# ACESSE O GITHUB PARA AS INFORMAÇÕES TOTAIS DE TODO O TRABALHO DESENVOLVIDO:

**CLIQUE NO LINK:**

👇👇👇👇👇👇👇👇👇👇👇

https://github.com/Josiel-Pantaleao/TELMA_heuristic

# Relatório Comparativo: Agente Reativo vs. Agente com Objetivo (BFS) em Ambiente NetLogo

Agente: Agente reativo Implemente no ambiente desenvolvido um agente reativo simples que faça a coleta de recursos até o limite da sua capacidade. O agente deve se mover de forma aleatória pelo ambiente e ser capaz de detectar que existe um recurso nas células vizinhas. Ao atingir o limite da sua capacidade o agente deve retornar para a posição inicial do mapa e mudar a sua cor para vermelho.

## 1. Introdução

Este projeto teve como objetivo demonstrar, por meio de uma simulação no NetLogo (integrada com Python via extensão), as diferenças de desempenho entre dois tipos de agentes para coleta de recursos:

- **Agente Reativo Simples:**
  Move-se de forma aleatória pelo ambiente, coletando recursos conforme os encontra. Ao atingir sua capacidade máxima, retorna à posição inicial (0,0) de maneira também aleatória e muda de cor para vermelho.

- **Agente com Objetivo (utilizando BFS):**
  Utiliza a busca em largura (BFS) para localizar o recurso mais próximo e seguir o caminho mais curto até ele. Dessa forma, o agente coleta recursos de forma planejada. Quando a capacidade máxima é atingida, o agente utiliza novamente o BFS para retornar à posição inicial, o que torna o percurso de retorno muito mais eficiente.

O ambiente simulado é uma grade (20×20 patches) onde cada patch tem 20% de chance de conter um recurso (representado visualmente pela cor verde). Ao coletar o recurso, o patch muda para cinza.

## 2. Metodologia

### Ambiente e Agente

- **Ambiente:**
  Cada patch é configurado para ter um recurso com probabilidade de 20%.

  - Se o patch contém recurso, ele é pintado de **verde**; caso contrário, é **cinza**.
- **Agente:**
  Uma única pessoa é criada para representar o agente. Sua cor inicial é azul e seu shape é "person".

  - **Capacidade:** O agente tem uma capacidade fixa (definida como 5 no código).
  - **Comportamento:**
    - No **modo reativo**, o agente move-se aleatoriamente e coleta recursos das células vizinhas sem planejamento.
    - No **modo com objetivo (BFS)**, o agente calcula o caminho mais curto até o recurso mais próximo e o segue, otimizando assim o tempo de coleta.
  - **Retorno:** Quando o agente atinge sua capacidade máxima, ele retorna à posição inicial. Se já estiver na posição inicial (0,0) com a capacidade cheia, ele muda de cor para vermelho e a simulação é encerrada.

### Implementação

A lógica do agente foi implementada utilizando a extensão Python para NetLogo.

- O código Python contém classes para o ambiente e o agente, funções para verificação e coleta de recursos, além do algoritmo BFS para planejamento de caminho.
- A integração é feita chamando a função `step_agent` a cada tick, que atualiza a posição do agente e coleta recursos conforme necessário.
- Testei os dois modos de operação:
  - **Modo Reativo (random):** O agente se move aleatoriamente.
  - **Modo com Objetivo (bfs):** O agente utiliza BFS para encontrar o caminho mais curto até o recurso.

# 3. Resultados

Em meus testes, observei diferenças significativas no tempo necessário para o agente completar sua missão (coletar até atingir a capacidade e retornar à base):

- **Agente Reativo Simples:**
  Em um teste, o agente levou cerca de **85 ticks** para atingir sua capacidade e retornar à posição inicial.

- A trajetória foi aleatória, resultando em movimentos desnecessários e caminhos mais longos.

