

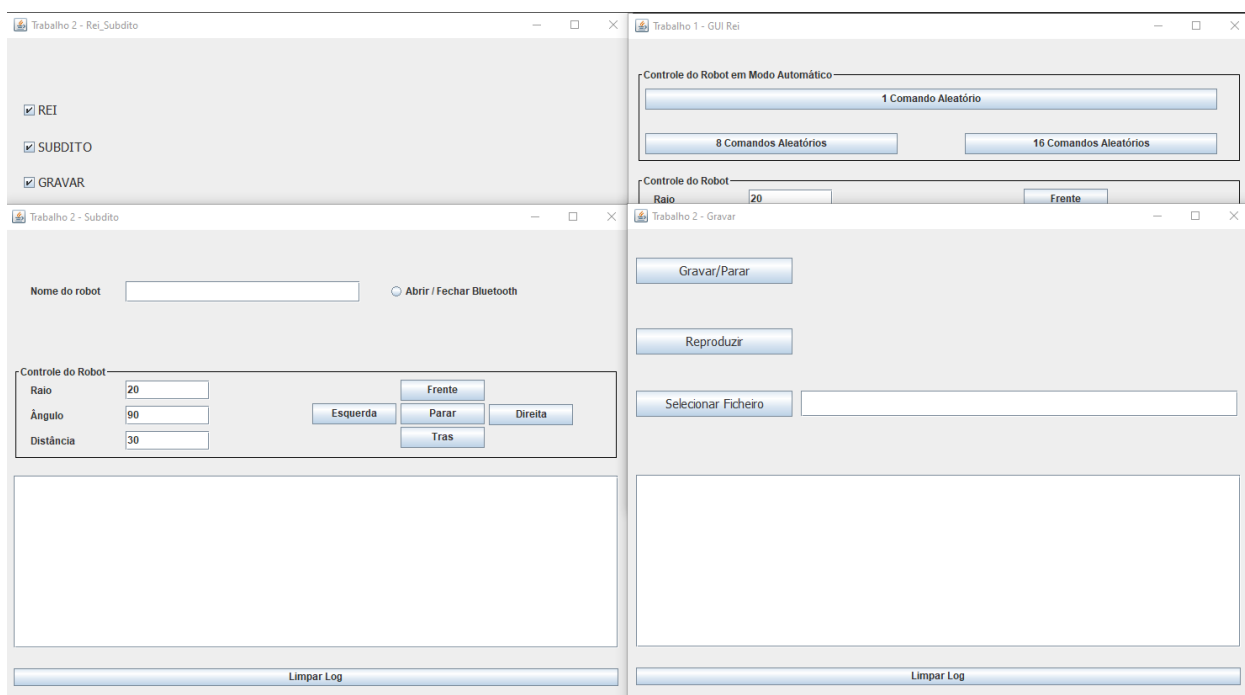


Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e Multimédia

Fundamentos de Sistemas Operativos - 2324SI

2º Trabalho Prático - Aula prática 1



Docente Carlos Carvalho

Realizado por (Grupo 7):

Diogo Santos 48626

Pedro Silva 48965

João Fonseca 49707

Conteúdo

1	Introdução	I
2	Desenvolvimento	I
2.1	GUIs_Base	I
2.2	GUI_Subdito, GUI_Rei e GUI_Gravar	II
2.3	GUI_Rei_Subdito	IV
3	Conclusões	IV
4	Bibliografia	IV
5	Código Java	V

1 Introdução

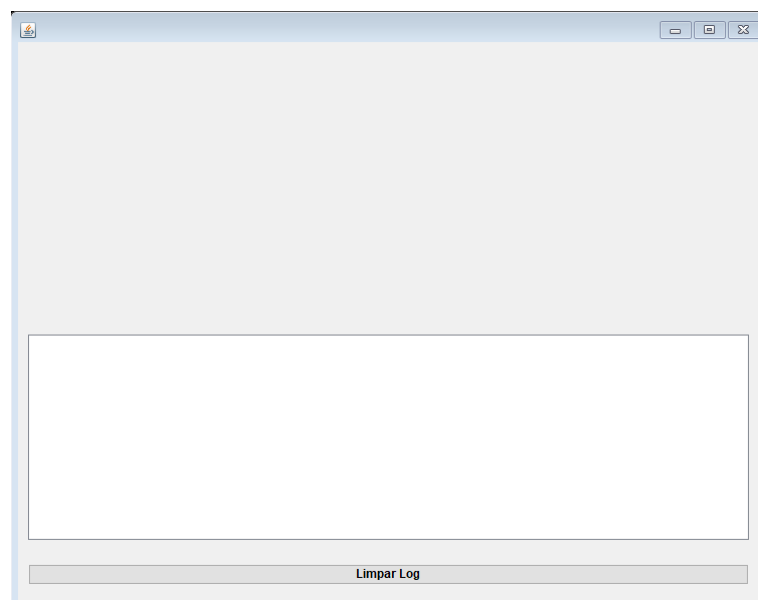
Neste segundo trabalho prático, Pretende-se o desenvolvimento de um processo REISUBDITO constituído por três tarefas JAVA: uma tarefa implementa o comportamento REI, outra tarefa implementa o comportamento SUBDITO, que conhece a API do robot, e a terceira tarefa, GRAVAR, permite gravar um conjunto de movimento do robot num ficheiro. O projeto está dividido em seis 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 o desenho das GUI's das três tarefas e do processo REISUBDITO. Começamos por construir uma GUI Base que vai ser estendida em todas as GUIs sucessivas. Esta vai ter o que é comum a todas ou seja os Logs e um botão para os limpar. Também criámos outra GUI Base mas mais especializada que a anterior onde vai estender às tarefas Rei e Súbdito e vai conter os botões para controlar o robot.

