

INSTITUTO SUPERIOR de ENGENHARIA de LISBOA

Licenciatura em Engenharia Informática e Multimédia

1º Semestre Letivo 2025/2026

Fundamentos de Sistemas Operativos

1º Trabalho Prático

Objetivos:

- Desenho duma GUI em *swing* utilizando o editor gráfico *WindowBuilder*.
- Desenho de diagramas de classes e de estado utilizando o *MagicDraw*.
- Desenvolvimento de uma aplicação multitarefa em Java.
- Sincronização entre tarefas Java no acesso a um recurso físico.

Pretende-se o desenvolvimento de uma aplicação para controlo de um robot constituída por uma **GUI** (Graphical User Interface) e uma tarefa Java, designada **Movimento Aleatório**.

A **GUI** deve permitir um utilizador interagir com o robot, ativar ou desativar a tarefa **Movimento Aleatório** e ter uma consola para visualizar o que o robot está a fazer.

A **GUI** deve ter a aparência sugerida na figura 1.

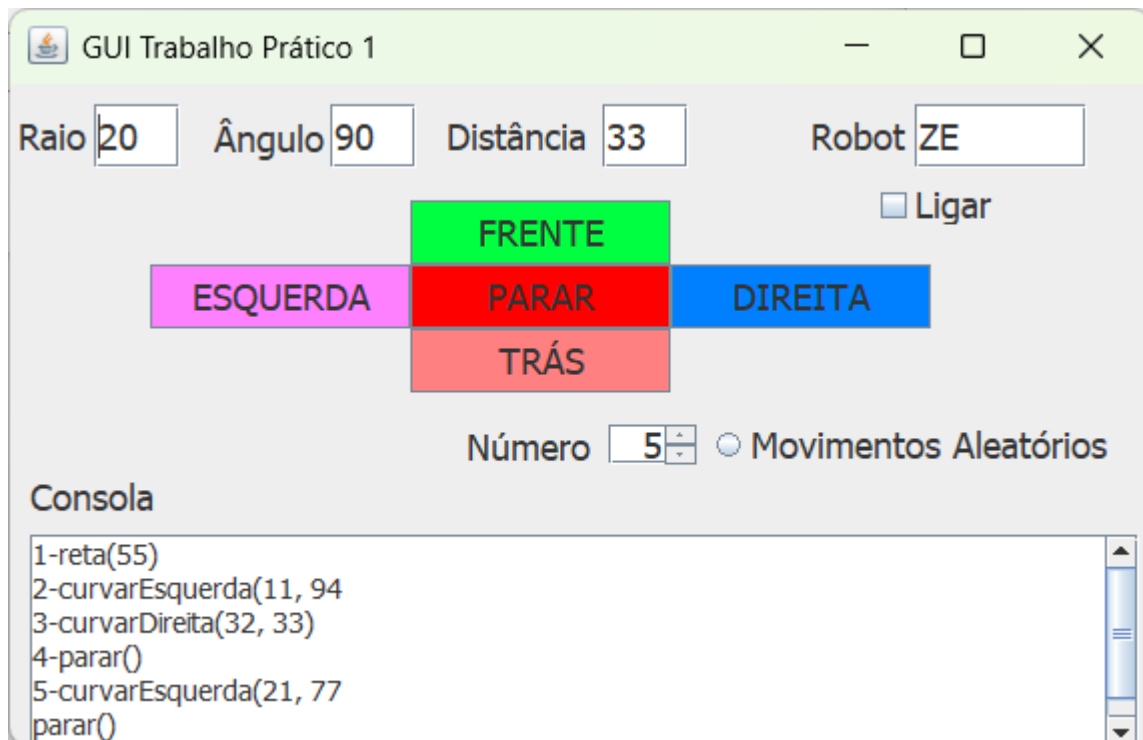


Figura 1 – Interface gráfica sugerida.

