



UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIAS
ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA: INTELIGÊNCIA ARTIFICIAL

Docente: Dr. Thales Levi Azevedo

AGENTES - MODIFICANDO O MODELO DAS FORMIGAS PARA UM AMBIENTE
COZINHA

Euderlan Freire da Silva Abreu - 2021052460
Hissa Bárbara Oliveira - 2021052405
Lucas Silva Costa - 2022019085

São Luís-MA
22/05/2025

Euderlan Freire da Silva Abreu - 2021052460

Hissa Bárbara Oliveira - 2021052405

Lucas Silva Costa - 2022019085

AGENTES - MODIFICANDO O MODELO DAS FORMIGAS PARA UM AMBIENTE
COZINHA

Este trabalho consiste na criação de um formigueiro numa cozinha, com obstáculos e predadores, e outras melhorias do modelo base. Este documento é referente a disciplina de Inteligência Artificial do Curso de Engenharia da Computação, ministrado pelo Prof. Dr. Thales Levi Azevedo

Sumário

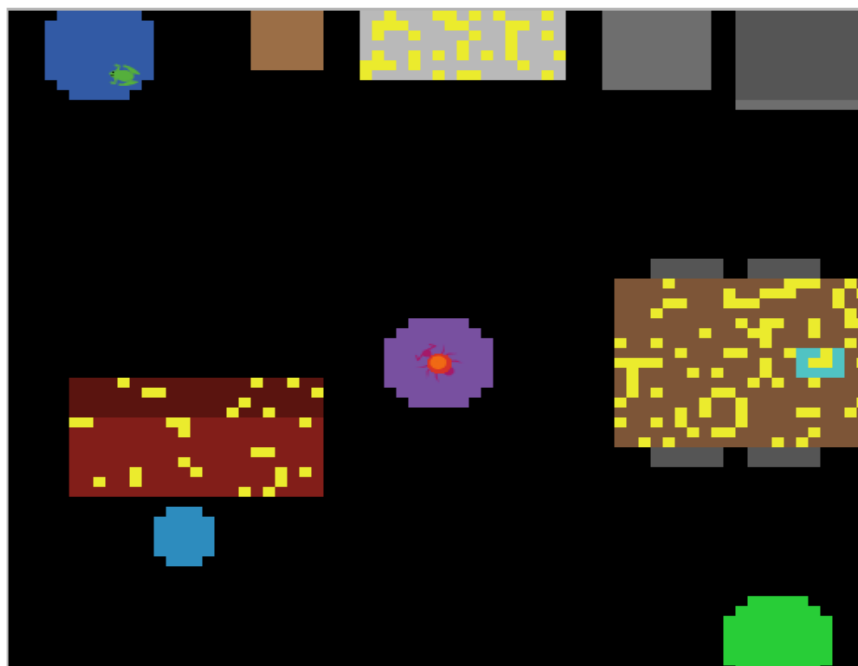
1. Introdução	5
1.2 Conceitos Teóricos Envolvidos	6
2. Objetivo	6
2.1 Objetivo Geral	6
2.2 Objetivos Específicos	6
3. Desenvolvimento	7
3.1 Arquitetura Multi-Agente	7
3.2 Sistema de Ambiente Dinâmico	
3.2.1 Modelagem da Cozinha	10
3.4 Sistema de Energia e Sobrevivência	12
3.5 Dinâmica Populacional	13
3.5.1 Reprodução Inteligente	13
3.5.2 Modo de Escassez	14
3.6 Sistema de Perigos	15
3.6.1 Sistema de Predação	15
3.6.1.1 Definição e Características	16
3.6.1.2 Criação no Setup	16
3.6.1.3 Comportamento de Caça	16
3.6.1.3 Interações e Efeitos no Sistema	17
3.6.1.5 Parâmetros de Controle	17
3.6.2 Detecção de Fogo	18
3.7 Algoritmos de Navegação	
3.7.1 Detecção de Obstáculos	19
3.8.1 Sistema de Aparição Probabilística	21
3.8.3 comportamento de Movimento Inteligente	23
4. Resultados	23
5. Discussão	24
6. Conclusão	25
7. Referências Bibliográficas	26

1. Introdução

Este projeto teve como objetivo principal expandir significativamente o modelo básico de formigas do NetLogo, transformando-o em um ecossistema complexo e dinâmico que simula um ambiente de cozinha com múltiplos tipos de agentes interagindo de forma realista. A proposta central foi criar um sistema que demonstrasse conceitos avançados de Inteligência Artificial através da implementação de comportamentos emergentes, auto-organização, adaptação ambiental e hierarquia social entre os agentes.

