



# RELATÓRIO TÉCNICO

Wesley de Oliveira Mendes - 828.507

## Tarefa 04 - Calculadora RMI

Prof. Rodrigo de Oliveira Plotze

# CalculatorService > src > service > ICalculator.java

```
1 package service;
2
3 import java.rmi.Remote;
4 import java.rmi.RemoteException;
5
6 /**
7 *
8 * @author Wesley
9 */
10 public interface ICalculator extends Remote {
11     // Propriedades do Servico
12     public static final int PORT = 2121;
13     public static final String SERVICE_NAME = "calculator";
14     public static final String SERVICE_HOST = "localhost";
15
16     // Assinaturas dos metodos que fazem parte do Servico
17     public abstract double getSum(double x, double y) throws RemoteException;
18     public abstract double getSubtraction(double x, double y) throws RemoteException;
19     public abstract double getMultiplication(double x, double y) throws RemoteException;
20     public abstract double getDivision(double x, double y) throws RemoteException;
21     public abstract double getSquareRoot(double n) throws RemoteException;
22     public abstract double getPotentiation(double b, double e) throws RemoteException;
23     public abstract double getLogarithm(double n) throws RemoteException;
24     public abstract double getFahrenheitToCelsius(double n) throws RemoteException;
25     public abstract double getCelsiusToFahrenheit(double n) throws RemoteException;
26 }
```

# CalculatorServer > src > server > Server.java

```
1 package server;
2
3 import java.rmi.RemoteException;
4 import java.rmi.registry.Registry;
5 import java.rmi.registry.LocateRegistry;
6 import java.rmi.server.UnicastRemoteObject;
7 import service.ICalculator;
8
9 /**
10 *
11 * @author Wesley
12 */
13
14 // UnicastRemoteObject: torna os metodos da classe acessiveis remotamente
15 public class Server extends UnicastRemoteObject implements ICalculator {
16
17     public Server() throws RemoteException{
18         super();
19     }
20
21     @Override
22     public double getSum(double x, double y) throws RemoteException {
23         return x + y;
24     }
25
26     @Override
27     public double getSubtraction(double x, double y) throws RemoteException {
28         return x - y;
29     }
30
31     @Override
32     public double getMultiplication(double x, double y) throws RemoteException {
33         return x * y;
34     }
35
36     @Override
37     public double getDivision(double x, double y) throws RemoteException {
38         return x / y;
39     }
40
41     @Override
42     public double getSquareRoot(double n) throws RemoteException {
43         return Math.sqrt(n);
44     }
45
46     @Override
47     public double getPotentiation(double b, double e) throws RemoteException {
48         return Math.pow(b, e);
49     }
50
51     @Override
52     public double getLogarithm(double n) throws RemoteException {
53         return Math.log10(n);
54     }
55
56     @Override
57     public double getFahrenheitToCelsius(double f) throws RemoteException {
58         return (f - 32.0) * 5.0 / 9.0;
59     }
60
61     @Override
62     public double getCelsiusToFahrenheit(double c) throws RemoteException {
63         return c * (9.0 / 5.0) + 32.0;
64     }
65
66     // EXECUTAR O SERVIDOR
67     public static void main(String[] args) {
68         try {
69             ICalculator srv = new Server();
70
71             // Registrar a porta de comunicacao
72             Registry rg = LocateRegistry.createRegistry(ICalculator.PORT);
73
74             // Disponibilizar o servico
75             rg.bind(ICalculator.SERVICE_NAME, srv);
76
77             System.err.println("Servidor em execucao...");
78         } catch (Exception e) {
79             System.err.println("ERRO: " + e.getMessage());
80         }
81     }
82 }
```