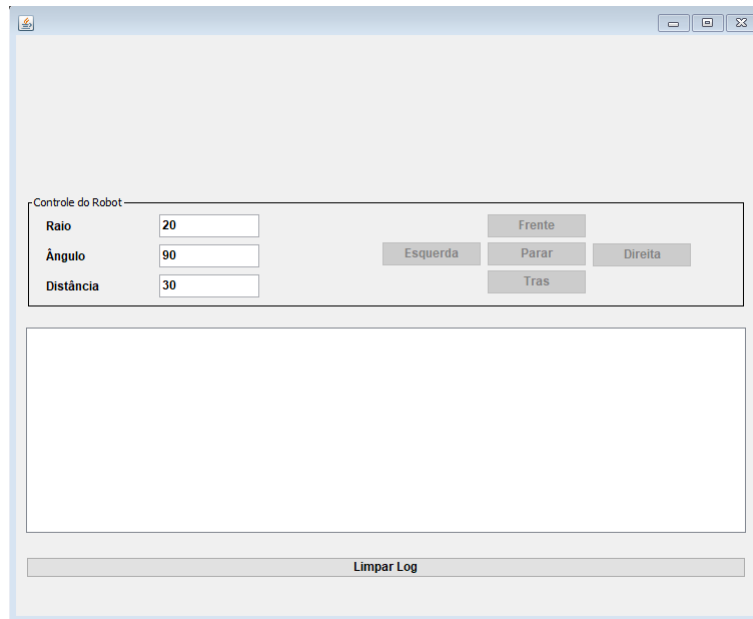
2 Desenvolvimento

2.1 GUIs_Base

Como foi dito anteriormente, esta primeira interface gráfica vai ter o objetivo de conter todos as componentes e funcionalidades comuns a todas as tarefas e processos deste trabalho 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 a caixa de texto onde vão ficar os Logs e um botão para a limpar. 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. Na GUI Base mais especializada vamos adicionar os botões para controlar o robot quer seja o Rei a enviar mensagens para o Súbdito ou o Súbdito a comunicar diretamente com o robot.

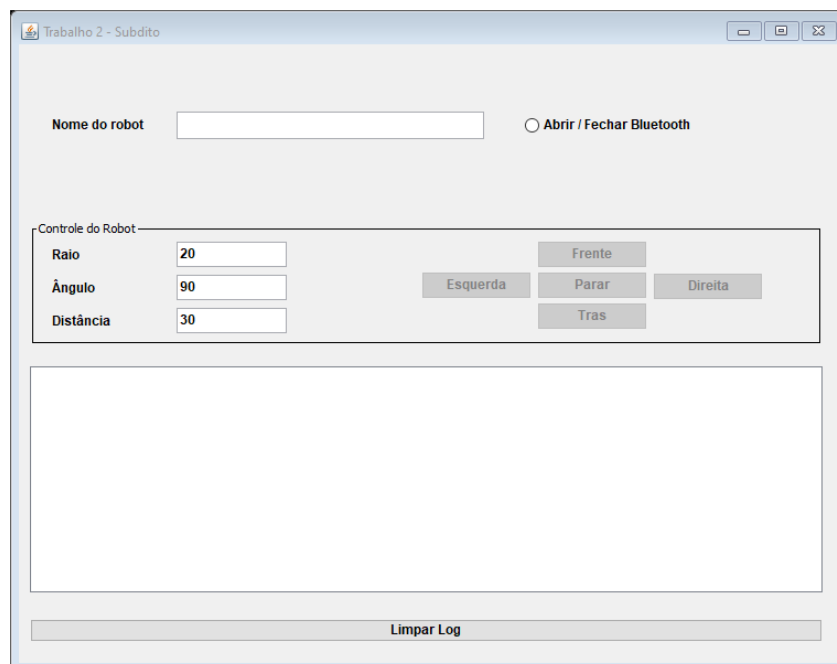


O Controlo do Robot é autoexplicativo, ou seja podemos controlar as funções motoras do Robot como a direção e distância para onde ele se dirige, para que lado se vira ao darmos o raio e o ângulo e quando deve parar.

