */

```

```

81      </editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
82      /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
83          * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
84          */
85          try {
86              for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
87                  if ("Nimbus".equals(info.getName())) {
88                      javax.swing.UIManager.setLookAndFeel(info.getClassName());
89                      break;
90                  }
91              }
92          } catch (ClassNotFoundException ex) {
93
          java.util.logging.Logger.getLogger(CircleFRM.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
94          } catch (InstantiationException ex) {
95
          java.util.logging.Logger.getLogger(CircleFRM.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
96          } catch (IllegalAccessException ex) {
97
          java.util.logging.Logger.getLogger(CircleFRM.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
98          } catch (javax.swing.UnsupportedLookAndFeelException ex) {
99
          java.util.logging.Logger.getLogger(CircleFRM.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
100         }
101         </editor-fold>
102         </editor-fold>
103
104         /* Create and display the form */
105         java.awt.EventQueue.invokeLater(() -> {
106             new CircleFRM().setVisible(true);
107         });
108     }
109
110     // Variables declaration - do not modify
111     // End of variables declaration
112 }
113

```

```

1  package View;
2  /**
3   * @author Victor Lavalle
4   */
5  import javax.swing.JButton;
6  import javax.swing.JLabel;
7  import javax.swing.JPanel;
8
9  public final class UIVotes extends javax.swing.JFrame {
10
11      int Discord = 0;
12      int Teams = 0;
13      int Zoom = 0;
14

```

```

15 CircleFRM c = new CircleFRM();
16 BarsFRM b = new BarsFRM();
17
18 public int getDiscord() {return Discord;}
19 public int getTeams() { return Teams;}
20 public int getZoom() {return Zoom; }
21 public CircleFRM getC() {return c;}
22 public BarsFRM getB() {return b;}
23 public JButton getVote() { return Vote; }
24 public JButton getVote2() { return Vote2;}
25 public JButton getVote3() {return Vote3;}
26 public JPanel getjPanel1() { return jPanel1;}
27
28 public void setDiscord(int Discord) { this.Discord = Discord; }
29 public void setTeams(int Teams) { this.Teams = Teams; }
30 public void setZoom(int Zoom) { this.Zoom = Zoom; }
31 public void setC(CircleFRM c) { this.c = c; }
32 public void setB(BarsFRM b) {this.b = b; }
33 public void setVote(JButton Vote) { this.Vote = Vote; }
34 public void setVote2(JButton Vote2) { this.Vote2 = Vote2;}
35 public void setVote3(JButton Vote3) { this.Vote3 = Vote3; }
36 public void setjPanel1(JPanel jPanel1) { this.jPanel1 = jPanel1; }
37
38
39 public UIVotes() {
40     initComponents();
41     this.setResizable(false);
42     this.setLocation(480,460);
43     this.setTitle("Voting System with Observer ");
44 }
45
46 @SuppressWarnings("unchecked")
47 // <editor-fold defaultstate="collapsed" desc="Generated Code">
48
49 private void initComponents() {
50
51     jPanel1 = new javax.swing.JPanel();
52     Name1 = new javax.swing.JLabel();
53     Name2 = new javax.swing.JLabel();
54     Name3 = new javax.swing.JLabel();
55     Title = new javax.swing.JLabel();
56     Credits = new javax.swing.JLabel();
57     ZoomIMG = new javax.swing.JLabel();
58     DiscordIMG = new javax.swing.JLabel();
59     TeamsIMG = new javax.swing.JLabel();
60     Vote = new javax.swing.JButton();
61     Vote2 = new javax.swing.JButton();
62     Vote3 = new javax.swing.JButton();
63
64     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
65     setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
66     getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());
67
68     jPanel1.setBackground(new java.awt.Color(32, 33, 36));
69     jPanel1.setForeground(new java.awt.Color(255, 255, 255));
70     jPanel1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
71     jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
72
73     Name1.setForeground(new java.awt.Color(255, 255, 255));
74     Name1.setText("a Discord");

```