# CalculatorClient > src > client > ClientTest.java

```
1 package client;
2
3 import java.rmi.NotBoundException;
4 import java.rmi.RemoteException;
5 import java.rmi.registry.LocateRegistry;
6 import java.rmi.registry.Registry;
7 import service.ICalculator;
8
9 /**
10 *
11 * @author Wesley
12 */
13 public class ClientTest {
14
15     public static void main(String[] args) {
16         try {
17             // Registrar o Servico que sera acessado
18             Registry srv = LocateRegistry.getRegistry(
19                 ICalculator.SERVICE_HOST,
20                 ICalculator.PORT
21             );
22
23             // Descobrir quais metodos remotos estao disponiveis
24             ICalculator op = (ICalculator)srv.lookup(ICalculator.SERVICE_NAME);
25
26             System.out.println("Sum " + op.getSum(2, 5));
27             System.out.println("Subtraction " + op.getSubtraction(5, 2));
28             System.out.println("Multiplication " + op.getMultiplication(2, 5));
29             System.out.println("Division " + op.getDivision(10, 2));
30             System.out.println("Square Root " + op.getSquareRoot(9));
31             System.out.println("Potentiation " + op.getPotentiation(5, 2));
32             System.out.println("Logarithm " + op.getLogarithm(2));
33             System.out.println("Fahrenheit To Celsius " + op.getFahrenheitToCelsius(90));
34             System.out.println("Celsius To Fahrenheit " + op.getCelsiusToFahrenheit(22));
35
36         } catch (NotBoundException | RemoteException e) {
37             System.err.println("ERRO: " + e.getMessage());
38         }
39     }
40 }
```

