

Projeto de Fundamentos de Programação

Manual do utilizador

GRUPO N° 9

106426	Pedro Rosa pedro.b.rosa@tecnico.ulisboa.pt
105995	João Miguel joao.tomas.miguel@tecnico.ulisboa.pt

Licenciatura em Engenharia Mecânica

Lisboa, junho de 2023

Manual do utilizador

Índice

1.	Manual do utilizador.....	3
1.1.	Descrição do programa	4
1.2.	Guia de utilização	6
1.3.	Limitações do programa.....	12



Manual do utilizador

1. Sumário

O programa *Robot Restaurant Cleaner* é um jogo desenvolvido em Python, utilizando a biblioteca *graphics* (John Zelle). Toda a interação é feita através de interface gráfica. Este jogo consiste num *robot* que limpa pontos de sujidade num restaurante e é constituído por 3 níveis/implementações: no nível 1 o utilizador seleciona onde quer que o robot limpe, no nível 2 o robot limpa as sujidades definidas pelo utilizador e o nível 3 é semelhante ao 2 sendo que os obstáculos estão distribuídos aleatoriamente.

O programa implementa:

Uma janela de menu onde os vários níveis de implementação são escolhidos pelo utilizador e executados (totalmente funcional). Em todas as janelas que o programa cria está disponível um botão para voltar para o menu.

No **nível 1**: O robot desloca-se, sem problemas, para o ponto que o utilizador deseja; o robot contorna o obstáculo em questão, nunca o trespassando. O robot apenas volta para a *docking station* (que está mais perto) assim que todos os pontos de sujidade tenham sido limpos.

No **nível 2**: O utilizador consegue escolher com liberdade os pontos de maior sujidade na sala. O robot desloca-se sem problemas, nunca colidindo com obstáculos e minimizando o tempo de limpeza. A bateria esgota-se quando o robot percorre duas vezes a área de navegação, dirigindo-se para a docking station e carregando. Possui a luz indicadora de bateria.

No **nível 3**: As opções gerar obstáculos aleatoriamente e através de coordenadas de um bloco de notas estão ambas funcionais. Quando os obstáculos são gerados aleatoriamente, nunca há sobreposição e há sempre espaço para o robot contornar os objetos. As opções de gerar sujidades através de cliques de rato também estão funcionais. O robot, em quaisquer destas opções, limpa a sala minimizando o tempo de limpeza. A bateria também funciona da mesma forma do que no nível 2.

A estimativa de tempo para a execução do projeto foi de 180h.

Manual do utilizador

2. Manual do utilizador



2.1. Descrição do programa

Menu

O programa contempla um menu que permite: aceder a implementações 1, 2 e 3; aceder a informações e sair do programa. Note-se que este menu é acessível em qualquer implementação e mesmo na janela de informações.

Barra de ferramentas

O programa contempla em todas as implementações uma barra de ferramentas que apresenta essencialmente duas utilidades. Por um lado, o utilizador pode voltar ao menu estando já a executar uma das implementações, e por outro lado, nas implementações 2 e 3 foi-nos conveniente colocar um botão de “começar” de modo ao *robot* iniciar a limpeza.

Implementação 1

Utilizámos, como requerido, uma classe obstáculo, chamada `Table`, que usa tanto a classe `Rectangle` como a classe `Circle`. Esta classe permite desenhar na janela gráfica mesas de três formatos: circulares, ovais e retangulares.

A área de navegação é retangular e estão disponíveis duas *docking stations* encostadas aos bordos.

Sendo assim, o utilizador insere as coordenadas dos pontos de maior sujidade no bloco de notas e, na interface gráfica do programa, clica onde quer que o robot se desloque. Se o utilizador clicar ao redor de um ponto de sujidade, o robot desloca-se até ele e limpa. Neste processo de limpeza existem 3 hipóteses: o *robot* tem espaço necessário e suficiente para efetuar um movimento espiral, o robot tem espaço para realizar esse movimento de espiral parcialmente (o movimento de espiral é interrompido quando o *robot* está perto de um bordo ou de um obstáculo), ou o *robot* não tem completamente espaço para efetuar o movimento de espiral (todos estes três casos funcionam). Após todas as sujidades terem sido limpas, o robot retorna à docking station que estiver mais próxima.