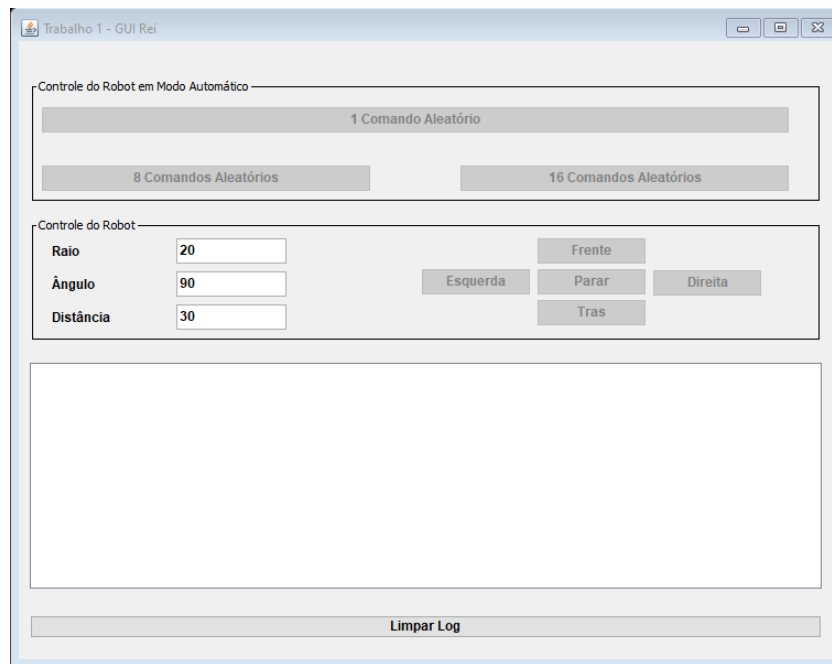
2.2 GUI_Subdito, GUI_Rei e GUI_Gravar

Tendo a interface gráfica base desenvolvida basta-nos estender às classes GUI_Subdito e classe GUI_Rei a classe GUI_BaseRS (a mais especializada) e utiliza-mos o comando "super();" com o propósito de chamar o construtor da classe pai.

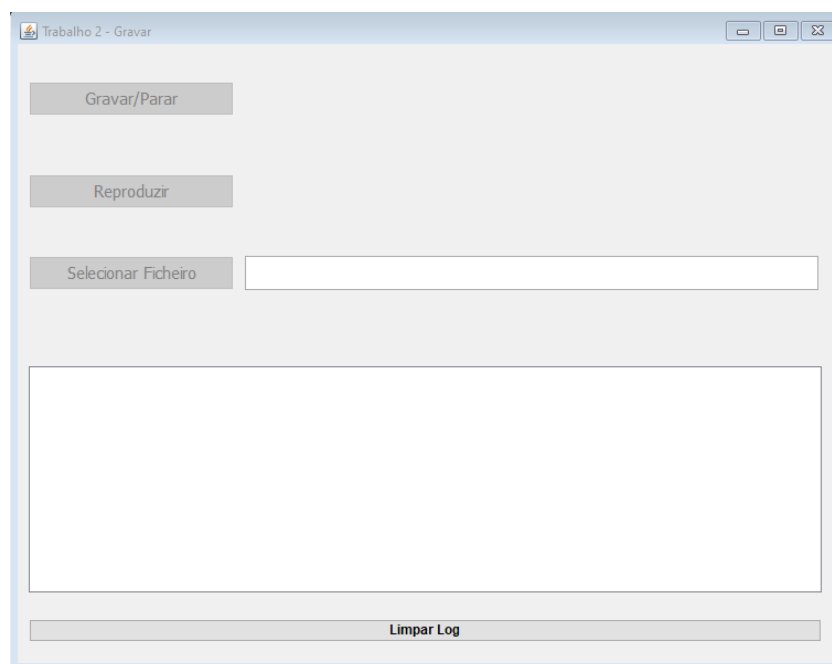
Tendo a classe sido chamada, podemos simplesmente inserir os conteúdos específicos da classe GUI_Subdito e GUI_Rei no espaço vazio da classe que estendem, ficando assim com os seguintes GUIs:



Na GUI_Subdito adicionámos os campos que nos permitirão fazer a ligação com o robot. Na GUI_Rei:



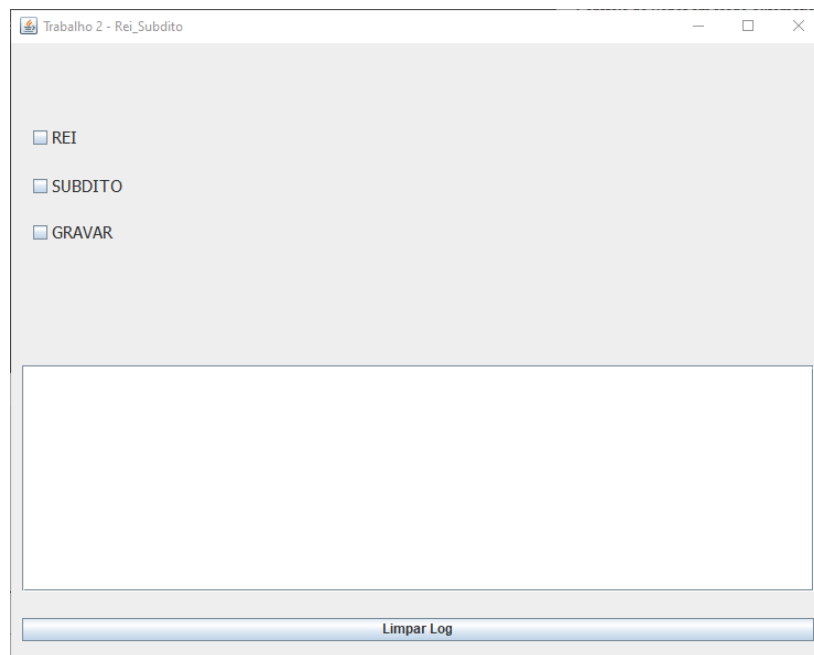
Adicionámos a possibilidade do utilizador enviar comandos aleatórios, podendo ser 1, 8 ou 16 comandos de cada vez. Na GUI.Gravar estendemos a GUI.Base e acrescentámos os seguintes campos:



Gravar/Parar que permite gravar ou parar a gravação num ficheiro, uma sequência de movimentos enviados para o robot, para posterior reprodução. Reproduzir que permite a reprodução de uma sequência de movimentos gravados num ficheiro. Durante uma reprodução não é permitido que os outros comportamentos enviem comandos para o robot. Definir o nome do ficheiro que permite definir o nome do ficheiro para gravação ou reprodução.

2.3 GUI_Rei_Subdito

Por último tivemos que desenvolver a GUI do processo Rei Súdito. Vamos estender outra vez a GUI_Base e desta vez acrescentamos 3 checkboxes que vão ligar os comportamentos das nossas tarefas:



Por enquanto, como ainda não desenvolvemos as tarefas estes botões vão apenas ativar ou desativar os botões correspondentes a cada uma destas tarefas.

3 Conclusões

No desenvolvimento destas GUIs tínhamos começado por apenas utilizar uma GUI_Base mas como observámos que existiria muito espaço livre e falta de controlo do utilizador nas tarefas Rei e Súdito decidimos criar mais uma GUI Base que permitisse uma maior gestão do robot ao acrescentarmos os botões de direção. Assim ficámos com GUIs com melhor aspeto e mais robustas.

4 Bibliografia

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

5 Código Java

```
1 Classe GUI_Base
2
3 package ptrabalho;
4
5 import java.awt.EventQueue;
6 import java.awt.Font;
7
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JPanel;
11 import javax.swing.border.EmptyBorder;
12 import javax.swing.JTextArea;
13 import java.awt.ScrollPane;
14 import java.awt.event.ActionEvent;
15 import java.awt.event.ActionListener;
16
17 import javax.swing.JToggleButton;
18 import javax.swing.JScrollPane;
19 import javax.swing.GroupLayout;
20 import javax.swing.GroupLayout.Alignment;
21
22 public class GUI_Base extends JFrame {
23
24     protected JPanel contentPane;
25
26     public GUI_Base() {
27         //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28         setBounds(0, 0, 758, 597);
29         contentPane = new JPanel();
30         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
31
32         setContentPane(contentPane);
33         contentPane.setLayout(null);
34
35         JTextArea txtLog = new JTextArea();
36         txtLog.setBounds(1, 1, 719, 203);
37         contentPane.add(txtLog);
38
39         JScrollPane scrollPane = new JScrollPane(txtLog);
40         scrollPane.setBounds(10, 292, 719, 205);
41
42         JButton btnLimpaLog = new JButton("Limpar Log");
43         btnLimpaLog.setBounds(10, 521, 719, 21);
44         btnLimpaLog.setFont(new Font("Arial", Font.BOLD, 12));
45         contentPane.add(scrollPane);
46         contentPane.add(btnLimpaLog);
47
48         btnLimpaLog.addActionListener(new ActionListener()
49         {
50             public void actionPerformed(ActionEvent e)
51             {
52                 txtLog.setText("");
53             }
54         });
55     }
56 }
57
58 Classe GUI_BaserS
59
```

```

60 package ptrabalho;
61
62 import java.awt.Color;
63 import java.awt.Font;
64
65 import javax.swing.BorderFactory;
66 import javax.swing.JButton;
67 import javax.swing.JLabel;
68 import javax.swing.JPanel;
69 import javax.swing.JTextField;
70 import javax.swing.SwingConstants;
71 import javax.swing.border.Border;
72 import javax.swing.border.LineBorder;
73 import javax.swing.border.TitledBorder;
74
75 public class GUI_BaseRS extends GUI_Base{
76
77     protected JPanel contentPane;
78     protected JTextField txtRaio;
79     protected JTextField txtAng;
80     protected JTextField txtDist;
81     protected JTextField txtFile;
82     protected JButton btnFrt;
83     protected JButton btnEsq;
84     protected JButton btnDir;
85     protected JButton btnParar;
86     protected JButton btnTras;
87
88     public GUI_BaseRS() {
89
90         Border borda_cont_robot = BorderFactory.createLineBorder(new
Color(0,0,0),1);
91         TitledBorder borda1 = BorderFactory.createTitledBorder(
borda_cont_robot,"Controle do Robot");
92
93         JLabel lblRaio = new JLabel("Raio");
94         lblRaio.setBounds(30, 179, 103, 25);
95         getContentPane().add(lblRaio);
96         lblRaio.setHorizontalAlignment(SwingConstants.LEFT);
97         lblRaio.setFont(new Font("Arial", Font.BOLD, 12));
98
99         txtRaio = new JTextField();
100         txtRaio.setBounds(143, 179, 100, 23);
101         getContentPane().add(txtRaio);
102         txtRaio.setText("20");
103
104         txtRaio.setFont(new Font("Arial", Font.BOLD, 12));
105         txtRaio.setColumns(10);
106
107         JLabel lblngulo = new JLabel("Ângulo");
108         lblngulo.setBounds(30, 209, 103, 25);
109         getContentPane().add(lblngulo);
110         lblngulo.setHorizontalAlignment(SwingConstants.LEFT);
111         lblngulo.setFont(new Font("Arial", Font.BOLD, 12));
112
113         JLabel lblDistncia = new JLabel("Distância");
114         lblDistncia.setBounds(30, 239, 103, 25);
115         getContentPane().add(lblDistncia);
116         lblDistncia.setHorizontalAlignment(SwingConstants.LEFT);
117         lblDistncia.setFont(new Font("Arial", Font.BOLD, 12));