```

74         jPanel1.add(Name1, new org.netbeans.lib.awtextra.AbsoluteConstraints(40,
75         50, -1, -1));
76
77         Name2.setForeground(new java.awt.Color(255, 255, 255));
78         Name2.setText("b) Microsoft Teams");
79         jPanel1.add(Name2, new org.netbeans.lib.awtextra.AbsoluteConstraints(40,
80         100, -1, -1));
81
82         Name3.setForeground(new java.awt.Color(255, 255, 255));
83         Name3.setText("c) Zoom");
84         jPanel1.add(Name3, new org.netbeans.lib.awtextra.AbsoluteConstraints(40,
85         150, -1, -1));
86
87         Title.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
88         Title.setForeground(new java.awt.Color(255, 255, 255));
89         Title.setText("WICH ONE DO YOU PREFER?");
90         jPanel1.add(Title, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
91         10, 180, 20));
92
93         Credits.setFont(new java.awt.Font("Dialog", 2, 12)); // NOI18N
94         Credits.setForeground(new java.awt.Color(51, 51, 51));
95         Credits.setText("By Victor Lavalle");
96         jPanel1.add(Credits, new
97         org.netbeans.lib.awtextra.AbsoluteConstraints(120, 180, -1, -1));
98
99         ZoomIMG.setIcon(new
100         javax.swing.ImageIcon(getClass().getResource("/Resources/Zoom1.png"))); // NOI18N
101         jPanel1.add(ZoomIMG, new
102         org.netbeans.lib.awtextra.AbsoluteConstraints(10, 150, 20, 20));
103
104         DiscordIMG.setIcon(new
105         javax.swing.ImageIcon(getClass().getResource("/Resources/Discord.png"))); //
106         NOI18N
107         jPanel1.add(DiscordIMG, new
108         org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50, 20, 20));
109
110         TeamsIMG.setIcon(new
111         javax.swing.ImageIcon(getClass().getResource("/Resources/Teams.png"))); // NOI18N
112         jPanel1.add(TeamsIMG, new
113         org.netbeans.lib.awtextra.AbsoluteConstraints(10, 100, 20, 20));
114
115         Vote.setBackground(new java.awt.Color(187, 134, 252));
116         Vote.setFont(new java.awt.Font("Consolas", 1, 14)); // NOI18N
117         Vote.setText("VOTE");
118         Vote.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
119         Vote.addActionListener(new java.awt.event.ActionListener() {
120             public void actionPerformed(java.awt.event.ActionEvent evt) {
121                 VoteActionPerformed(evt);
122             }
123         });
124         jPanel1.add(Vote, new org.netbeans.lib.awtextra.AbsoluteConstraints(260,
125         40, -1, 30));
126
127         Vote2.setBackground(new java.awt.Color(187, 134, 252));
128         Vote2.setFont(new java.awt.Font("Consolas", 1, 14)); // NOI18N
129         Vote2.setText("VOTE");
130         Vote2.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
131         Vote2.addActionListener(new java.awt.event.ActionListener() {
132             public void actionPerformed(java.awt.event.ActionEvent evt) {
133                 Vote2ActionPerformed(evt);
134             }
135         });

```

