



# Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia

Fundamentos de Sistemas Operativos - 2324SI

---

## 1<sup>o</sup> Trabalho Prático - Aula prática 1

Trabalho 1 - GUI Base

Canal de Comunicação

Ficheiro do Canal  ...

Nº msg

Controle do Robot

Raio

Ângulo

Distância

☐ Ativar / Desativar Comportamento

Log

Docente Carlos Carvalho

Realizado por (Grupo 7):

Diogo Santos 48626

Pedro Silva 48965

João Fonseca 49707

26 de setembro de 2024

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>I</b>
<b>2</b>	<b>Desenvolvimento</b>	<b>II</b>
2.1	GUI_Base . . . . .	II
2.2	GUI_Subdito e GUI_Rei . . . . .	III
<b>3</b>	<b>Conclusões</b>	<b>III</b>
<b>4</b>	<b>Bibliografia</b>	<b>III</b>
<b>5</b>	<b>Código Java GUI_Base, GUI_Subdito, GUI_Rei</b>	<b>IV</b>

# 1 Introdução

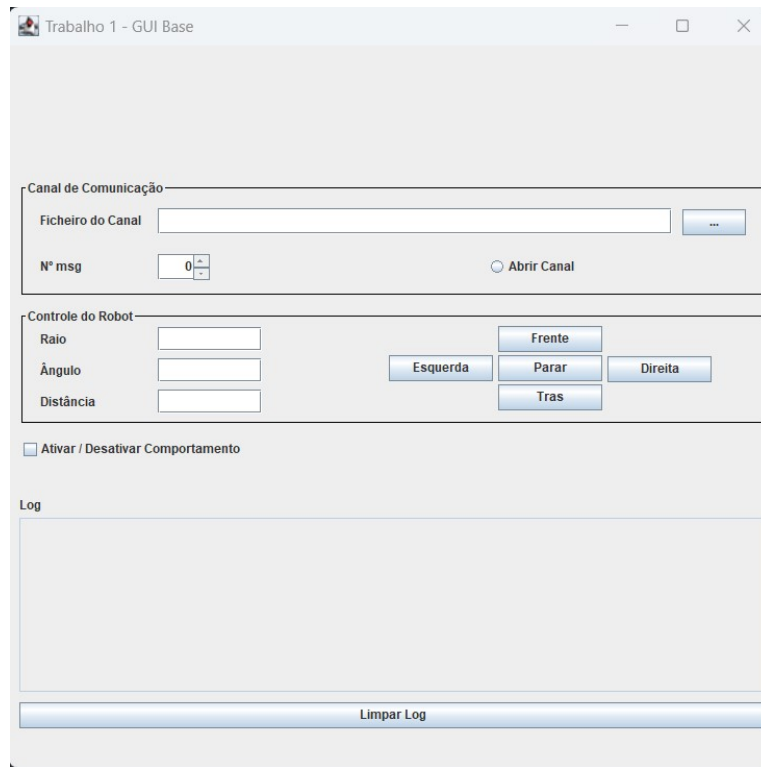
Neste primeiro trabalho prático, pretende-se o desenvolvimento de uma aplicação multiprocesso composta por dois processos Java para suportar o jogo “O Rei manda”. Um dos processos implementa o comportamento Rei, o segundo processo implementa o comportamento Súdito que conhece a API do Robot. O projeto está dividido em sete aulas práticas onde no final de cada temos de fazer a validação do trabalho realizado com o docente e antes da próxima elaborar um relatório sobre a mesma.

Esta primeira aula tivemos como objetivo a instalação do ”WindowBuilder” sobre o Eclipse e o desenho duma GUI com os componentes comuns aos processos Súdito e Rei. Esta primeira GUI vai servir de base para as outras duas sendo assim, apenas necessário estender a classe com a GUI base e acrescentar os conteúdos específicos de cada uma. Estas interfaces gráficas vão nos permitir controlar o robot e fazer a comunicação de processos entre o Súdito e o Rei.

## 2 Desenvolvimento

### 2.1 GUI\_Base

Como foi dito anteriormente, esta primeira interface gráfica vai ter o objetivo de conter todos as componentes e funcionalidades comuns aos processos Súdito e Rei para que posteriormente possamos apenas acrescentar o que cada uma tem de único. A imagem abaixo contém esta interface.