A área de navegação, o robot e os obstáculos têm tamanhos representativos do cenário proposto.

Manual do utilizador

Implementação 2

Nesta implementação, a sala tem o número de obstáculos (Mesas, cadeiras, pianos e vasos) definidos pelo utilizador, que insere as coordenadas no bloco de notas conforme explicado abaixo. O *robot* limpa a totalidade da sala e minimiza o tempo de limpeza – vai de um extremo ao outro da sala (na vertical), sofrendo um pequeno incremento na horizontal quando completa cada trajetória vertical. O *robot* contorna qualquer obstáculo, independentemente das suas posições relativas. O utilizador escolhe o número de sujidades pretendido através de cliques de rato. O *robot* apresenta uma bateria que tem necessariamente de ser carregada após o *robot* percorrer duas vezes a área de navegação. Essa bateria apresenta quatro cores representativas do estado de bateria: verde, amarelo, laranja e vermelho. O robot muda de cor enquanto está a carregar (luz azul). Adicionalmente, acrescentámos luzes indicadoras às *docking stations*: verde quando o *robot* está a carregar ou posicionado na *dock* e vermelho caso o *robot* simplesmente não esteja na *dock*.

Implementação 3

a configuração da sala e obstáculos na sala pode ser feita de duas formas: lida de um ficheiro (sala.txt); gerada aleatoriamente pelo programa (distribuição esparsa dos objetos). A escolha de uma ou outra forma é feita através de uma janela de menu. Os pontos de maior sujidade são introduzidos de duas formas nesta implementação: dados por um ficheiro ou dados através de cliques do rato – esta escolha é feita através de uma nova janela de menu, tal como pedido. No caso de leitura de dados de um ficheiro, as entradas que definem a localização dos móveis na sala estão em termos percentuais relativos às dimensões da janela (Note bem: como foi pedido, estas dimensões são percentuais relativas à janela, ou seja, a contar com o espaço da barra de ferramentas não ocupada pelo *robot* – ler generalidades). Neste caso também é lido do bloco de notas as dimensões da janela. No caso da distribuição esparsa de objetos, decidimos deixar não restringir o utilizador a escolher os tipos de obstáculo que tenciona. É então possível escolher (em 5 obstáculos) o tipo número de cada tipo de obstáculos (mesas, cadeiras, vasos e pianos) numa nova janela de menu. Após estarem definidos os tipos de obstáculos, o programa distribui os obstáculos aleatoriamente garantindo que não há sobreposição entre obstáculos e garantindo que estes não se encostam aos bordos.

Manual do utilizador

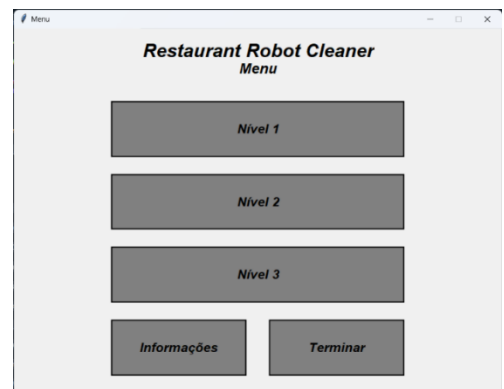
2.2. Guia de utilização

Generalizações

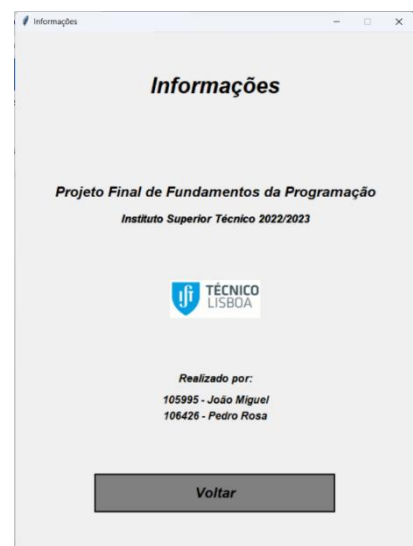
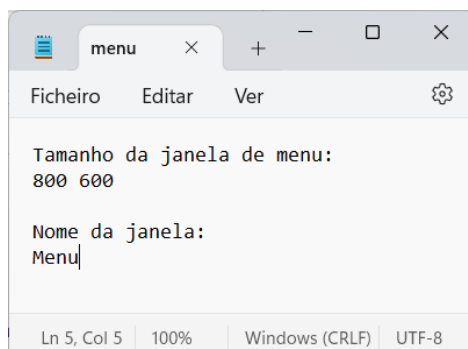
Devido à imposição que nos foi dada de utilizar a instrução `win.setCoords(0.0,0.0,100.0,100.0)` de modo às coordenadas dos objetos ficarem em percentagem em relação à janela gráfica, e devido a termos em todo o programa uma barra de ferramentas naturalmente não acessível pelo robô, existe um cuidado a ter. A área acessível ao robot é aquela em que, considerando um referencial xy , o x varia entre 0 e 100 e o y varia entre 0 e 86. É muito importante ter isto em consideração, embora o nosso programa avise, por exemplo, quando é colocado um obstáculo ou uma sujidade na barra de tarefas.