As implementações no projeto visam não apenas adicionar complexidade ao modelo, mas principalmente explorar como regras simples aplicadas a agentes individuais podem resultar em comportamentos coletivos sofisticados e inteligentes. O ambiente escolhido - uma cozinha doméstica (ver figura 1) oferece um contexto familiar e intuitivo para observar essas dinâmicas, permitindo uma melhor compreensão dos conceitos teóricos através de analogias com situações do mundo real.

Figura 1 : Ambiente de cozinha doméstica



Fonte: NetLogo

1.2 Conceitos Teóricos Envolvidos

O desenvolvimento deste projeto fundamenta-se em diversos conceitos centrais da Inteligência Artificial e sistemas multi-agentes. O comportamento

emergente representa um dos pilares teóricos mais importantes, demonstrando como padrões complexos e sofisticados podem surgir naturalmente das interações simples entre agentes individuais, sem necessidade de coordenação centralizada ou programação explícita desses padrões.

A auto-organização constitui conceito fundamental explorado no projeto, manifestando-se e otimizando-se de forma autônoma, com os agentes desenvolvendo divisões de trabalho, rotas eficientes e estratégias de sobrevivência sem direcionamento externo. Esse fenômeno é observável tanto na formação espontânea de trilhas químicas quanto na especialização funcional das diferentes castas de formigas.

A adaptação representa um terceiro pilar conceitual, implementada através de mecanismos que permitem aos agentes responder dinamicamente às mudanças ambientais, como escassez de recursos, presença de predadores ou alterações nas condições climáticas. Esta capacidade adaptativa é crucial para a sobrevivência e prosperidade da colônia em um ambiente dinâmico e desafiador.

Por fim, a hierarquia social e especialização funcional demonstram como sistemas complexos podem beneficiar-se da diversidade de papéis e capacidades, com cada tipo de agente contribuindo de forma única para o sucesso coletivo, criando sinergias que não existiriam em um sistema homogêneo.

2. Objetivo

2.1 Objetivo Geral

- O presente trabalho tem como objetivo implementar e analisar modificações no modelo de simulação de formigas em NetLogo, explorando conceitos fundamentais de Inteligência Artificial através do desenvolvimento de um sistema multi-agentes complexo que demonstra comportamentos emergentes, auto-organização e adaptação dinâmica.

2.2 Objetivos Específicos

- Implementar hierarquia social através da criação de diferentes tipos de formigas (rainha, operárias e formigas de solo) com comportamentos, capacidades e características específicas;

- Desenvolver sistema de energia e sobrevivência que simule necessidades biológicas realísticas, incluindo consumo energético, recuperação através da alimentação e mecanismos de inanição;
- Introduzir predadores inteligentes (sapos-cururus) que demonstrem comportamentos de caça adaptativos, respeitando territórios e implementando estratégias de sobrevivência;
- Criar ambiente complexo representando uma cozinha com obstáculos, móveis e diferentes alturas, testando a capacidade de navegação e especialização dos agentes;
- Implementar agentes humanos dinâmicos que interagem com o ambiente deixando recursos alimentares de forma estocástica;
- Desenvolver sistema de perigos ambientais através da simulação de fogo que afeta diretamente a sobrevivência dos agentes;
- Diferenciar tipos de alimento com valores energéticos distintos, criando hierarquia de preferências e comportamentos de forrageamento otimizados;
- Estabelecer modo de escassez adaptativo que modifica comportamentos individuais e coletivos conforme disponibilidade de recursos no ambiente;
- Analisar comportamentos emergentes resultantes das interações entre agentes simples e regras locais, demonstrando princípios de auto-organização e inteligência coletiva;
- Validar conceitos teóricos de IA através da implementação prática de agentes reativos, comunicação por feromônios e sistemas distribuídos sem controle centralizado.

