

# Trabalho Final

Aluno: Victor Dias Frota 16-99988

Este programa consiste em 4 partes: `main`, `robo_limpeza`, `robo` e `estado`.

## Main

O script `main` simplesmente executa o programa e define os tamanhos do espaço de limpeza do robô, a quantidade de obstáculos e sujeiras.

## RoboLimpeza

É a classe que define as regras do exercício, por assim dizer.

### Atributos:

- **tamanho:** define a dimensão do campo em que o robô pode andar.
- **passos:** registra o número de movimentos do robô.
- **robo:** classe que controla o robô.
- **matriz, x e y:** representam o espaço onde tudo acontece.

### `criar_matriz()`

Recebe o tamanho, número de obstáculos e sujeiras e cria uma matriz com os valores especificados, além de um robô posicionado aleatoriamente.

### `exibir_matriz()`

Originalmente exibia a matriz com os valores de cada elemento:

- Espaço vazio: `0`
- Sujeira: `1`
- Obstáculo: `1`
- Robô: `x`

Porém, durante o desenvolvimento do programa, percebi que seria melhor trocar esses elementos por caracteres ASCII para melhor visualização:

- Espaço vazio (`0`) ⇒ fundo escuro
- Sujeira (`1`) ⇒ círculo verde ●
- Obstáculo (`1`) ⇒ fundo vermelho
- Robô (`x`) ⇒ losango azul ◆

### `mover_robo()`

Recebe uma direção no formato WASD e a converte em coordenadas na matriz. Exemplo:

'W': (-1, 0), 'S': (1, 0), 'A': (0, -1), 'D': (0, 1)

O método verifica se o movimento do robô colidirá com sujeira, obstáculo ou borda do mapa.

- Se colidir com uma sujeira ( `1` ), adiciona ao contador e remove da matriz.
- A função retorna `True` se todas as sujeiras forem recolhidas.

### `iniciar_limpeza()`

Cria um loop que:

1. Executa o próximo passo do robô.
2. Exibe a matriz.
3. Incrementa o contador de passos.
4. Encerra o programa se a tecla de saída ( `q` ) for pressionada ou se todas as sujeiras forem limpas.

Além disso, adicionei `os.system('cls')` para limpar a tela do terminal a cada iteração, melhorando a visualização.

## Robo

É a classe que determina os movimentos do robô.

### `run()`

Define para qual direção o robô irá andar ( `'W'`, `'A'`, `'S'`, `'D'` ), baseado principalmente no método `encontrar_caminho_para_qualquer_sujeira()` da classe `Estado`.

Além disso, exibe o total de sujeiras e limpezas realizadas.

## Estado

É a classe que decide a estratégia de movimentação do robô.

### `get_num_sujeiras()` e `get_sujeiras_limpas()`

Retornam os valores dos atributos para serem utilizados.

### `set_sujeiras_limpas()`

Atualiza o valor das sujeiras limpas.

### `calcular_distancia()`

Calcula a distância entre duas células da matriz usando a função `abs()`. Como o robô não pode se mover na diagonal, a métrica usada é a **distância de Manhattan**, que se assemelha à forma de andar por quarteirões de uma cidade.

### `listar_sujeiras()`

Cria uma lista com as coordenadas de todas as sujeiras no mapa, analisando cada elemento da matriz.

### `encontrar_caminho_para_sujeira()`

Recebe a posição do robô, uma posição de destino e a matriz do mapa.

1. Cria uma fila para busca, contendo a posição inicial do robô e uma lista de caminhos.
2. Mantém uma lista de posições já visitadas.

3. Inicia um loop onde:

- Pega o primeiro elemento da fila.
- Se a posição atual for a do destino, retorna o caminho escolhido.
- Se a posição já estiver na lista de caminhos visitados, ignora o restante e continua para a próxima iteração.
- Caso contrário, simula todas as direções possíveis e adiciona novos caminhos à fila de busca, se não houver obstáculos ou saída dos limites do mapa.

O loop termina quando não houver mais caminhos disponíveis para alcançar a sujeira desejada.

#### `encontrar_caminho_para_qualquer_sujeira()`

É uma extensão do método anterior, analisando caminhos para múltiplas sujeiras.

1. Obtém a lista de sujeiras via `listar_sujeiras()`.
2. Se não houver sujeiras, retorna imediatamente.
3. Ordena a lista de sujeiras pela distância até o robô.
4. Percorre a lista de sujeiras da mais próxima para a mais distante:
  - Se a sujeira estiver na lista de inacessíveis, pula para a próxima.
  - Chama `encontrar_caminho_para_sujeira()` e armazena o caminho retornado.
  - Se nenhum caminho for encontrado, adiciona a sujeira à lista de inacessíveis.

Se o loop se encerrar sem encontrar caminhos acessíveis, a função retorna `None`, indicando que todas as sujeiras foram limpas ou que não há sujeiras acessíveis restantes.