Menu

Ao iniciar o programa, é criada uma janela de menu onde podemos seleccionar os níveis de execução através de botões de cor cinzenta. O menu contém também um botão para terminar o programa e outro botão para a página de informações. Todas as páginas possuem um botão para voltar ao menu, de modo a que o utilizador possa navegar pelas várias páginas sem ter que sair do programa e o volte a correr.



Para o bom funcionamento da página de informações, não é necessário nenhum *input* por parte do utilizador. No entanto, para o menu, será necessária uma nota com o nome “menu.txt” apresentada da seguinte forma:



Manual do utilizador

Nível 1

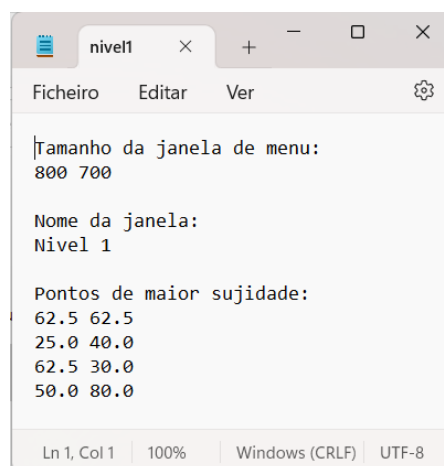
Ao carregar no botão “Nível 1” no menu, o programa corre a primeira implementação: é criada uma janela gráfica onde aparece uma barra de navegação e uma sala de um restaurante com um *robot*, duas *docking stations*, uma mesa e um número de sujidades escolhido pelo utilizador.

Interpretámos (e confirmámos com o docente responsável) que nesta implementação o utilizador deve clicar no local onde a sujidade deverá ser eliminada. Sendo assim, ficou ao nosso critério escolher o método de adquirir as coordenadas dos pontos de sujidade: decidimos que estas sejam colocadas no documento de texto `nivel1.txt`.

Para o bom funcionamento do nível 1 será necessária uma nota com o nome “`nivel1.txt`” apresentada da forma apresentada à direita. Note bem a forma de indicar as coordenadas:

Coordenada x + espaço + coordenada y

É permitido inserir tantas coordenadas de sujidade quanto o utilizador desejar.



Neste nível o utilizador poderá carregar em qualquer zona da página para que o robot se desloque (no caso dessa zona ser a barra de navegação ou o obstáculo o robot não se desloca). O robot irá andar com os cliques do utilizador até que todas as sujidades tenham sido limpas. Nessa altura, o robot voltará à *docking station*. O programa termina aparecendo a mensagem:

Limpeza concluída!
Carregue no botão "sair" para voltar para o menu

Caso uma ou mais sujidades esteja nas na área da barra de navegação, a interseção algum obstáculo ou posicionada fora da janela, o programa irá mostrar a mensagem à direita.

Erro nas coordenadas!
Coordenadas fora da janela ou a interseção um obstáculo
Clique em qualquer lugar para voltar para o menu

Tal como diz na mensagem, basta clicar em qualquer sítio da janela para voltar para o menu.

Manual do utilizador

Nível 2

Ao carregar no botão “Nível 2” no menu, aparece a página da segunda implementação: gera-se uma janela gráfica com barra de navegação e uma sala de um restaurante com um *robot*, duas *docking stations*, e um número de obstáculos escolhido pelo utilizador.