```

122     });
123     jPanel1.add(Vote2, new org.netbeans.lib.awtextra.AbsoluteConstraints(260,
124     90, -1, 30));
125
126     Vote3.setBackground(new java.awt.Color(187, 134, 252));
127     Vote3.setFont(new java.awt.Font("Consolas", 1, 14)); // NOI18N
128     Vote3.setText("VOTE");
129     Vote3.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
130     Vote3.addActionListener(new java.awt.event.ActionListener() {
131         public void actionPerformed(java.awt.event.ActionEvent evt) {
132             Vote3ActionPerformed(evt);
133         }
134     });
135     jPanel1.add(Vote3, new org.netbeans.lib.awtextra.AbsoluteConstraints(260,
136     140, -1, 30));
137
138     getContentPane().add(jPanel1, new
139     org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 340, 200));
140
141     pack();
142 } // </editor-fold>
143
144 private void VoteActionPerformed(java.awt.event.ActionEvent evt) { /*TODO*/}
145
146 private void Vote2ActionPerformed(java.awt.event.ActionEvent evt) { /*TODO*/}
147
148 private void Vote3ActionPerformed(java.awt.event.ActionEvent evt) { /*TODO*/}
149
150 /**
151  * @param args the command line arguments
152  */
153 public static void main(String args[]) {
154     /* Create and display the form */