```



```

118
119     txtAng = new JTextField();
120     txtAng.setBounds(143, 209, 100, 23);
121     getContentPane().add(txtAng);
122     txtAng.setText("90");
123     txtAng.setFont(new Font("Arial", Font.BOLD, 12));
124     txtAng.setColumns(10);
125
126     txtDist = new JTextField();
127     txtDist.setBounds(143, 239, 100, 23);
128     getContentPane().add(txtDist);
129     txtDist.setText("30");
130     txtDist.setFont(new Font("Arial", Font.BOLD, 12));
131     txtDist.setColumns(10);
132
133     btnParar = new JButton("Parar");
134     btnParar.setBounds(470, 206, 100, 25);
135     getContentPane().add(btnParar);
136     btnParar.setEnabled(false);
137     btnParar.setFont(new Font("Arial", Font.BOLD, 12));
138
139     btnFrt = new JButton("Frente");
140     btnFrt.setBounds(470, 178, 100, 25);
141     getContentPane().add(btnFrt);
142     btnFrt.setEnabled(false);
143     btnFrt.setFont(new Font("Arial", Font.BOLD, 12));
144
145     btnDir = new JButton("Direita");
146     btnDir.setBounds(575, 207, 100, 25);
147     getContentPane().add(btnDir);
148     btnDir.setEnabled(false);
149     btnDir.setFont(new Font("Arial", Font.BOLD, 12));
150
151     btnEsq = new JButton("Esquerda");
152     btnEsq.setBounds(365, 206, 100, 25);
153     getContentPane().add(btnEsq);
154     btnEsq.setEnabled(false);
155     btnEsq.setFont(new Font("Arial", Font.BOLD, 12));
156
157     btnTras = new JButton("Tras");
158     btnTras.setBounds(470, 234, 100, 25);
159     getContentPane().add(btnTras);
160     btnTras.setEnabled(false);
161     btnTras.setFont(new Font("Arial", Font.BOLD, 12));
162     JPanel panel_1 = new JPanel();
163     panel_1.setBounds(10, 160, 719, 113);
164     getContentPane().add(panel_1);
165     panel_1.setFont(new Font("Arial", Font.BOLD, 12));
166     panel_1.setName("Controle do Robot");
167     panel_1.setBorder(new LineBorder(new Color(0, 0, 0)));
168     panel_1.setBorder(borda1);
169 }
170
171 protected void start()
172 {
173     btnFrt.setEnabled(true);
174     btnEsq.setEnabled(true);
175     btnParar.setEnabled(true);
176     btnDir.setEnabled(true);
177     btnTras.setEnabled(true);

```

```

178     }
179
180     protected void off()
181     {
182         btnFrt.setEnabled(false);
183         btnEsq.setEnabled(false);
184         btnParar.setEnabled(false);
185         btnDir.setEnabled(false);
186         btnTras.setEnabled(false);
187     }
188
189 }
190
191
192 Classe Subdito
193
194 package ptrabalho;
195
196 import java.awt.EventQueue;
197
198 import javax.swing.BorderFactory;
199 import javax.swing.JFileChooser;
200 import javax.swing.JFrame;
201 import javax.swing.JPanel;
202 import javax.swing.border.Border;
203 import javax.swing.border.EmptyBorder;
204 import javax.swing.border.LineBorder;
205 import javax.swing.border.TitledBorder;
206
207 import java.awt.Color;
208 import javax.swing.JLabel;
209 import java.awt.Font;
210 import java.awt.event.ActionEvent;
211 import java.awt.event.ActionListener;
212 import java.awt.event.WindowAdapter;
213 import java.awt.event.WindowEvent;
214
215 import javax.swing.SwingConstants;
216 import javax.swing.JTextField;
217 import javax.swing.JRadioButton;
218 import javax.swing.JTextArea;
219
220 public class Subdito extends GUI_BaseRS
221 {
222
223     private JPanel contentPane;
224     private JTextField txtNome;
225
226
227     /**
228      * Launch the application.
229      */
230     public Subdito()
231     {
232         super();
233         EventQueue.invokeLater(new Runnable()
234         {
235             public void run()
236             {
237                 try