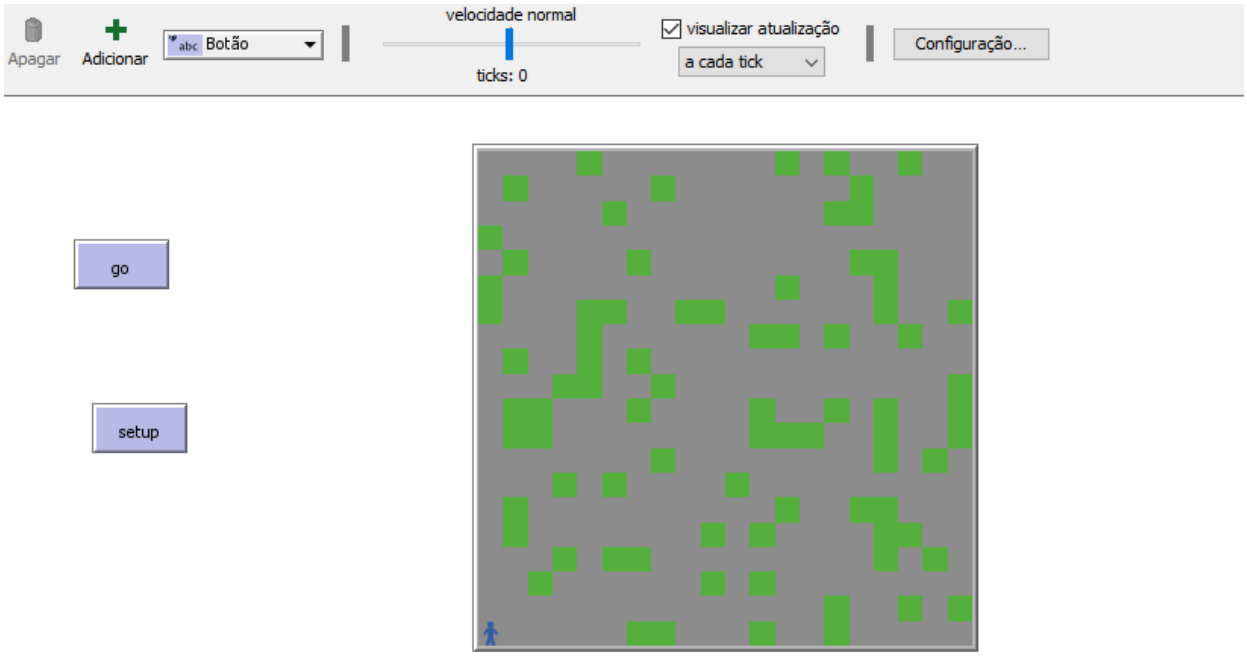
- **Agente com Objetivo (BFS):**
  Em outro teste, o agente completou a tarefa em cerca de **17 ticks**.

  - Ao utilizar o BFS para planejar o caminho, o agente encontrou o recurso mais próximo e seguiu a rota mais curta, otimizando seu tempo.