155     java.awt.EventQueue.invokeLater(new Runnable() {
156         public void run() {
157             new UIVotos().setVisible(true);
158         }
159     });
160 }
161
162 // Variables declaration - do not modify
163 private javax.swing.JLabel Credits;
164 private javax.swing.JLabel DiscordIMG;
165 private javax.swing.JLabel Name1;
166 private javax.swing.JLabel Name2;
167 private javax.swing.JLabel Name3;
168 private javax.swing.JLabel TeamsIMG;
169 private javax.swing.JLabel Title;
170 private javax.swing.JButton Vote;
171 private javax.swing.JButton Vote2;
172 private javax.swing.JButton Vote3;
173 private javax.swing.JLabel ZoomIMG;
174 private javax.swing.JPanel jPanel1;
175 // End of variables declaration
176 }

```

Diagrama de Clases

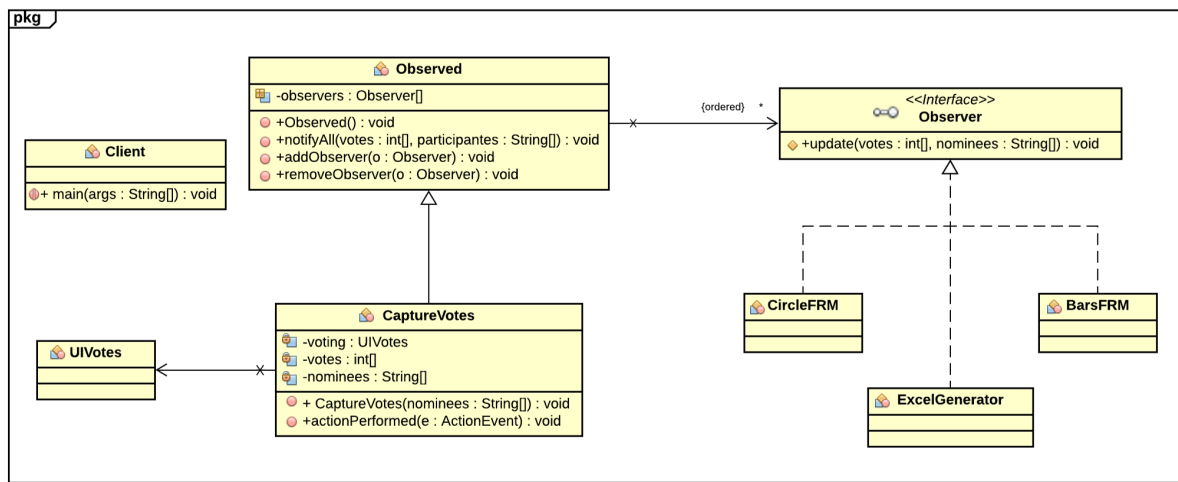
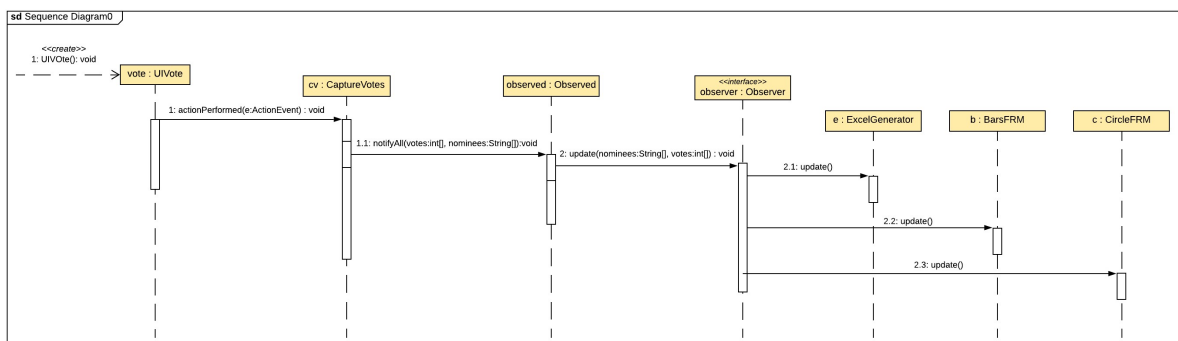
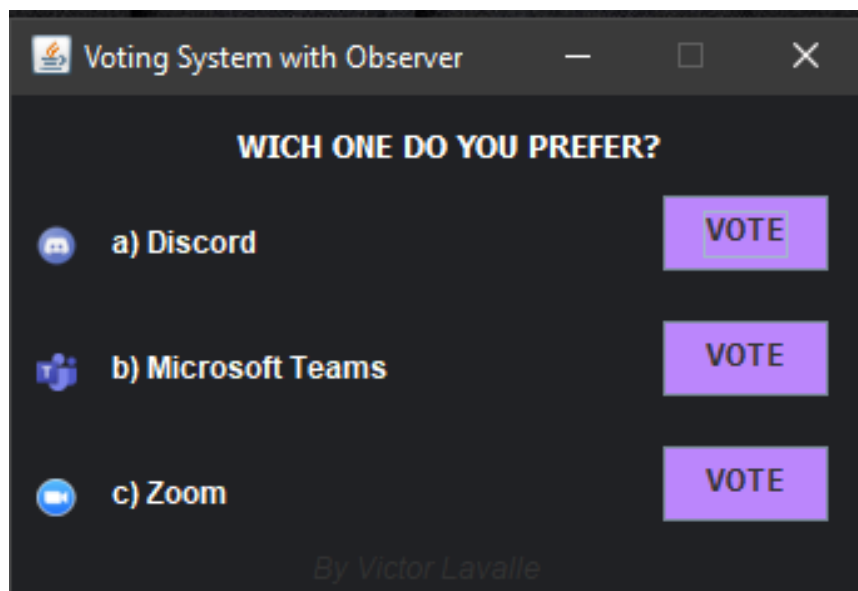


Diagrama de Secuencia

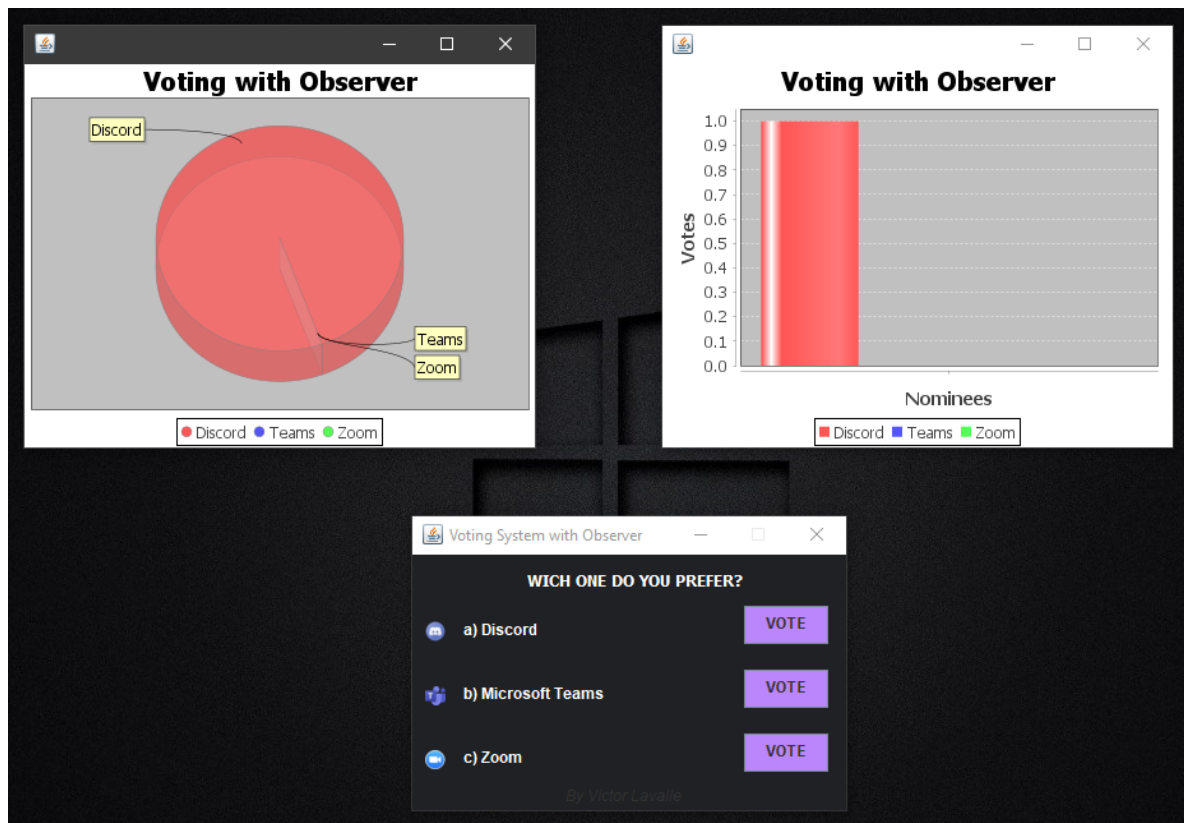


Ejecución del Programa

Al ejecutarse se muestra la ventana de *UIVotes*



Al hacer click en alguna opción de votar se crean las ventanas con graficas.



Link de Video de Demostración:

https://drive.google.com/file/d/1TM0pkTuEBdFWU4k_7Owr8IhfLcAZ3-MA/view?usp=sharing