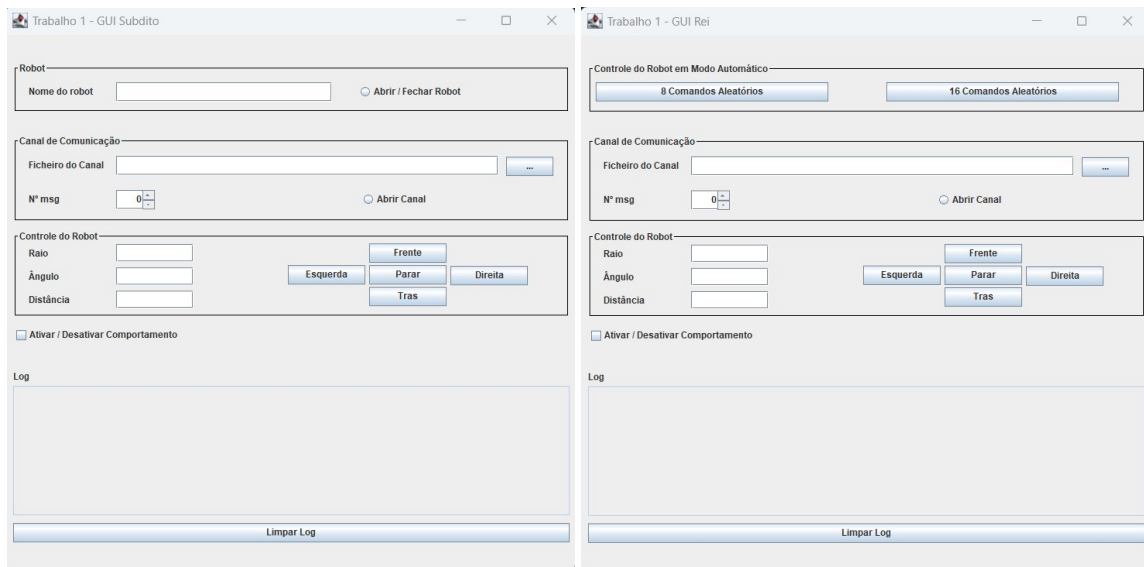


As componentes comuns vão ser então o Canal de Comunicação, o Controlo do Robot, e a Consola. O Canal de Comunicação, como o nome indica, vai fazer a comunicação entre os dois processos. O Controlo do Robot é autoexplicativo, ou seja podemos controlar as funções motoras do Robot como a direção e distância para onde ele se dirige, para que lado se vira ao darmos o raio e o ângulo e quando deve parar. A consola vai manter um registo de todos os comandos que realizamos ao longo da utilização do Robot, como por exemplo a sua conexão com o computador, os seus movimentos etc.

## 2.2 GUI\_Subdito e GUI\_Rei

Tendo a interface gráfica base desenvolvida basta-nos estender tanto à classe GUI\_Subdito como à classe GUI\_Rei a classe GUI\_Base e utiliza-mos o comando "super();" com o propósito de chamar o construtor da classe GUI\_Base. Este construtor define os parâmetros específicos de cada um dos objetos inseridos no JFrame da classe.

Tendo a classe sido chamada, podemos simplesmente inserir os conteúdos específicos da classe GUI\_Subdito e GUI\_Rei no espaço vazio da classe que estendem, ficando assim com os seguintes GUIs:



## 3 Conclusões

No desenvolvimento da GUI foram encontrados alguns problemas relacionados com os IDEs que estavam a ser utilizados. Os dois mais populares nesta altura da licenciatura são o IntelliJ e o Eclipse. O WindowBuilder de ambos é relativamente diferente e algumas informações passadas pelo docente não estavam a ser corretamente concretizadas no IntelliJ pois ao contrário do Eclipse, este constrói apenas variáveis e não os métodos associados a cada componente inserida dentro do JFrame. Um dos membros do grupo conseguiu realizar o que era esperado deste primeiro projeto utilizando o Eclipse e por isso foi feita a escolha pelos restantes membros de trocar de IDE para este mesmo, não só pela construção automática de métodos que governam os componentes, como também pela liberdade total da colocação destas mesmas componentes dentro do JFrame (ao contrário do IntelliJ que restringe o lugar onde estas podem ser inseridas).