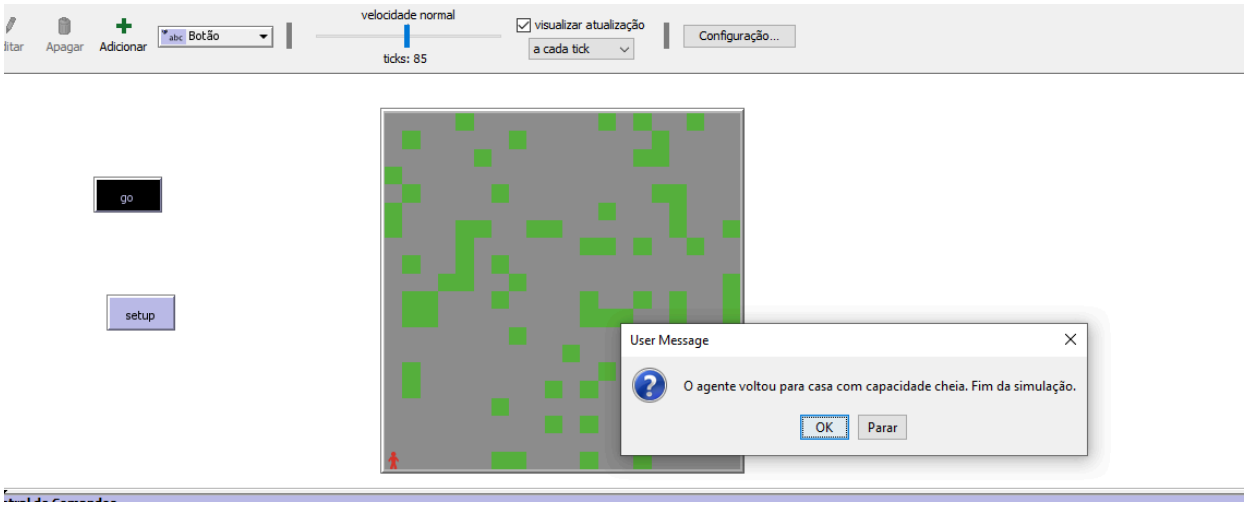
## Agente reativo:

Implemente no ambiente desenvolvido um agente reativo simples que faça a coleta de recursos até o limite da sua capacidade. O agente deve se mover de forma aleatória pelo ambiente e ser capaz de detectar que existe um recurso nas células vizinhas. Ao atingir o limite da sua capacidade o agente deve retornar para a posição inicial do mapa e mudar a sua cor para vermelho.

## Como começou:



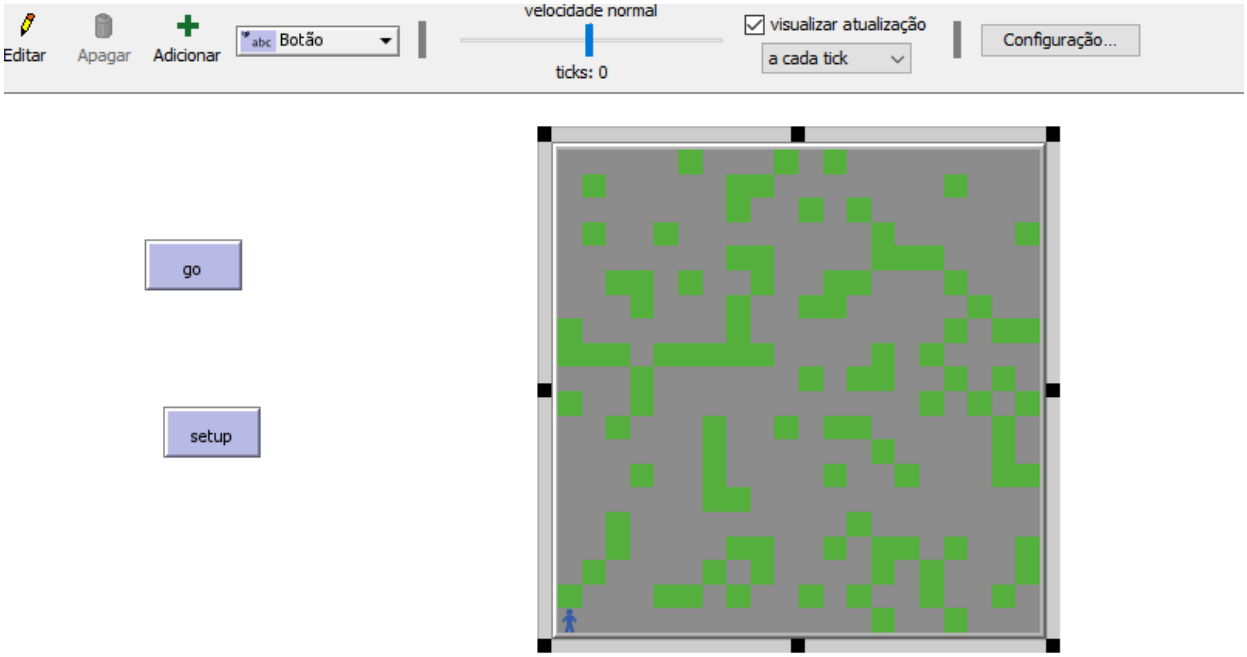
## Quantos passos:



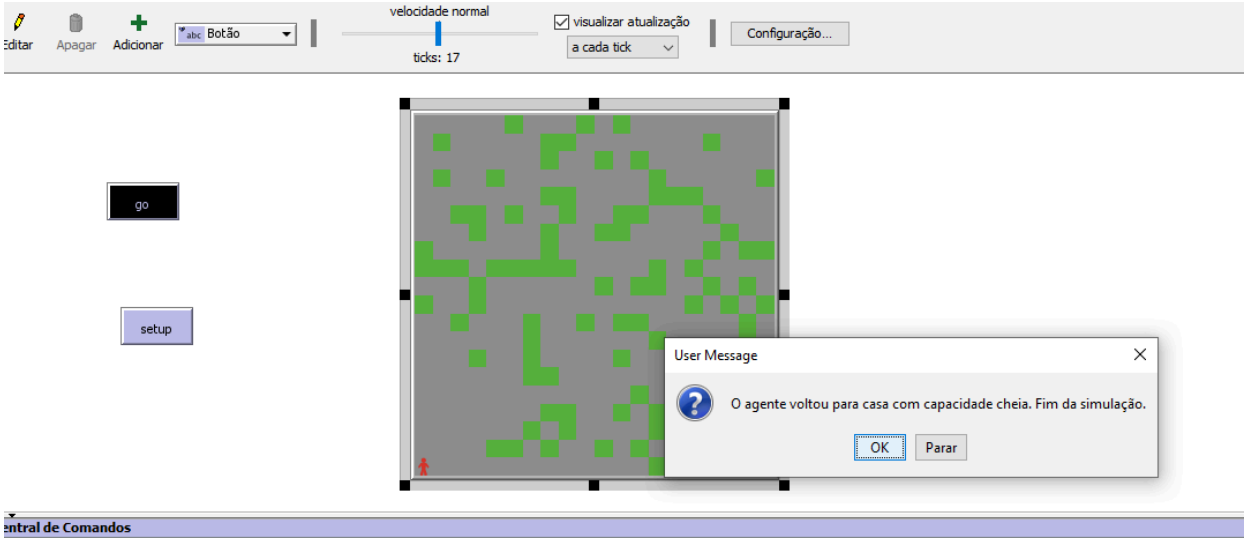

## Agente com Objetivo:

Nesta parte da atividade modifique o modelo criado anteriormente para que agora o agente possua um objetivo. O objetivo do agente passa a ser coletar o limite da sua capacidade de recursos no menor tempo possível. Para executar o agente deve fazer uso do método de busca em largura (DFS) para encontrar o menor caminho entre

## Como começou:




## Quantos passos:



# 4. Discussão

## Vantagens e Desvantagens

### Agente Reativo Simples

- **Vantagens:**
  - **Simplicidade:** Fácil de implementar, pois não requer algoritmos de planejamento.
  - **Robustez em Ambientes Dinâmicos:** Pode ser vantajoso em cenários onde o ambiente muda muito rapidamente e o planejamento prévio se torna obsoleto.
- **Desvantagens:**
  - **Ineficiência:** Os movimentos aleatórios podem resultar em trajetórias longas e redundantes, aumentando o tempo para coletar os recursos.
  - **Variabilidade:** O desempenho pode variar muito de uma execução para outra.