Os componentes da **GUI** são os seguintes:

Raio: **JLabel** – identificação do propósito da **JTextField** em frente;

“20”: **JTextField** – onde edita o valor do raio a utilizar;

Ângulo: **JLabel** – identificação do propósito da **JTextField** em frente;

“90”: **JTextField** – onde edita o valor do ângulo a utilizar;

Distância: **JLabel** – identificação do propósito da **JTextField** em frente;

“33”: **JTextField** – onde edita o valor da distância a utilizar;

Robot: **JLabel** – identificação do propósito da **JTextField** em frente;

“ZE”: **JTextField** – onde edita o nome do robot a utilizar;

Ligar: **JCheckBox** – onde permite a comunicação eliação via *bluetooth* ao robot;

FRENTE, ESQUERDA, PARAR, DIREITA, TRÁS: **JButton** – permitem comandar o robot;

Número: **JLabel** – identificação do propósito do **JSpinner**;

“5”: **JSpinner** – indica o número de comandos aleatórios que a tarefa **Movimentos Aleatórios** deve gerar e enviar para o robot. Valores entre 1 e 16;

Movimentos Aleatórios: **JRadioButton** – permite ativar ou desativar o funcionamento da tarefa **Movimentos Aleatórios**.

Consola: **JLabel** – identificação do componente abaixo;

Texto: **JArea** contida num **JScrollPane** – área de texto onde a aplicação deve afixar todas as mensagens informativas do que o robot está a fazer. Também pode servir para afixar as mensagens de *debugging* do programa.

A tarefa **Movimentos Aleatórios**, quando ativa, deve gerar aleatoriamente uma quantidade (valor definido no **JSpinner** no momento da geração) de comandos e enviá-los para o robot. Os comandos a gerar aleatoriamente e a enviar para o robot são:

`reta(int distancia) : void` – a distância deve ter um valor aleatório entre 10cm e 50cm;

`curvarDireita(raio:int, angulo: int) : void` – o raio deve ter um valor aleatório entre 10 e 30 cm e o angulo um valor aleatório entre 20 e 90 graus.

`curvarEsquerda(raio:int, angulo: int) : void` – o raio deve ter um valor aleatório entre 10 e 30 cm e o angulo um valor aleatório entre 20 e 90 graus.

`parar(false:boolean) : void` – pára o robot depois de todos os comandos anteriores serem cumpridos.

A tarefa **Movimentos Aleatórios** depois de enviar um ou mais comandos aleatórios, deve esperar o tempo de execução dos comandos enviados.

As fórmulas para cálculo do tempo de execução de cada comando são:

$$\begin{aligned} tempo_{reta}[ms] &= \frac{distância[cm]}{velocidade \left[\frac{cm}{ms} \right]} + tempo_{comunicação}[ms] \\ tempo_{curvar}[ms] &= \frac{raio[cm] \times angulo[rad]}{velocidade \left[\frac{cm}{ms} \right]} + tempo_{comunicação}[ms] \\ tempo_{parar}[ms] &= tempo_{comunicação}[ms] \end{aligned}$$

Considere a velocidade do robot de 20cm/s e o tempo de comunicação de 100ms.

Objetivos das aulas práticas

Aula prática 1 – Instalação do *WindowBuilder* sobre o Eclipse. Desenho da GUI.

Aula prática 2 – Instalação das bibliotecas do robot. Tratamento dos eventos dos componentes da **GUI**. Criar a classe **Dados** para guardar a informação introduzida pelo utilizador. Teste da **GUI** com o robot.

Aula prática 3 – Desenho do diagrama de estados da tarefa **Movimentos Aleatórios** utilizando o *MagicDraw*. Desenho do diagrama de classes envolvendo as classes **Dados**, **GUI** e a tarefa **Movimentos Aleatórios** utilizando o *MagicDraw*.

Aula prática 4 – Implementação e teste da tarefa **Movimentos Aleatórios**.

Aula prática 5 – Implementação da sincronização entre a **GUI** e a tarefa **Movimentos aleatórios** no acesso ao robot.

Aula prática 6 – Apresentação e avaliação do trabalho prático 1.

Avaliação

O trabalho é realizado em grupo e está sujeito a avaliação.

O trabalho começa na segunda aula prática do semestre.

A duração do trabalho prático é de 6 aulas práticas.

Por cada aula prática, cada grupo deverá entregar um relatório semanal que descreve o trabalho desenvolvido na aula.

Cada um dos relatórios semanais corresponde a uma secção do relatório final do trabalho.

O relatório final deve incluir uma capa, índices, uma introdução e conclusões.

A discussão do trabalho será realizada na(s) semana(s) seguinte(s) à entrega do relatório final após a última aula prática do trabalho.

Os Docentes,
Jorge Pais, Carlos Gonçalves e Carlos Carvalho