3. Desenvolvimento

3.1 Arquitetura Multi-Agente

O projeto foi desenvolvido com base em uma arquitetura multi-agente descentralizada, onde diferentes tipos de agentes autônomos interagem entre si e com o ambiente. Cada agente possui comportamentos próprios definidos por regras simples, mas que, em conjunto, produzem dinâmicas complexas e emergentes. Foram implementadas raças distintas (rainha, operárias, formigas de solo, sapos

predadores e humanos), cada uma com funções específicas dentro do sistema. As decisões dos agentes são baseadas apenas em informações locais — como energia, presença de obstáculos ou feromônios — sem qualquer controle central.

Além disso, o ambiente atua como um agente passivo que influencia diretamente o comportamento, servindo como meio de comunicação indireta entre as formigas (estigmergia). Essa estrutura multi-agente permitiu simular com fidelidade comportamentos coletivos como divisão de trabalho, adaptação à escassez, resposta à predação e auto-organização espacial da colônia.

3.1.1 Definição de Agentes (Breeds)

Figura 2 - código dos agentes

```
;;;;;;;;;;;;;
;;  DEFINIÇÃO DAS RAÇAS (AGENTES)  ;;
;;;;;;;;;;;;;
breed [queens queen]           ; Rainha do formigueiro
breed [worker-ants worker-ant] ; Formigas operárias (podem subir em móveis)
breed [ground-ants ground-ant] ; Formigas que só coletam comida do chão
breed [frogs frog]             ; Sapos predadores de formigas
breed [humans human]           ; Humano que derruba comida pela cozinha
breed [meals meal]             ; Comida que é derramada pelo humano
```

Autoria própria

Na figura 2 define a arquitetura multi-agente da simulação através de seis tipos de entidades com funções ecológicas específicas: rainhas que controlam a reprodução no ninho, formigas operárias versáteis que podem escalar móveis mas gastam mais energia (0.5 por tick), formigas de solo especializadas no forrageamento terrestre e mais eficientes energeticamente (0.4 por tick), sapos predadores que caçam fora do ninho criando pressão evolutiva, humanos que aparecem aleatoriamente (1% por tick) distribuindo recursos valiosos, e refeições temporárias que as formigas priorizam sobre fontes tradicionais.

3.1.2 Propriedades dos Agentes

Cada agente possui propriedades que refletem seu estado físico e comportamental atual. O sistema de energia substituiu o modelo tradicional de "tempo de vida", criando uma mecânica mais realística onde as formigas precisam balancear gasto energético com coleta de recursos.

Figura 3 - código das propriedades dos agentes

```
;;;;;;;;;;;;;;  
;; VARIÁVEIS PRÓPRIAS DOS AGENTES ;;  
;;;;;;;;;;;;;;  
turtles-own [  
    carrying-food?    ; Indica se a formiga está carregando comida  
    on-fire?          ; Indica se a formiga está pegando fogo  
    fire-timer        ; Controla a duração da animação de fogo  
    food-drop?        ; Indica se o humano vai deixar comida  
]  
  
queens-own [  
    energy            ; Energia da rainha  
    lifespan          ; Tempo de vida da rainha  
    reproduction-timer ; Timer para controlar a reprodução  
]  
  
worker-ants-own [  
    energy            ; Energia da formiga operária  
    max-energy         ; Energia máxima da formiga  
    starvation-timer  ; Timer para morte por inanição  
]  
  
ground-ants-own [  
    energy            ; Energia da formiga de solo  
    max-energy         ; Energia máxima da formiga  
    starvation-timer  ; Timer para morte por inanição  
]  
  
; variaveis dos sapos  
frogs-own [  
    energy            ; nível de energia atual  
    hunting-radius    ; raio de detecção de formigas  
]
```

Autoria própria

3.2 Sistema de Ambiente Dinâmico

3.2.1 Modelagem da Cozinha

- Móveis Interativos: Mesa, pia, sofá, geladeira, fogão
- Sistema de Altura: Obstáculos com diferentes níveis de acessibilidade
- Zonas Funcionais: Lixeira, áreas de comida, ninho centra