# CalculatorClient > src > client > ClientForm.java

```
1 package client;
2
3 import java.awt.Color;
4 import java.rmi.NotBoundException;
5 import java.rmi.RemoteException;
6 import java.rmi.registry.LocateRegistry;
7 import java.rmi.registry.Registry;
8 import java.util.Arrays;
9 import service.ICalculator;
10
11 /**
12 *
13 * @author Wesley
14 */
15 public class ClientForm extends javax.swing.JFrame {
16
17     public ClientForm() {
18         initComponents();
19         this.setLocationRelativeTo(null);
20         this.setTitle("Calculator Client");
21         this.setResizable(false);
22         txtValue1.requestFocus();
23         txtResult.setEditable(false);
24         txtOperation.setEditable(false);
25     }
26
27     @SuppressWarnings("unchecked")
28 // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
29     private void initComponents() {
30
31         txtResult = new javax.swing.JTextField();
32         txtValue1 = new javax.swing.JTextField();
33         lblValue1 = new javax.swing.JLabel();
34         txtOperation = new javax.swing.JTextField();
35         lblOperation = new javax.swing.JLabel();
36         txtValue2 = new javax.swing.JTextField();
37         lblValue2 = new javax.swing.JLabel();
38         btnSum = new javax.swing.JButton();
39         lblSum = new javax.swing.JLabel();
40         btnSub = new javax.swing.JButton();
41         lblSub = new javax.swing.JLabel();
42         btnMult = new javax.swing.JButton();
43         lblMult = new javax.swing.JLabel();
44         btnDiv = new javax.swing.JButton();
45         lblDiv = new javax.swing.JLabel();
46         btnEqual = new javax.swing.JButton();
47         btnSqrt = new javax.swing.JButton();
48         lblSqrt = new javax.swing.JLabel();
49         btnPoten = new javax.swing.JButton();
50         lblPoten = new javax.swing.JLabel();
51         btnLog = new javax.swing.JButton();
52         lblLog = new javax.swing.JLabel();
53         btnFtoC = new javax.swing.JButton();
54         lblFtoC = new javax.swing.JLabel();
55         btnCtoF = new javax.swing.JButton();
56         lblCtoF = new javax.swing.JLabel();
57         lblValue3 = new javax.swing.JLabel();
58
59         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
60
61         txtResult.setBackground(new java.awt.Color(204, 204, 204));
62         txtResult.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
63         txtResult.setHorizontalAlignment(javax.swing.JTextField.CENTER);
64         txtResult.setActionCommand("<Not Set>");
65
66         txtValue1.setFont(new java.awt.Font("Tahoma", 0, 20)); // NOI18N
67         txtValue1.setHorizontalAlignment(javax.swing.JTextField.CENTER);
68         txtValue1.setActionCommand("<Not Set>");
69
70         lblValue1.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
71         lblValue1.setText("V1");
72
73         txtOperation.setFont(new java.awt.Font("Tahoma", 0, 20)); // NOI18N
74         txtOperation.setHorizontalAlignment(javax.swing.JTextField.CENTER);
75         txtOperation.setActionCommand("<Not Set>");
76
77         lblOperation.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
78         lblOperation.setText("Operação");
79
80         txtValue2.setFont(new java.awt.Font("Tahoma", 0, 20)); // NOI18N
81         txtValue2.setHorizontalAlignment(javax.swing.JTextField.CENTER);
82         txtValue2.setActionCommand("<Not Set>");
```

```

84     lblValue2.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
85     lblValue2.setText("V2");
86
87     btnSum.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
88     btnSum.setText("+");
89     btnSum.setToolTipText("Precisa do V1 e V2.");
90     btnSum.addActionListener(new java.awt.event.ActionListener() {
91         public void actionPerformed(java.awt.event.ActionEvent evt) {
92             btnSumActionPerformed(evt);
93         }
94     });
95
96     lblSum.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
97     lblSum.setText("V1 + V2");
98
99     btnSub.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
100    btnSub.setText("-");
101    btnSub.setToolTipText("Precisa do V1 e V2.");
102    btnSub.addActionListener(new java.awt.event.ActionListener() {
103        public void actionPerformed(java.awt.event.ActionEvent evt) {
104            btnSubActionPerformed(evt);
105        }
106    });
107
108    lblSub.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
109    lblSub.setText("V1 - V2");
110
111    btnMult.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
112    btnMult.setText("x");
113    btnMult.setToolTipText("Precisa do V1 e V2.");
114    btnMult.addActionListener(new java.awt.event.ActionListener() {
115        public void actionPerformed(java.awt.event.ActionEvent evt) {
116            btnMultActionPerformed(evt);
117        }
118    });
119
120    lblMult.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
121    lblMult.setText("V1 * V2");
122
123    btnDiv.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
124    btnDiv.setText("/");
125    btnDiv.setToolTipText("Precisa do V1 e V2.");
126    btnDiv.addActionListener(new java.awt.event.ActionListener() {
127        public void actionPerformed(java.awt.event.ActionEvent evt) {
128            btnDivActionPerformed(evt);
129        }
130    });
131
132    lblDiv.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
133    lblDiv.setText("V1 / V2");
134
135    btnIqual.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
136    btnIqual.setText "=";
137    btnIqual.addActionListener(new java.awt.event.ActionListener() {
138        public void actionPerformed(java.awt.event.ActionEvent evt) {
139            btnIqualActionPerformed(evt);
140        }
141    });
142
143    btnSqrt.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
144    btnSqrt.setText("√");
145    btnSqrt.setToolTipText("Precisa apenas do V1.");
146    btnSqrt.addActionListener(new java.awt.event.ActionListener() {
147        public void actionPerformed(java.awt.event.ActionEvent evt) {
148            btnSqrtActionPerformed(evt);
149        }
150    });
151
152    lblSqrt.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
153    lblSqrt.setText("sqrt(V1)");
154
155    btnPoten.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
156    btnPoten.setText("x2");
157    btnPoten.setToolTipText("Precisa do V1 e V2.");
158    btnPoten.addActionListener(new java.awt.event.ActionListener() {
159        public void actionPerformed(java.awt.event.ActionEvent evt) {
160            btnPotenActionPerformed(evt);
161        }
162    });
163
164    lblPoten.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
165    lblPoten.setText("V12V2");
166