### Agente com Objetivo (BFS)

- **Vantagens:**
  - **Eficiência:** Ao usar BFS, o agente encontra o caminho mais curto até os recursos, otimizando o tempo e reduzindo o consumo de energia.
  - **Previsibilidade:** A abordagem planejada tende a produzir trajetórias mais consistentes e confiáveis.
- **Desvantagens:**
  - **Dependência do Modelo:** Requer que o ambiente esteja bem modelado; alterações frequentes podem invalidar o planejamento.
  - **Custo Computacional:** Em ambientes muito grandes ou complexos, o algoritmo BFS pode se tornar mais custoso em termos de processamento.

Segue abaixo um exemplo de texto revisado com exemplos práticos para cada abordagem (agente reativo simples e agente com objetivo utilizando BFS):

---

# Aplicações no Mundo Real

Os conceitos e técnicas demonstrados podem ser aplicados em diversas áreas, tais como:

## Robô que limpa a acasa

- **Agente Reativo Simples:**
  Eu tenho um robô em casa (KABUM SMART100) que faz aspiração aleatória no

ambiente. Ele vai andando aleatóriamente pelo ambiente e aspirando conforme vai encontrando sujeira, até se encher e voltar a base de carregamento de forma automática. Essa abordagem é vantajosa em ambientes altamente dinâmicos (por exemplo, com pessoas se movendo) onde um planejamento complexo pode se tornar obsoleto rapidamente.

- **Agente com Objetivo (BFS):**
  Agora o robô XIAOMI X20 MAX. Ele anda diretamente até a sujeita e faz a identificação total do ambiente e ele realmente utiliza a busca em largura (BFS), o robô calcula o caminho mais curto até a sujeira, economizando energia e tempo e após isso volta a base de carregamento. Essa abordagem é muito eficiente em ambientes estáveis, onde os obstáculos e a disposição dos itens são conhecidos.

### Logística

- **Agente Reativo Simples:**
  Em um cenário onde veículos autônomos precisam operar em ruas muito congestionadas ou com trânsito imprevisível, um sistema reativo pode ajudar o veículo a responder rapidamente a mudanças repentinas, como freios bruscos de outros carros ou obstáculos inesperados, mesmo que não seja a rota mais curta.

- **Agente com Objetivo (BFS):**
  Em sistemas de roteirização de entregas, utilizar algoritmos de busca pode ajudar a definir a rota mais eficiente entre vários pontos de entrega. Por exemplo, um sistema que calcula a rota ideal para um caminhão de entrega, minimizando o tempo total de deslocamento entre os pontos fixos, aumenta a eficiência e reduz custos.

---

# Conclusão

A comparação entre os dois tipos de agentes demonstra que:

- **Eficiência:**
  O agente com objetivo (BFS) mostrou uma performance superior ao planejar trajetórias curtas e eficientes, reduzindo significativamente o tempo de execução (por exemplo, de 85 ticks para 17 ticks em nossos testes).

- **Aplicabilidade:**

    - O **agente reativo simples** é adequado para ambientes altamente dinâmicos, onde o planejamento pode ser prejudicado por mudanças constantes.

    - O **agente com objetivo (BFS)** é ideal para ambientes mais estáveis, onde um planejamento prévio pode otimizar significativamente os recursos e o tempo.

Em aplicações do mundo real, a escolha entre uma abordagem reativa e uma baseada em planejamento depende das características do ambiente e dos requisitos da tarefa. No meu caso seria muito melhor o da XIAOMI, mas a integração de algoritmos de busca é fundamental. Em contrapartida, em ambientes imprevisíveis, uma abordagem reativa pode oferecer maior flexibilidade.

---

# Referências


1. NetLogo Official Website
2. Python Extension for NetLogo GitHub
3. Russell, S. & Norvig, P. (2013). *Inteligência Artificial* (3ª ed.). Elsevier Editora.
4. Tutoriais e vídeos sobre NetLogo e Agentes Inteligentes (veja links no README original).