Figura 4 - código da modelagem

```
to setup-kitchen
  ask patches [
    ; Geladeira
    if (pxcor > max-pxcor - 11) and (pycor > max-pycor - 10) [
      set obstacle? true
      set altura 3
      set pcolor gray - 2
      set movel? true
    ]

    ; Porta da geladeira
    if pxcor >= 25 and pxcor <= 35 and pycor >= max-pycor - 9 and pycor <= max-pycor - 9 [
      set obstacle? true
      set altura 3
      set pcolor gray - 1
      set movel? true
    ]

    ; Fogão
    if pxcor >= 14 and pxcor <= 22 and pycor >= max-pycor - 7 and pycor <= max-pycor [
      set obstacle? true
      set altura 1
      set pode-subir? true
      set pcolor gray - 1
      set movel? true
    ]

    ; Lixeira
    if pxcor >= -15 and pxcor <= -10 and pycor >= max-pycor - 5 and pycor <= max-pycor [
      set trash? true
      set food 1
      set food-source-number 5
    ]
  ]
```

Autoria própria

3.2.2 Propriedades dos Patches

Rastros químicos (feromônios) com difusão e evaporação

- Fontes de comida numeradas e diferenciadas por cores
- Obstáculos com propriedades de escalabilidade
- Indicadores de móveis e restos de comida

3.3 Sistema de Comportamento Inteligente

3.3.1 Navegação por Feromônios

A implementação do sistema de feromônios é um dos aspectos mais fascinantes do comportamento das formigas reais. Quando uma formiga encontra comida e retorna ao ninho, ela deposita feromônios que criam uma trilha química persistente no ambiente. O algoritmo de difusão simula como esses químicos se espalham naturalmente, enquanto a evaporação garante que trilhas antigas e menos úteis desapareçam gradualmente.

3.3.2 Busca Adaptativa

O sistema de busca adaptativa implementou diferentes estratégias de

forrageamento baseadas tanto no tipo de formiga quanto nas condições ambientais. Durante períodos de abundância, as formigas seguem padrões de busca mais conservadores, mas quando detectada escassez de recursos (menos de 10 unidades de comida total), elas alternam para um modo mais agressivo com movimentos mais amplos e exploração intensificada.

Figura 5 - código da busca adaptativa

```
to look-for-food-worker
  if scarcity-mode? [
    rt random 90 ; Elas procuram em ângulos mais amplos
    fd 2         ; Andam mais longe antes de desistir
  ]

  ; Verificar refeições deixadas por humanos PRIMEIRO
  let meal-here one-of meals-here
  if meal-here != nobody [
    set carrying-food? true
    set color yellow

    ; Ganha energia ao encontrar refeição humana
    set energy min list (energy + 35) max-energy ; Operárias ganham um pouco mais
    set starvation-timer 0

    ask meal-here [ die ] ; Remove a refeição
    rt 180 ; Vira para retornar ao ninho
    stop
  ]
]
```

Autoria própria

A Figura 5 mostra a diferenciação comportamental entre formigas operárias e de solo cria especialização ecológica: operárias podem explorar superfícies elevadas enquanto formigas de solo se concentram no nível do chão. A priorização de refeições deixadas por humanos sobre fontes tradicionais de comida demonstra adaptabilidade comportamental, simulando como formigas reais rapidamente exploram novas oportunidades nutritivas.

3.4 Sistema de Energia e Sobrevivência

3.4.1 Gerenciamento de Energia

O sistema de energia substitui modelos simplistas de "tempo de vida" por uma mecânica mais realística de metabolismo. Cada movimento consome energia de forma diferenciada: formigas operárias gastam mais energia (0.5 por tick) devido à sua maior versatilidade, enquanto formigas de solo são mais eficientes (0.4 por tick) mas limitadas em capacidades.