Neste nível o utilizador pode carregar em qualquer zona da página para que apareça uma sujidade, não incluindo a barra de navegação, as *docking stations* ou os obstáculos na sala (nada acontece quando aqui se clica). De seguida, o utilizador deve carregar no botão “Começar” para que o *robot* inicie o seu percurso de limpeza.

Quando este acaba as duas voltas à sala e de recarregar o programa repete-se deixando o utilizador inserir mais coordenadas (para, depois, limpar novamente a sala) ou que volte para o menu.

Para o bom funcionamento do nível 2 será necessária uma nota com o nome “nivel2.txt” apresentada da forma representada à direita.

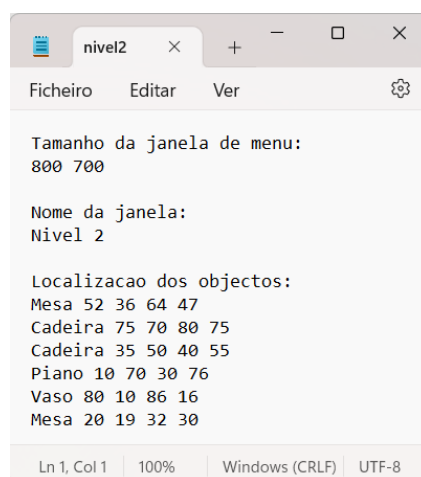
Note bem: É dada a opção ao utilizador de escolher a localização dos obstáculos (embora não seja um requerimento). Para isso, o utilizador deve ter em conta as dimensões de cada obstáculo ($x \times y$):

Mesa – 12x11; **Cadeira** – 5x5; **Piano** – 20x6; **Vaso** – 6x6

O formato a usar no bloco de notas é o seguinte (/ = espaço):

Nome do obstáculo / x1 / y1 / x2 / y2

Com x1, y1 e x2, y2 extremos/vértices de cada obstáculo.



```
Tamanho da janela de menu:
800 700

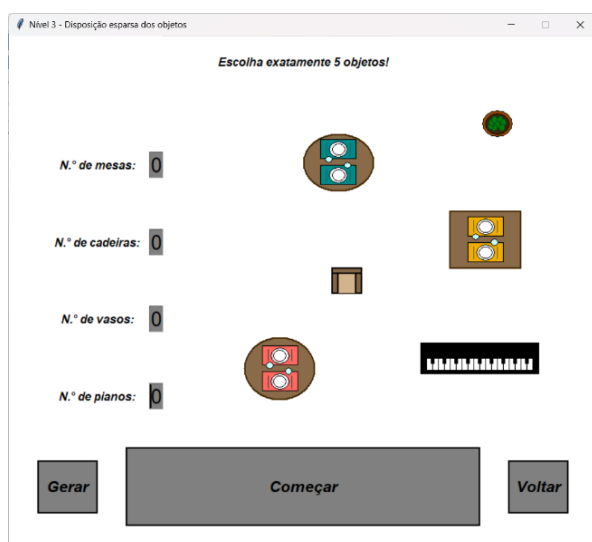
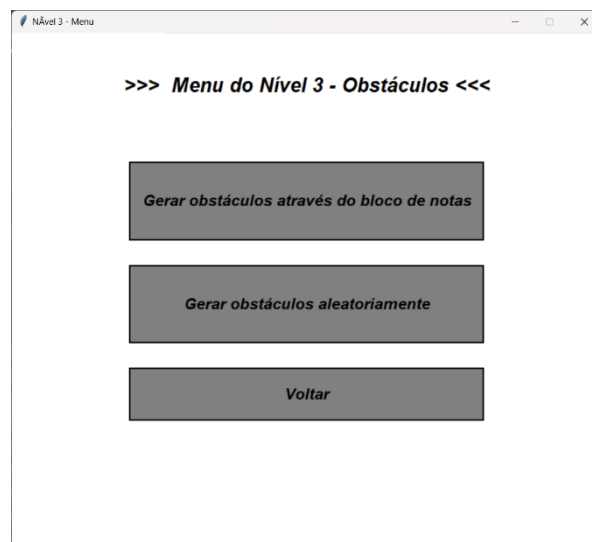
Nome da janela:
Nivel 2

Localizacao dos objectos:
Mesa 52 36 64 47
Cadeira 75 70 80 75
Cadeira 35 50 40 55
Piano 10 70 30 76
Vaso 80 10 86 16
Mesa 20 19 32 30
```