```



```

251         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
252             .addComponent(lblLog)
253             .addComponent(btnLog, javax.swing.GroupLayout.PREFERRED_SIZE, 65, javax.swing.GroupLayout.PREFERRED_SIZE)))
254     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
255         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
256             .addComponent(lblValue1)
257             .addComponent(txtValue1, javax.swing.GroupLayout.PREFERRED_SIZE, 201, javax.swing.GroupLayout.PREFERRED_SIZE))
258         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
259     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
260         .addComponent(txtOperation, javax.swing.GroupLayout.PREFERRED_SIZE, 101, javax.swing.GroupLayout.PREFERRED_SIZE)
261         .addComponent(lblOperation))
262     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
263     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
264         .addGroup(layout.createSequentialGroup()
265             .addComponent(lblValue2)
266             .addGap(0, 0, Short.MAX_VALUE)
267             .addComponent(txtValue2))))
268     .addContainerGap())
269 );
270 layout.setVerticalGroup(
271     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
272         .addGroup(layout.createSequentialGroup()
273             .addComponent(lblValue3)
274             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
275             .addComponent(txtResult, javax.swing.GroupLayout.PREFERRED_SIZE, 70, javax.swing.GroupLayout.PREFERRED_SIZE)
276             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
277         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
278             .addComponent(lblValue1)
279             .addComponent(lblOperation)
280             .addComponent(lblValue2))
281             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
282         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
283             .addComponent(txtValue1, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
284             .addComponent(txtOperation, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
285             .addComponent(txtValue2, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE))
286             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
287         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
288             .addComponent(lblSum)
289             .addComponent(lblSub)
290             .addComponent(lblMult)
291             .addComponent(lblDiv)
292             .addComponent(lblSqrt)
293             .addComponent(lblPoten)
294             .addComponent(lblLog))
295         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
296             .addGroup(layout.createSequentialGroup()
297                 .addComponent(btnSum, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
298                 .addComponent(btnSub, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
299                 .addComponent(btnMult, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
300                 .addComponent(btnDiv, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
301                 .addComponent(btnSqrt, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)))
302         .addGroup(layout.createSequentialGroup()
303             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
304                 .addComponent(btnPoten, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
305                 .addComponent(btnLog, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)))
306             .addGap(6, 6)
307         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
308             .addComponent(btnCtoF, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
309             .addComponent(btnLog, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)))
310         .addGap(18, 18, 18)
311         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
312             .addComponent(lblFtoC)
313             .addComponent(lblCtoF))
314         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
315         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
316             .addComponent(btnCtoF, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
317             .addComponent(btnEqual, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)
318             .addComponent(btnFtoC, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE))
319         .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
320 );
321
322     pack();
323 } // </editor-fold>//GEN-END:initComponents
324