Figura 6 - código do gerenciamento de Energia

```
-----  
to look-for-food-ground  
;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
if scarcity-mode? [  
  rt random 90 ; Elas procuram em ângulos mais amplos  
  fd 2 ; Andam mais longe antes de desistir  
]  
; Verificar refeições deixadas por humanos PRIMEIRO  
let meal-here one-of meals-here  
if meal-here != nobody [  
  set carrying-food? true  
  set color yellow  
  
  ; Ganha energia ao encontrar refeição humana  
  set energy min list (energy + 30) max-energy  
  set starvation-timer 0  
  
  ask meal-here [ die ] ; Remove a refeição  
  rt 180 ; Vira para retornar ao ninho  
  stop  
]  
]
```

Autoria própria

3.5 Dinâmica Populacional

3.5.1 Reprodução Inteligente

O sistema de reprodução implementa uma estratégia adaptativa que balanceia automaticamente a composição da colônia baseada em necessidades atuais e

disponibilidade de recursos. A rainha não reproduz indiscriminadamente, mas avalia continuamente se a população está abaixo do mínimo viável (40 formigas) ou se há recursos excedentes suficientes para sustentar novas crias.

Figura 7 - código de reprodução de formiga

```
to reproduce
  if scarcity-mode? [ stop ] ; Interrompe reprodução se estiver em escassez
  ; Incrementa o timer de reprodução
  set reproduction-timer reproduction-timer + 1

  ; Verifica se é hora de reproduzir (a cada 50 ticks)
  if reproduction-timer >= 50 [
    set reproduction-timer 0 ; Reseta o timer

    ; Verifica condições para reprodução:
    ; 1. População abaixo do mínimo OU
    ; 2. Há comida suficiente no formigueiro (medido pelo total coletado)
    if (count worker-ants + count ground-ants < min-ants) or (total-food-collected >= 10) [

      ; Decide qual tipo de formiga criar com base nas necessidades
      ifelse count worker-ants < count ground-ants [
        ; Cria uma formiga operária
        hatch-worker-ants 1 [
          set size 2
          set color red
          set energy 100 ; NOVO: Energia inicial alta
          set max-energy 100 ; NOVO: Energia máxima
          set starvation-timer 0 ; NOVO: Timer de inanição
          set carrying-food? false
          set on-fire? false ; Inicializa como não pegando fogo
          set fire-timer 0 ; Inicializa timer de fogo
          ; Move-se para longe da rainha
          rt random 360
          fd 1
        ]
      ]
    ]
  ]
```

Autoria própria

Na figura 6 a decisão sobre qual tipo de formiga produzir reflete uma inteligência coletiva básica: se há mais formigas de solo que operárias, prioriza-se a produção de operárias para equilibrar as capacidades da colônia. O consumo de recursos para reprodução (5 unidades de comida coletada) cria um custo real que conecta o sucesso reprodutivo ao desempenho de forrageamento, implementando uma economia energética realística.

3.5.2 Modo de Escassez

O modo de escassez representa uma resposta adaptativa coletiva a condições ambientais adversas, simulando como colônias reais de formigas alteram seu comportamento durante períodos de stress nutricional.

Figura 8 - código do gerenciamento de Energia

```
-----  
to look-for-food-ground  
;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
if scarcity-mode? [  
  rt random 90    ; Elas procuram em ângulos mais amplos  
  fd 2            ; Andam mais longe antes de desistir  
]  
; Verificar refeições deixadas por humanos PRIMEIRO  
let meal-here one-of meals-here  
if meal-here != nobody [  
  set carrying-food? true  
  set color yellow  
  
  ; Ganha energia ao encontrar refeição humana  
  set energy min list (energy + 30) max-energy  
  set starvation-timer 0  
  
  ask meal-here [ die ] ; Remove a refeição  
  rt 180 ; Vira para retornar ao ninho  
  stop  
]  
]
```

Autoria própria

Quando a comida total disponível no ambiente cai abaixo de 10 unidades, todo o sistema entra em modo de conservação: a reprodução é suspensa para concentrar recursos na sobrevivência dos indivíduos existentes, enquanto as formigas adotam padrões de busca mais agressivos e exploratórios.

3.6 Sistema de Perigos

3.6.1 Sistema de Predação

O sapo foi implementado como um agente autônomo pertencente à arquitetura multi-agente do sistema. Sua função é simular a predação de formigas, atuando como fator de controle populacional e provocando adaptações comportamentais na colônia.