Manual do utilizador

Nível 3:

Ao carregar no botão “Nível 3” no menu, aparece uma nova janela de menu onde o utilizador escolhe se quer que os obstáculos sejam dados por um documento txt ou gerados aleatoriamente. Adicionalmente, existe o botão de Voltar:

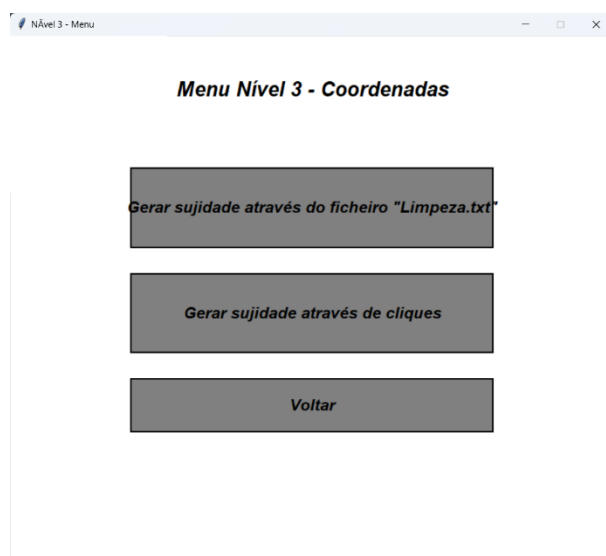


Caso o utilizador escolha o modo aleatório será criada outra janela onde o utilizador poderá escolher a o número de cada tipo de obstáculos, sendo que o número de obstáculos tem que ser necessariamente igual a 5. O botão “Gerar” faz com que o programa gere aleatoriamente o número de cada tipo de obstáculos. De seguida o utilizador deve carregar no botão “Começar”.

Caso escolha para gerar os obstáculos através do bloco de notas, as coordenadas devem estar em

conformidade com o formato apresentado mais abaixo.

Depois de escolher o método de gerar obstáculos, aparecerá a janela onde o utilizador terá que escolher se quer que as coordenadas das sujidades sejam dadas pela nota “Limpeza.txt” ou se quer que as sujidades apareçam através do clique do rato, tal como no nível 2.

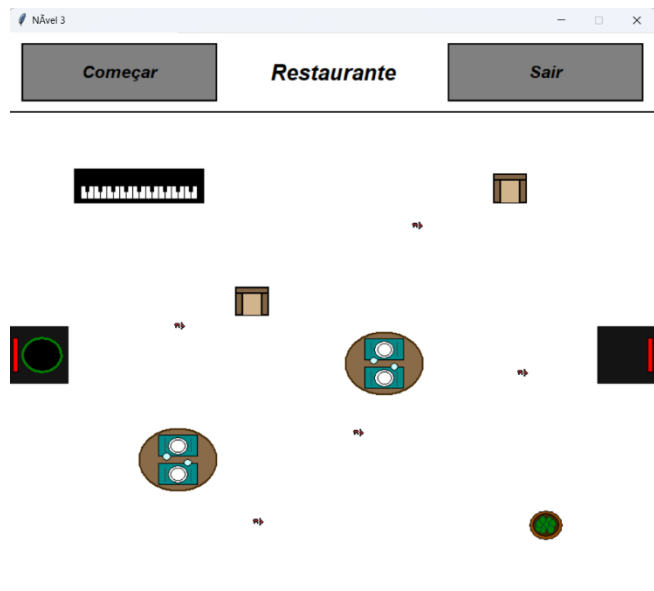


Manual do utilizador

10

Caso escolha a primeira opção, as sujidades aparecerão assim que o utilizador entre na página com o restaurante, conformes as coordenadas estejam no documento txt (explicado abaixo). Se uma ou mais sujidades esteja nas coordenadas da barra de navegação, algum obstáculo ou fora da janela o programa irá mostrar a mesma mensagem apresentada no nível 1. De seguida, o utilizador deve carregar no botão “Começar” para que o *robot* inicie o seu percurso de limpeza.