## 4 Bibliografia

1. Folhas de Computação Física - Jorge Pais, 2023/2024

## 5 Código Java GUI\_Base, GUI\_Subdito, GUI\_Rei

```
1 Classe GUI_Base
2
3 package ptrabalho;
4
5 import java.awt.EventQueue;
6
7 import javax.swing.JFrame;
8 import javax.swing.JPanel;
9 import javax.swing.border.Border;
10 import javax.swing.border.EmptyBorder;
11 import javax.swing.BoxLayout;
12 import java.awt.GridLayout;
13
14 import javax.swing.BorderFactory;
15 import javax.swing.Box;
16 import java.awt.CardLayout;
17 import javax.swing.JTextField;
18 import java.awt.FlowLayout;
19 import javax.swing.JLabel;
20 import javax.swing.SwingConstants;
21 import java.awt.Font;
22 import javax.swing.JButton;
23 import javax.swing.JToggleButton;
24 import javax.swing.JSpinner;
25 import javax.swing.JRadioButton;
26 import javax.swing.JSeparator;
27 import javax.swing.JComboBox;
28 import javax.swing.JTextPane;
29 import java.awt.Panel;
30 import java.awt.Color;
31 import java.awt.Canvas;
32 import javax.swing.border.LineBorder;
33 import javax.swing.border.TitledBorder;
34 import javax.swing.JCheckBox;
35 import java.awt.event.ActionListener;
36 import java.awt.event.ActionEvent;
37
38 public class GUI_Base extends JFrame {
39
40     private JPanel contentPane;
41     private JTextField textField;
42     protected JTextField txtRaio;
43     protected JTextField txtAng;
44     protected JTextField txtDist;
45     private JTextField textField_4;
46     protected JButton btnFrt;
47     protected JButton btnEsq;
48     protected JButton btnDir;
49     protected JButton btnParar;
50     protected JButton btnTras;
51
52     /**
53      * Launch the application.
54      */
55     public static void main(String[] args) {
56         EventQueue.invokeLater(new Runnable() {
57             public void run() {
58                 try {
59                     GUI_Base frame = new GUI_Base();
```

```

60         frame.setVisible(true);
61     } catch (Exception e) {
62         e.printStackTrace();
63     }
64 }
65 });
66 }
67
68 /**
69  * Create the frame.
70  */
71 public GUI_Base() {
72     setTitle("Trabalho 1 - GUI Base");
73     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
74     setBounds(100, 100, 754, 691);
75     contentPane = new JPanel();
76     contentPane.setBorder(new EmptyBorder(100, 100, 100, 100));
77
78     setContentPane(contentPane);
79     contentPane.setLayout(null);
80
81     textField = new JTextField();
82     textField.setFont(new Font("Arial", Font.BOLD, 12));
83     textField.setBounds(143, 156, 494, 25);
84     contentPane.add(textField);
85     textField.setColumns(10);
86
87     JLabel lblNewLabel = new JLabel("Ficheiro do Canal");
88     lblNewLabel.setFont(new Font("Arial", Font.BOLD, 12));
89     lblNewLabel.setHorizontalAlignment(SwingConstants.LEFT);
90     lblNewLabel.setBounds(30, 156, 103, 25);
91     contentPane.add(lblNewLabel);
92
93     JButton btnNewButton = new JButton("...");
94     btnNewButton.setFont(new Font("Arial", Font.BOLD, 12));
95     btnNewButton.setBounds(647, 157, 61, 25);
96     contentPane.add(btnNewButton);
97
98     JLabel lblNMsg = new JLabel("N    msg");
99     lblNMsg.setHorizontalAlignment(SwingConstants.LEFT);
100    lblNMsg.setFont(new Font("Arial", Font.BOLD, 12));
101    lblNMsg.setBounds(30, 200, 103, 25);
102    contentPane.add(lblNMsg);
103
104    JSpinner spinner = new JSpinner();
105    spinner.setFont(new Font("Arial", Font.BOLD, 12));
106    spinner.setBounds(143, 200, 50, 25);
107    contentPane.add(spinner);
108
109    JRadioButton rdbtnNewRadioButton = new JRadioButton("Abrir
110    Canal");
111    rdbtnNewRadioButton.setFont(new Font("Arial", Font.BOLD, 12));
112    rdbtnNewRadioButton.setBounds(459, 200, 103, 25);
113    contentPane.add(rdbtnNewRadioButton);
114
115    Border simpb = BorderFactory.createLineBorder(new Color(0,0,0),1);
116    TitledBorder border_simp = BorderFactory.createTitledBorder(
117    simpb,"Canal de Comunicação");
118    JPanel panel = new JPanel();