```

```

238     {
239         init_Subdito();
240
241     } catch (Exception e)
242     {
243         e.printStackTrace();
244     }
245 }
246 });
247 }
248
249 /**
250  * Create the frame.
251  */
252 public void init_Subdito()
253 {
254     setTitle("Trabalho 2 - Subdito");
255
256     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
257     setBounds(50, 400, 760, 600);
258
259
260     JLabel lblNomeDoRobot = new JLabel("Nome do robot");
261     lblNomeDoRobot.setHorizontalAlignment(SwingConstants.LEFT);
262     lblNomeDoRobot.setFont(new Font("Arial", Font.BOLD, 12));
263     lblNomeDoRobot.setBounds(30, 61, 103, 25);
264     getContentPane().add(lblNomeDoRobot);
265
266     JRadioButton rdbtnAbrirFecharBlt = new JRadioButton("Abrir /
Fechar Bluetooth");
267     /*rdbtnAbrirFecharBlt.addActionListener(new ActionListener()
268     {
269         public void actionPerformed(ActionEvent e) {
270             if (bd.isLigado())
271             {
272                 System.out.println("Desligando...");
273                 bd.getRobot().CloseEV3();
274                 bd.setLigado(false);
275                 if(rdbtnAtivarDesativarComp.isSelected() ) {
276                     btnFrt.setEnabled(false);
277                     btnEsq.setEnabled(false);
278                     btnParar.setEnabled(false);
279                     btnDir.setEnabled(false);
280                     btnTras.setEnabled(false);
281                 }
282
283             } else
284             {
285                 System.out.println("Ligando...");
286                 bd.setLigado(bd.getRobot().OpenEV3(bd.getNome()));
287                 System.out.println(bd.isLigado());
288                 if (!bd.isLigado())
289                     rdbtnAbrirFecharBlt.setSelected(false);
290                 if(rdbtnAtivarDesativarComp.isSelected() ) {
291                     btnFrt.setEnabled(true);
292                     btnEsq.setEnabled(true);
293                     btnParar.setEnabled(true);
294                     btnDir.setEnabled(true);
295                     btnTras.setEnabled(true);}
296

```

```

297
298         //bd.setLigado(true);
299     }
300 }
301 });*/
302
303 txtNome = new JTextField();
304 /*txtNome.addActionListener(new ActionListener()
305 {
306     public void actionPerformed(ActionEvent e)
307     {
308         bd.setNome(txtNome.getText());
309         txtLog.append(" O nome do Robot é " + bd.getNome() + "\n");
310     }
311 });*/
312
313
314 txtNome.setFont(new Font("Arial", Font.BOLD, 12));
315 txtNome.setColumns(10);
316 txtNome.setBounds(143, 61, 279, 25);
317 getContentPane().add(txtNome);
318
319
320
321 rdbtnAbrirFecharBlt.setFont(new Font("Arial", Font.BOLD, 12));
322 rdbtnAbrirFecharBlt.setBounds(455, 61, 188, 25);
323 getContentPane().add(rdbtnAbrirFecharBlt);
324
325 setVisible(true);
326 }
327
328 }
329
330
331 Classe Rei
332
333 package ptrabalho;
334
335 import java.awt.Color;
336 import java.awt.EventQueue;
337 import java.awt.Font;
338 import java.awt.event.ActionEvent;
339 import java.awt.event.ActionListener;
340 import java.util.Random;
341
342 import javax.swing.BorderFactory;
343 import javax.swing.JButton;
344 import javax.swing.JFrame;
345 import javax.swing.JPanel;
346 import javax.swing.border.Border;
347 import javax.swing.border.EmptyBorder;
348 import javax.swing.border.LineBorder;
349 import javax.swing.border.TitledBorder;
350
351 public class Rei extends GUI_BaseRS
352 {
353     private int id = 0;
354     protected JButton btn8com;
355     protected JButton btn16com;
356     protected JButton btn1com;