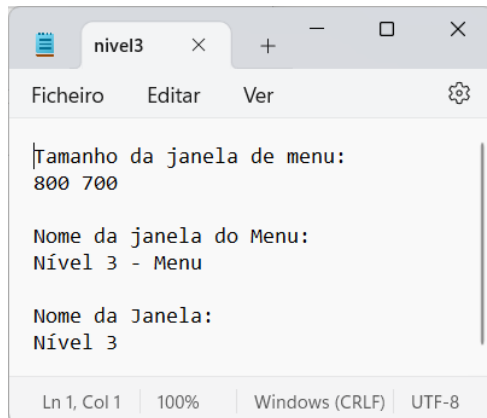
Caso escolha a segunda opção o programa irá decorrer da mesma forma que o nível 2, apenas aparecendo a sujidade com o clique do utilizador (se este não for em cima da barra de navegação, das *docking stations* ou dos obstáculos na sala). De seguida, o utilizador deve carregar no botão “Começar” para que o *robot* inicie o seu percurso de limpeza.



Quando este acaba as duas voltas à sala e de carregar, o programa estará pronto para receber mais pontos de sujidade através do clique do utilizador independentemente da opção escolhida para a localização inicial das sujidades. Pode-se também voltar para o menu.

Manual do utilizador

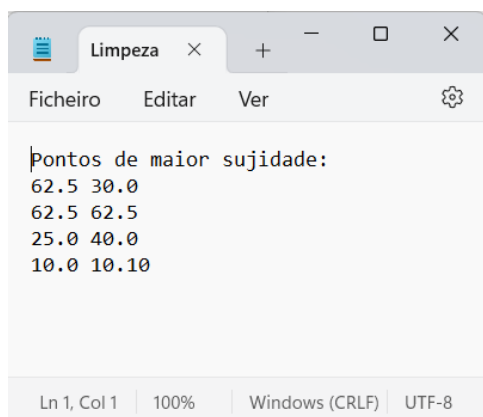
Para o bom funcionamento do nível 3 serão necessárias três notas com o nome “nivel3.txt”, “Limpeza.txt” e “Sala.txt” apresentadas da seguinte forma:



A screenshot of a text editor window titled "nivel3". The window has a menu bar with "Ficheiro", "Editar", and "Ver". The text inside the editor is as follows:

```
Tamanho da janela de menu:  
800 700  
  
Nome da janela do Menu:  
Nível 3 - Menu  
  
Nome da Janela:  
Nível 3
```

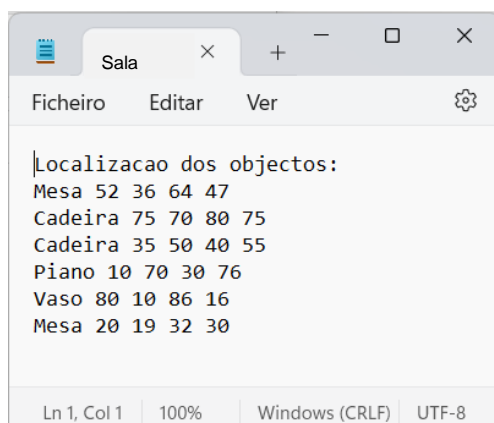
The status bar at the bottom shows "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8".



A screenshot of a text editor window titled "Limpeza". The window has a menu bar with "Ficheiro", "Editar", and "Ver". The text inside the editor is as follows:

```
Pontos de maior sujidade:  
62.5 30.0  
62.5 62.5  
25.0 40.0  
10.0 10.10
```

The status bar at the bottom shows "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8".



A screenshot of a text editor window titled "Sala". The window has a menu bar with "Ficheiro", "Editar", and "Ver". The text inside the editor is as follows:

```
Localizacao dos objectos:  
Mesa 52 36 64 47  
Cadeira 75 70 80 75  
Cadeira 35 50 40 55  
Piano 10 70 30 76  
Vaso 80 10 86 16  
Mesa 20 19 32 30
```

The status bar at the bottom shows "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8".

Manual do utilizador

2.3. Limitações do programa



Durante o funcionamento de uma função não é possível paralelamente executar outra função. Isto implica que enquanto o robot se move não é possível carregar em nenhum botão. Por exemplo, se o utilizador quiser sair do programa terá de esperar que o robot acabe de limpar.