```

```
325     private void btnSumActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnSumActionPerformed
326         txtValue2.setBackground(Color.WHITE);
327         txtValue2.setEditable(true);
328         txtOperation.setText("+");
329     }//GEN-LAST:event_btnSumActionPerformed
330 
331     private void btnSubActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnSubActionPerformed
332         txtValue2.setBackground(Color.WHITE);
333         txtValue2.setEditable(true);
334         txtOperation.setText("-");
335     }//GEN-LAST:event_btnSubActionPerformed
336 
337     private void btnMultActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnMultActionPerformed
338         txtValue2.setBackground(Color.WHITE);
339         txtValue2.setEditable(true);
340         txtOperation.setText("x");
341     }//GEN-LAST:event_btnMultActionPerformed
342 
343     private void btnDivActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnDivActionPerformed
344         txtValue2.setBackground(Color.WHITE);
345         txtValue2.setEditable(true);
346         txtOperation.setText("/");
347     }//GEN-LAST:event_btnDivActionPerformed
348 
349     private void btnSqrtActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnSqrtActionPerformed
350         txtValue2.setBackground(Color.LIGHT_GRAY);
351         txtValue2.setEditable(false);
352         txtOperation.setText("√");
353     }//GEN-LAST:event_btnSqrtActionPerformed
354 
355     private void btnPotenActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnPotenActionPerformed
356         txtValue2.setBackground(Color.WHITE);
357         txtValue2.setEditable(true);
358         txtOperation.setText("^");
359     }//GEN-LAST:event_btnPotenActionPerformed
360 
361     private void btnLogActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnLogActionPerformed
362         txtValue2.setBackground(Color.LIGHT_GRAY);
363         txtValue2.setEditable(false);
364         txtOperation.setText("log");
365     }//GEN-LAST:event_btnLogActionPerformed
366 
367     private void btnFtoCActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnFtoCActionPerformed
368         txtValue2.setBackground(Color.LIGHT_GRAY);
369         txtValue2.setEditable(false);
370         txtOperation.setText("°F > °C");
371     }//GEN-LAST:event_btnFtoCActionPerformed
372 
373     private void btnCtoFActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnCtoFActionPerformed
374         txtValue2.setBackground(Color.LIGHT_GRAY);
375         txtValue2.setEditable(false);
376         txtOperation.setText("°C > °F");
377     }//GEN-LAST:event_btnCtoFActionPerformed
378 }
```

```
379 private void btnIqualActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnIqualActionPerformed
380     double v1 = -1, v2 = -1, result;
381     String operation = "", v1_s, v2_s, value_s = "";
382     String[] opArray = new String[] {"√", "log", "°F > °C", "°C > °F"};
383
384     try {
385         // Registrar o Servico que sera acessado
386         Registry srv = LocateRegistry.getRegistry(
387             ICalculator.SERVICE_HOST,
388             ICalculator.PORT
389         );
390
391         // Descobrir quais metodos remotos estao disponiveis
392         ICalculator op = (ICalculator)srv.lookup(ICalculator.SERVICE_NAME);
393
394         operation = txtOperation.getText();
395
396         try {
397             v1_s = txtValue1.getText();
398             if (!v1_s.isEmpty()) {
399                 v1 = Double.parseDouble(v1_s.replace(",","."));
400             } else {
401                 v1 = -1;
402                 operation = "v1";
403             }
404
405             v2_s = txtValue2.getText();
406             if (!v2_s.isEmpty()) {
407                 v2 = Double.parseDouble(v2_s.replace(",","."));
408             } else if (Arrays.asList(opArray).contains(operation)) {
409                 v2 = -1;
410             } else if (!"v1".equals(operation)) {
411                 v2 = -1;
412                 operation = "v2";
413             }
414         } catch (NumberFormatException e) {
415             v1 = -1;
416             v2 = -1;
417             operation = "";
418         }
419     }
```

```

420     switch (operation) {
421         case "+" -> {
422             result = op.getSum(v1, v2);
423             value_s = String.format("%.2f", result);
424             System.out.println(result);
425         }
426         case "-" -> {
427             result = op.getSubtraction(v1, v2);
428             value_s = String.format("%.2f", result);
429             System.out.println(result);
430         }
431         case "x" -> {
432             result = op.getMultiplication(v1, v2);
433             value_s = String.format("%.2f", result);
434             System.out.println(result);
435         }
436         case "/" -> {
437             result = op.getDivision(v1, v2);
438             value_s = String.format("%.2f", result);
439             System.out.println(result);
440         }
441         case "√" -> {
442             result = op.getSquareRoot(v1);
443             value_s = String.format("%.2f", result);
444             System.out.println(result);
445         }
446         case "^" -> {
447             result = op.getPotentiation(v1, v2);
448             value_s = String.format("%.2f", result);
449             System.out.println(result);
450         }
451         case "log" -> {
452             result = op.getLogarithm(v1);
453             value_s = String.format("%.2f", result);
454             System.out.println(result);
455         }
456         case "°F > °C" -> {
457             result = op.getFahrenheitToCelsius(v1);
458             value_s = String.format("%.2f", result);
459             System.out.println(result);
460         }
461         case "°C > °F" -> {
462             result = op.getCelsiusToFahrenheit(v1);
463             value_s = String.format("%.2f", result);
464             System.out.println(result);
465         }
466         case "v1" -> {
467             value_s = String.valueOf("V1 não pode ser vazio...");
468         }
469         case "v2" -> {
470             value_s = String.valueOf("V2 não pode ser vazio...");
471         }
472         default -> {
473             value_s = "OPERATION ERROR!!!";
474         }
475     }
476
477     txtResult.setText(value_s);
478 } catch (NotBoundException | RemoteException e) {
479     System.err.println("ERRO: " + e.getMessage());
480 }
481 //GEN-LAST:event_btnIqualActionPerformed
482