```

```

117 panel.setName("Canal de Comunicação");
118 panel.setBorder(new LineBorder(new Color(0, 0, 0)));
119 panel.setBounds(10, 128, 719, 113);
120 panel.setBorder(border_simp);
121 contentPane.add(panel);
122
123 JLabel lblRaio = new JLabel("Raio");
124 lblRaio.setHorizontalAlignment(SwingConstants.LEFT);
125 lblRaio.setFont(new Font("Arial", Font.BOLD, 12));
126 lblRaio.setBounds(30, 270, 103, 25);
127 contentPane.add(lblRaio);
128
129 txtRaio = new JTextField();
130
131 txtRaio.setFont(new Font("Arial", Font.BOLD, 12));
132 txtRaio.setColumns(10);
133 txtRaio.setBounds(143, 270, 100, 23);
134 contentPane.add(txtRaio);
135
136 JLabel lblngulo = new JLabel("Ângulo");
137 lblngulo.setHorizontalAlignment(SwingConstants.LEFT);
138 lblngulo.setFont(new Font("Arial", Font.BOLD, 12));
139 lblngulo.setBounds(30, 300, 103, 25);
140 contentPane.add(lblngulo);
141
142 JLabel lblDistncia = new JLabel("Distância");
143 lblDistncia.setHorizontalAlignment(SwingConstants.LEFT);
144 lblDistncia.setFont(new Font("Arial", Font.BOLD, 12));
145 lblDistncia.setBounds(30, 330, 103, 25);
146 contentPane.add(lblDistncia);
147
148 txtAng = new JTextField();
149 txtAng.setFont(new Font("Arial", Font.BOLD, 12));
150 txtAng.setColumns(10);
151 txtAng.setBounds(143, 300, 100, 23);
152 contentPane.add(txtAng);
153
154 txtDist = new JTextField();
155 txtDist.setFont(new Font("Arial", Font.BOLD, 12));
156 txtDist.setColumns(10);
157 txtDist.setBounds(143, 330, 100, 23);
158 contentPane.add(txtDist);
159
160 btnParar = new JButton("Parar");
161 btnParar.setFont(new Font("Arial", Font.BOLD, 12));
162 btnParar.setBounds(470, 297, 100, 25);
163 contentPane.add(btnParar);
164
165 btnFrt = new JButton("Frente");
166 btnFrt.setFont(new Font("Arial", Font.BOLD, 12));
167 btnFrt.setBounds(470, 269, 100, 25);
168 contentPane.add(btnFrt);
169
170 btnDir = new JButton("Direita");
171 btnDir.setFont(new Font("Arial", Font.BOLD, 12));
172 btnDir.setBounds(575, 298, 100, 25);
173 contentPane.add(btnDir);
174
175 btnEsq = new JButton("Esquerda");
176 btnEsq.setFont(new Font("Arial", Font.BOLD, 12));