```

```

357
358
359 public Rei()
360 {
361     super();
362     EventQueue.invokeLater(new Runnable()
363     {
364         public void run()
365         {
366             try
367             {
368                 init_Rei();
369
370             } catch (Exception e)
371             {
372                 e.printStackTrace();
373             }
374         }
375     });
376 }
377
378 /**
379  * Create the frame.
380  */
381 public void init_Rei()
382 {
383     setTitle("Trabalho 1 - GUI Rei");
384     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
385     setBounds(1000, 0, 760, 600);
386
387     Border cost = BorderFactory.createLineBorder(new Color(0,0,0)
388     ,1);
389     TitledBorder borda_rei = BorderFactory.createTitledBorder(cost,
390     "Controle do Robot em Modo Automático");
391     JPanel panel_1_1 = new JPanel();
392     panel_1_1.setLayout(null);
393     panel_1_1.setName("Controle do Robot em Modo Automático");
394     panel_1_1.setBorder(new LineBorder(new Color(0, 0, 0)));
395     panel_1_1.setBounds(10, 34, 719, 113);
396     panel_1_1.setBorder(borda_rei);
397     getContentPane().add(panel_1_1);
398
399     btn8com = new JButton("8 Comandos Aleatórios");
400     btn8com.setEnabled(false);
401     btn8com.setFont(new Font("Arial", Font.BOLD, 12));
402     btn8com.setBounds(10, 78, 300, 25);
403     panel_1_1.add(btn8com);
404
405     btn16com = new JButton("16 Comandos Aleatórios");
406     btn16com.setEnabled(false);
407     btn16com.setFont(new Font("Arial", Font.BOLD, 12));
408     btn16com.setBounds(390, 78, 300, 25);
409     panel_1_1.add(btn16com);
410
411     btn1com = new JButton("1 Comando Aleatório");
412     btn1com.setFont(new Font("Arial", Font.BOLD, 12));
413     btn1com.setEnabled(false);
414     btn1com.setBounds(10, 25, 680, 25);
415     panel_1_1.add(btn1com);
416     contentPane = new JPanel();

```

```

415     contentPane.setBorder(new EmptyBorder(100, 100,100, 100));
416
417
418     /*btn8com.addActionListener(new ActionListener()
419     {
420         public void actionPerformed(ActionEvent e)
421         {
422             for (int i = 0; i<8; i++)
423             {
424                 msg = gerarRandomMensagem();
425                 bd.addMensagem(msg);
426
427             }
428         }
429     });
430
431     btn16com.addActionListener(new ActionListener()
432     {
433         public void actionPerformed(ActionEvent e)
434         {
435             for (int i = 0; i<16; i++)
436             {
437                 msg = gerarRandomMensagem();
438                 bd.addMensagem(msg);
439             }
440
441         }
442     });*/
443
444     setVisible(true);
445 }
446
447 protected void start()
448 {
449     super.start();
450     btn1com.setEnabled(true);
451     btn16com.setEnabled(true);
452     btn8com.setEnabled(true);
453 }
454
455 protected void off()
456 {
457     super.off();
458     btn1com.setEnabled(false);
459     btn16com.setEnabled(false);
460     btn8com.setEnabled(false);
461 }
462 }
463
464
465 Classe Gravar
466
467 package ptrabalho;
468
469 import java.awt.EventQueue;
470 import java.awt.event.ActionEvent;
471 import java.awt.event.ActionListener;
472
473 import javax.swing.JFileChooser;
474 import javax.swing.JFrame;

```

```

475 import javax.swing.JPanel;
476 import javax.swing.border.EmptyBorder;
477 import javax.swing.JToggleButton;
478 import java.awt.Font;
479 import javax.swing.JTextField;
480 import javax.swing.JButton;
481
482 public class Gravar extends GUI_Base {
483
484     private JPanel contentPane;
485     private JTextField textField;
486     protected JToggleButton tglGravar;
487     protected JToggleButton tglReproduzir;
488     protected JButton btnFile;
489
490
491     public Gravar()
492     {
493         super();
494         EventQueue.invokeLater(new Runnable()
495         {
496             public void run()
497             {
498                 try
499                 {
500                     init_Gravar();
501
502                 } catch (Exception e)
503                 {
504                     e.printStackTrace();
505                 }
506             }
507         });
508     }
509
510     public void init_Gravar() {
511
512         setTitle("Trabalho 2 - Gravar");
513         //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
514         setBounds(1000, 400, 760, 600);
515         textField = new JTextField();
516         textField.setBounds(206, 192, 520, 31);
517         getContentPane().add(textField);
518         textField.setColumns(10);
519
520         tglGravar = new JToggleButton("Gravar/Parar");
521         tglGravar.setEnabled(false);
522         tglGravar.setFont(new Font("Tahoma", Font.PLAIN, 15));
523         tglGravar.setBounds(10, 34, 186, 31);
524         getContentPane().add(tglGravar);
525
526         tglReproduzir = new JToggleButton("Reproduzir");
527         tglReproduzir.setEnabled(false);
528         tglReproduzir.setFont(new Font("Tahoma", Font.PLAIN, 15));
529         tglReproduzir.setBounds(10, 118, 186, 31);
530         getContentPane().add(tglReproduzir);
531
532         btnFile = new JButton("Selecionar Ficheiro");
533         btnFile.setEnabled(false);
534         btnFile.setFont(new Font("Tahoma", Font.PLAIN, 15));