```

```

483 /**
484 * @param args the command line arguments
485 */
486 public static void main(String args[]) {
487     /* Set the Nimbus look and feel */
488     //editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
489     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
490      * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
491     */
492     try {
493         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
494             if ("Nimbus".equals(info.getName())) {
495                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
496                 break;
497             }
498         }
499     } catch (ClassNotFoundException ex) {
500         java.util.logging.Logger.getLogger(ClientForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
501     } catch (InstantiationException ex) {
502         java.util.logging.Logger.getLogger(ClientForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
503     } catch (IllegalAccessException ex) {
504         java.util.logging.Logger.getLogger(ClientForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
505     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
506         java.util.logging.Logger.getLogger(ClientForm.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
507     }
508     //
509
510     /* Create and display the form */
511     java.awt.EventQueue.invokeLater(new Runnable() {
512         public void run() {
513             new ClientForm().setVisible(true);
514         }
515     });
516 }
517
518 // Variables declaration - do not modify//GEN-BEGIN:variables
519 private javax.swing.JButton btnCtoF;
520 private javax.swing.JButton btnDiv;
521 private javax.swing.JButton btnFtoC;
522 private javax.swing.JButton btnIqual;
523 private javax.swing.JButton btnLog;
524 private javax.swing.JButton btnMult;
525 private javax.swing.JButton btnPoten;
526 private javax.swing.JButton btnSqrt;
527 private javax.swing.JButton btnSub;
528 private javax.swing.JButton btnSum;
529 private javax.swing.JLabel lblCtoF;
530 private javax.swing.JLabel lblDiv;
531 private javax.swing.JLabel lblFtoC;
532 private javax.swing.JLabel lblLog;
533 private javax.swing.JLabel lblMult;
534 private javax.swing.JLabel lblOperation;
535 private javax.swing.JLabel lblPoten;
536 private javax.swing.JLabel lblSqrt;
537 private javax.swing.JLabel lblSub;
538 private javax.swing.JLabel lblSum;
539 private javax.swing.JLabel lblValue1;
540 private javax.swing.JLabel lblValue2;
541 private javax.swing.JLabel lblValue3;
542 private javax.swing.JTextField txtOperation;
543 private javax.swing.JTextField txtResult;
544 private javax.swing.JTextField txtValue1;
545 private javax.swing.JTextField txtValue2;
546 // End of variables declaration//GEN-END:variables
547 }

```

## PROJECTS:

<https://github.com/WesGtoX/Distributed-and-Concurrent-Systems/tree/main/CalculatorService>  
<https://github.com/WesGtoX/Distributed-and-Concurrent-Systems/tree/main/CalculatorServer>  
<https://github.com/WesGtoX/Distributed-and-Concurrent-Systems/tree/main/CalculatorClient>

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

V1                      Operação              V2

--	--	--

V1 + V2    V1 - V2    V1 \* V2    V1 / V2    sqrt(V1)    V1^V2    log(V1)