```



```

177     btnEsq.setBounds(365, 297, 100, 25);
178     contentPane.add(btnEsq);
179
180     btnTras = new JButton("Tras");
181     btnTras.setFont(new Font("Arial", Font.BOLD, 12));
182     btnTras.setBounds(470, 325, 100, 25);
183     contentPane.add(btnTras);
184
185     Border borda_cont_robot = BorderFactory.createLineBorder(new
Color(0,0,0),1);
186     TitledBorder borda1 = BorderFactory.createTitledBorder(
borda_cont_robot,"Controle do Robot");
187     JPanel panel_1 = new JPanel();
188     panel_1.setFont(new Font("Arial", Font.BOLD, 12));
189     panel_1.setName("Controle do Robot");
190     panel_1.setBorder(new LineBorder(new Color(0, 0, 0)));
191     panel_1.setBounds(10, 251, 719, 113);
192     panel_1.setBorder(borda1);
193     contentPane.add(panel_1);
194
195
196
197
198
199     JCheckBox chckbxNewCheckBox = new JCheckBox("Ativar / Desativar
Comportamento");
200     chckbxNewCheckBox.setFont(new Font("Arial", Font.BOLD, 12));
201     chckbxNewCheckBox.setBounds(10, 365, 262, 45);
202     contentPane.add(chckbxNewCheckBox);
203
204     JLabel lblLog = new JLabel("Log");
205     lblLog.setHorizontalAlignment(SwingConstants.LEFT);
206     lblLog.setFont(new Font("Arial", Font.BOLD, 12));
207     lblLog.setBounds(10, 429, 103, 25);
208     contentPane.add(lblLog);
209
210     textField_4 = new JTextField();
211     textField_4.setEditable(false);
212     textField_4.setFont(new Font("Arial", Font.BOLD, 12));
213     textField_4.setColumns(10);
214     textField_4.setBounds(10, 453, 719, 167);
215     contentPane.add(textField_4);
216
217     JButton btnNewButton_2_2 = new JButton("Limpar Log");
218     btnNewButton_2_2.setFont(new Font("Arial", Font.BOLD, 12));
219     btnNewButton_2_2.setBounds(10, 630, 719, 25);
220     contentPane.add(btnNewButton_2_2);
221 }
222 }
223
224 Classe GUI_Subdito
225
226 package ptrabalho;
227
228 import java.awt.EventQueue;
229
230 import javax.swing.BorderFactory;
231 import javax.swing.JFrame;
232 import javax.swing.JPanel;
233 import javax.swing.border.Border;

```

```

234 import javax.swing.border.EmptyBorder;
235 import javax.swing.border.LineBorder;
236 import javax.swing.border.TitledBorder;
237
238 import java.awt.Color;
239 import javax.swing.JLabel;
240 import java.awt.Font;
241 import java.awt.event.ActionEvent;
242 import java.awt.event.ActionListener;
243
244 import javax.swing.SwingConstants;
245 import javax.swing.JTextField;
246 import javax.swing.JRadioButton;
247
248 public class GUI_Subdito extends GUI_Base {
249
250     private JPanel contentPane;
251     private JTextField txtNome;
252     private App_Subdito app;
253
254     /**
255      * Launch the application.
256      */
257     public GUI_Subdito(App_Subdito app) {
258         super();
259         EventQueue.invokeLater(new Runnable() {
260             public void run() {
261                 try {
262                     init_GUI_Subdito();
263
264                 } catch (Exception e) {
265                     e.printStackTrace();
266                 }
267             }
268         });
269     }
270
271     /**
272      * Create the frame.
273      */
274     public void init_GUI_Subdito() {
275
276         setTitle("Trabalho 1 - GUI Subdito");
277         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
278         setBounds(100, 100, 754, 695);
279
280
281         JLabel lblNomeDoRobot = new JLabel("Nome do robot");
282         lblNomeDoRobot.setHorizontalAlignment(SwingConstants.LEFT);
283         lblNomeDoRobot.setFont(new Font("Arial", Font.BOLD, 12));
284         lblNomeDoRobot.setBounds(30, 61, 103, 25);
285         getContentPane().add(lblNomeDoRobot);
286
287         btnFrt.addActionListener(new ActionListener() {
288             public void actionPerformed(ActionEvent e) {
289                 app.retaFrente();
290             }
291         });
292
293         btnEsq.addActionListener(new ActionListener() {

```

```

294     public void actionPerformed(ActionEvent e) {
295         app.curvarEsquerda();
296     }
297 });
298
299 btnDir.addActionListener(new ActionListener() {
300     public void actionPerformed(ActionEvent e) {
301         app.curvarDireita();
302     }
303 });
304
305 btnTras.addActionListener(new ActionListener() {
306     public void actionPerformed(ActionEvent e) {
307         app.retaTras();
308     }
309 });
310
311 btnParar.addActionListener(new ActionListener() {
312     public void actionPerformed(ActionEvent e) {
313         app.parar();
314     }
315 });
316
317 JRadioButton rdbtnAbrirFechar = new JRadioButton("Abrir /
Fechar Robot");
318 rdbtnAbrirFechar.addActionListener(new ActionListener() {
319     public void actionPerformed(ActionEvent e) {
320
321     }
322 });
323
324 txtDist.addActionListener(new ActionListener() {
325     public void actionPerformed(ActionEvent e) {
326         app.getBD().setDist(Integer.parseInt(txtDist.getText()));
327     }
328 });
329
330 txtAng.addActionListener(new ActionListener() {
331     public void actionPerformed(ActionEvent e) {
332         app.getBD().setAng(Integer.parseInt(txtAng.getText()));
333     }
334 });
335
336 txtRaio.addActionListener(new ActionListener() {
337     public void actionPerformed(ActionEvent e) {
338         app.getBD().setRaio(Float.parseFloat(txtRaio.getText()));
339     }
340 });
341
342 txtNome = new JTextField();
343 txtNome.addActionListener(new ActionListener() {
344     public void actionPerformed(ActionEvent e) {
345         app.getBD().setNome(txtRaio.getText());
346     }
347 });
348
349
350 txtNome.setFont(new Font("Arial", Font.BOLD, 12));
351 txtNome.setColumns(10);
352 txtNome.setBounds(143, 61, 279, 25);