```

```

535     btnFile.setBounds(10, 192, 186, 31);
536     getContentPane().add(btnFile);
537     btnFile.addActionListener(new ActionListener()
538     {
539         public void actionPerformed(ActionEvent e)
540         {
541             JFileChooser fileChooser = new JFileChooser(System.
542             getProperty("user.dir"));
543             if (fileChooser.showSaveDialog(null) ==
544             JFileChooser.APPROVE_OPTION)
545             {
546                 String file = fileChooser.getSelectedFile().
547                 getAbsolutePath();
548                 textField.setText(file);
549             }
550         }
551     });
552     setVisible(true);
553 }
554
555 protected void start()
556 {
557     btnFile.setEnabled(true);
558     tglReproduzir.setEnabled(true);
559     tglGravar.setEnabled(true);
560 }
561
562 protected void off()
563 {
564     btnFile.setEnabled(false);
565     tglReproduzir.setEnabled(false);
566     tglGravar.setEnabled(false);
567 }
568
569 Classe Rei_Subdito
570
571
572 package ptrabalho;
573
574 import java.awt.EventQueue;
575
576 import javax.swing.JFrame;
577 import javax.swing.JPanel;
578 import javax.swing.border.EmptyBorder;
579 import javax.swing.JCheckBox;
580 import java.awt.Font;
581 import java.awt.event.ActionEvent;
582 import java.awt.event.ActionListener;
583
584 import javax.swing.SwingConstants;
585
586 public class Rei_Subdito extends GUI_Base {
587
588     private JPanel contentPane;
589     private Rei gui_Rei;
590     private Subdito gui_Subdito;
591     private Gravar gui_Gravar;

```



```

592
593
594     public Rei_Subdito()
595     {
596         super();
597         EventQueue.invokeLater(new Runnable()
598         {
599             public void run()
600             {
601                 try
602                 {
603                     init_Rei_Subdito();
604
605                 } catch (Exception e)
606                 {
607                     e.printStackTrace();
608                 }
609             }
610         });
611     }
612
613
614     public void init_Rei_Subdito() {
615
616         setTitle("Trabalho 2 - Rei_Subdito");
617         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
618         setBounds(0, 0, 754, 600);
619
620         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
621         getContentPane().setLayout(null);
622         JCheckBox chckbxRei = new JCheckBox("REI");
623         chckbxRei.setHorizontalAlignment(SwingConstants.LEFT);
624         chckbxRei.setFont(new Font("Tahoma", Font.PLAIN, 15));
625         chckbxRei.setBounds(16, 56, 97, 59);
626         getContentPane().add(chckbxRei);
627         JCheckBox chckbxSubdito = new JCheckBox("SUBDITO");
628         chckbxSubdito.setHorizontalAlignment(SwingConstants.LEFT);
629         chckbxSubdito.setFont(new Font("Tahoma", Font.PLAIN, 15));
630         chckbxSubdito.setBounds(16, 100, 195, 59);
631         getContentPane().add(chckbxSubdito);
632         JCheckBox chckbxGravar = new JCheckBox("GRAVAR");
633         chckbxGravar.setHorizontalAlignment(SwingConstants.LEFT);
634         chckbxGravar.setFont(new Font("Tahoma", Font.PLAIN, 15));
635         chckbxGravar.setBounds(16, 142, 108, 59);
636         getContentPane().add(chckbxGravar);
637         gui_Rei = new Rei();
638         gui_Subdito = new Subdito();
639         gui_Gravar = new Gravar();
640
641         chckbxRei.addActionListener(new ActionListener()
642         {
643             public void actionPerformed(ActionEvent e)
644             {
645                 if(chckbxRei.isSelected())
646                     gui_Rei.start();
647                 else
648                     gui_Rei.off();
649
650                 //txtLog.append(" O robot avançou " + bd.getDist() + "\n");

```

```

651     }
652   });
653
654   chkcbxSubdito.addActionListener(new ActionListener()
655   {
656     public void actionPerformed(ActionEvent e)
657     {
658       if(chkcbxSubdito.isSelected())
659         gui_Subdito.start();
660       else
661         gui_Subdito.off();
662
663       //txtLog.append(" O robot avançou " + bd.getDist() + "\n");
664     }
665   });
666
667   chkcbxGravar.addActionListener(new ActionListener()
668   {
669     public void actionPerformed(ActionEvent e)
670     {
671       if(chkcbxGravar.isSelected())
672         gui_Gravar.start();
673       else
674         gui_Gravar.off();
675
676       //txtLog.append(" O robot avançou " + bd.getDist() + "\n");
677     }
678   });
679
680   setVisible(true);
681 }
682
683 }
684
685
686
687 Classe App_Rei_Subdito
688
689 package ptrabalho;
690
691 public class App_Rei_Subdito
692 {
693     @SuppressWarnings("unused")
694     private Rei_Subdito gui;
695     private int state = 2;
696     private int counter = 0;
697     private final int escreverMensagem = 1;
698     private final int dormir = 2;
699     private final int esperarTempoExecucao = 3;
700
701     public App_Rei_Subdito(String[] args)
702     {
703         gui = new Rei_Subdito();
704     }
705
706
707
708     public void run() throws InterruptedException

```

```
709     {
710         while(true) {
711             //System.out.println("sai");
712             Thread.sleep(100);
713         }
714
715
716     }
717
718     public static void main(String[] args) throws
719     InterruptedException
720     {
721         App_Rei_Subdito app = new App_Rei_Subdito(args);
722         System.out.println("A aplicação começou.");
723         app.run();
724         System.out.println("A aplicação terminou.");
725     }
726
727 }
```