+    -    x    /    √    x<sup>2</sup>    Log

°F = V1                      °C = V1

°F -> °C    =    °C -> °F

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

7,00

V1                      Operação              V2

5	+	2
---	---	---

V1 + V2    V1 - V2    V1 \* V2    V1 / V2    sqrt(V1)    V1^V2    log(V1)

+    -    x    /    √    x<sup>2</sup>    Log

°F = V1                      °C = V1

°F -> °C    =    °C -> °F

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

3,00

V1                      Operação              V2

5	-	2
---	---	---

V1 + V2    V1 - V2    V1 \* V2    V1 / V2    sqrt(V1)    V1^V2    log(V1)

+    -    x    /    √    x<sup>2</sup>    Log

°F = V1                      °C = V1

°F -> °C    =    °C -> °F

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

10,00

V1                      Operação              V2

5	x	2
---	---	---

V1 + V2    V1 - V2    V1 \* V2    V1 / V2    sqrt(V1)    V1^V2    log(V1)

+    -    x    /    √    x<sup>2</sup>    Log

°F = V1                      °C = V1

°F -> °C    =    °C -> °F

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

2,50

V1                      Operação              V2

5	/	2
---	---	---

V1 + V2    V1 - V2    V1 \* V2    V1 / V2    sqrt(V1)    V1^V2    log(V1)

+    -    x    /    √    x<sup>2</sup>    Log

°F = V1                      °C = V1

°F -> °C    =    °C -> °F

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

3,00

V1                      Operação              V2

9	√	
---	---	--

V1 + V2    V1 - V2    V1 \* V2    V1 / V2    sqrt(V1)    V1^V2    log(V1)

+    -    x    /    √    x<sup>2</sup>    Log

°F = V1                      °C = V1

°F -> °C    =    °C -> °F

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

81,00

V1	Operação	V2
9	$\wedge$	2

$V1 + V2$     $V1 - V2$     $V1 * V2$     $V1 / V2$     $\sqrt{V1}$     $V1^V2$     $\log(V1)$

+   -   x   /    $\sqrt{\phantom{x}}$     $x^2$    Log

${}^\circ F = V1$     ${}^\circ C = V1$

${}^\circ F \rightarrow {}^\circ C$    =    ${}^\circ C \rightarrow {}^\circ F$

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

1,95

V1	Operação	V2
90	log	

$V1 + V2$     $V1 - V2$     $V1 * V2$     $V1 / V2$     $\sqrt{V1}$     $V1^V2$     $\log(V1)$

+   -   x   /    $\sqrt{\phantom{x}}$     $x^2$    Log

${}^\circ F = V1$     ${}^\circ C = V1$

${}^\circ F \rightarrow {}^\circ C$    =    ${}^\circ C \rightarrow {}^\circ F$

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

32,22

V1	Operação	V2
90	${}^\circ F > {}^\circ C$	

$V1 + V2$     $V1 - V2$     $V1 * V2$     $V1 / V2$     $\sqrt{V1}$     $V1^V2$     $\log(V1)$

+   -   x   /    $\sqrt{\phantom{x}}$     $x^2$    Log

${}^\circ F = V1$     ${}^\circ C = V1$

${}^\circ F \rightarrow {}^\circ C$    =    ${}^\circ C \rightarrow {}^\circ F$

**Output**

CalculatorServer (run) × CalculatorClient (run) ×

run:  
Servidor em execução...

**Calculator Client**

**RESULT**

90,00

V1	Operação	V2
32.22	${}^\circ C > {}^\circ F$	

$V1 + V2$     $V1 - V2$     $V1 * V2$     $V1 / V2$     $\sqrt{V1}$     $V1^V2$     $\log(V1)$

+   -   x   /    $\sqrt{\phantom{x}}$     $x^2$    Log

${}^\circ F = V1$     ${}^\circ C = V1$

${}^\circ F \rightarrow {}^\circ C$    =    ${}^\circ C \rightarrow {}^\circ F$