```

```

353     getContentPane().add(txtNome);
354
355
356
357     rdbtnAbrirFechar.setFont(new Font("Arial", Font.BOLD, 12));
358     rdbtnAbrirFechar.setBounds(455, 61, 158, 25);
359     getContentPane().add(rdbtnAbrirFechar);
360
361
362     Border bords_robot= BorderFactory.createLineBorder(new Color
(0,0,0),1);
363     TitledBorder borda_robot = BorderFactory.createTitledBorder(
bords_robot, "Robot");
364     JPanel panel = new JPanel();
365     panel.setName("Canal de Comunicação");
366     panel.setBorder(new LineBorder(new Color(0, 0, 0)));
367     panel.setBounds(10, 34, 719, 66);
368     panel.setBorder(borda_robot);
369     getContentPane().add(panel);
370     contentPane = new JPanel();
371     contentPane.setBorder(new EmptyBorder(100, 100,100, 100));
372
373     setVisible(true);
374 }
375
376 }
377
378 Classe GUI_Rei
379
380 package ptrabalho;
381
382 import java.awt.Color;
383 import java.awt.EventQueue;
384 import java.awt.Font;
385
386 import javax.swing.BorderFactory;
387 import javax.swing.JButton;
388 import javax.swing.JFrame;
389 import javax.swing.JPanel;
390 import javax.swing.border.Border;
391 import javax.swing.border.EmptyBorder;
392 import javax.swing.border.LineBorder;
393 import javax.swing.border.TitledBorder;
394
395 public class GUI_Rei extends GUI_Base {
396
397     private JPanel contentPane;
398
399     /**
400      * Launch the application.
401      */
402     public static void main(String[] args) {
403         EventQueue.invokeLater(new Runnable() {
404             public void run() {
405                 try {
406                     GUI_Rei frame = new GUI_Rei();
407                     frame.setVisible(true);
408                 } catch (Exception e) {
409                     e.printStackTrace();
410                 }

```

```

411     }
412     });
413 }
414
415 /**
416  * Create the frame.
417  */
418 public GUI_Rei() {
419     super();
420     setTitle("Trabalho 1 - GUI Rei");
421     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
422     setBounds(100, 100, 754, 697);
423
424     Border cost = BorderFactory.createLineBorder(new Color(0,0,0)
425 ,1);
426     TitledBorder borda_rei = BorderFactory.createTitledBorder(cost,
427 "Controle do Robot em Modo Automático");
428     JPanel panel_1_1 = new JPanel();
429     panel_1_1.setLayout(null);
430     panel_1_1.setName("Controle do Robot em Modo Automático");
431     panel_1_1.setBorder(new LineBorder(new Color(0, 0, 0)));
432     panel_1_1.setBounds(10, 34, 719, 66);
433     panel_1_1.setBorder(borda_rei);
434     getContentPane().add(panel_1_1);
435
436     JButton btnNewButton_2 = new JButton("8 Comandos Aleatórios");
437     btnNewButton_2.setFont(new Font("Arial", Font.BOLD, 12));
438     btnNewButton_2.setBounds(10, 26, 300, 25);
439     panel_1_1.add(btnNewButton_2);
440
441     JButton btnNewButton_2_1 = new JButton("16 Comandos Aleatórios"
442 );
443     btnNewButton_2_1.setFont(new Font("Arial", Font.BOLD, 12));
444     btnNewButton_2_1.setBounds(385, 26, 300, 25);
445     panel_1_1.add(btnNewButton_2_1);
446     contentPane = new JPanel();
447     contentPane.setBorder(new EmptyBorder(100, 100,100, 100));
448 }

```