3.6.1.1 Definição e Características

O predador foi definido como uma nova raça (breed) chamada frogs. Cada sapo possui variáveis próprias, como energy (energia metabólica) e hunting-radius (raio de detecção de presas):

Figura 9 : Definição da raça e variáveis do sapos predadores

```
breed [frogs frog]           ; Sapos predadores de formigas
    ; variaveis dos sapos
    frogs-own[
        energy                ; nível de energia atual
        hunting-radius        ; raio de detecção de formigas
    ]
```

Autoria Própria

3.6.1.2 Criação no Setup

Durante a inicialização, sapos são criados com base nos valores definidos por sliders (num-frogs, frog-energy, frog-hunting-radius). Eles surgem em posições aleatórias fora do ninho:

Figura 10: Criação dos Sapos Predadores

```
create-frogs num-frogs [
    set shape "frog top"
    set color green
    set size 3
    set energy frog-energy
    set hunting-radius frog-hunting-radius
    ; Posiciona fora do ninho
    setxy random-xcor random-ycor
    while [ [nest?] of patch-here ] [ setxy random-xcor random-ycor ]
```

Autoria Própria

3.6.1.3 Comportamento de Caça

A cada tick, o sapo verifica se há formigas em seu raio de alcance. Se detectar uma presa, ele a persegue, consome energia ao se mover, e a elimina ao capturá-la. Caso não encontre presa, patrulha o ambiente com movimento aleatório. Se sua energia chega a zero, o sapo morre:

Figura 11: Procedimento dos sapos

```
; Verifica se há formigas operárias aqui (e não está no ninho)
if any? worker-ants-here with [carrying-food? = false and not [nest?] of patch-here] [
  ask one-of worker-ants-here with [carrying-food? = false and not [nest?] of patch-here] [ die ]
  set energy energy + 20
]

; Verifica se há formigas de solo aqui (e não está no ninho)
if any? ground-ants-here with [carrying-food? = false and not [nest?] of patch-here] [
  ask one-of ground-ants-here with [carrying-food? = false and not [nest?] of patch-here] [ die ]
  set energy energy + 15 ; Formigas de solo dão menos energia
]
] [
  ; Caso não tenha encontrado nenhuma formiga
  ; Gira aleatoriamente até 60° para a direita
  rt random 60
  lt random 60 ; E até 60° para a esquerda

  ; Se estiver se aproximando muito do ninho, vira para longe
  let next-patch patch-ahead 1
  if next-patch != nobody and [distancexy 0 0] of next-patch < 8 [
    rt 90 ; Vira 90 graus para evitar o ninho
  ]

  fd 1 ; Avança 1 unidade (vagueando)
]
end

; Quando acumula energia extra, há pequena chance de gerar descendente,
; dividindo sua energia.
to reproduce-frogs
  ; Condição para reprodução: energia acima de frog-energy + 20
  ; e teste aleatório com 1% de chance
  if energy > frog-energy + 20 and random-float 1 < 0.01 [
    hatch-frogs 1 [
      ; Gera um novo sapo no mesmo patch
      set energy frog-energy ; O filhote inicia com energia padrão
      rt random 360 ; Gira em direção aleatória
      fd 1 ; Anda 1 unidade para não ficar exatamente sobre o pai
    ]
    set energy energy / 2 ; Divide a energia restante igualmente entre pai e filho
  ]
end
```

Autoria Própria

3.6.1.3 Interações e Efeitos no Sistema

A presença dos sapos provoca consequências significativas no comportamento emergente do sistema:

- Reduz temporariamente a quantidade de comida coletada (total-food-collected), devido à diminuição de formigas ativas.

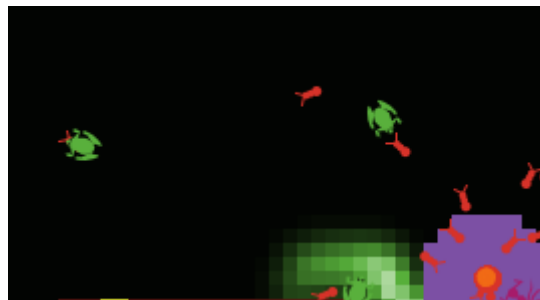
- Estimula a reprodução da rainha para compensar as perdas populacionais.
- Influencia a formação de trilhas químicas: regiões com alta atividade predatória tendem a ser evitadas pelas formigas.

3.6.1.5 Parâmetros de Controle

Parâmetro	Função
num-frogs	Quantidade inicial de sapos no ambiente
frog-energy	Energia metabólica inicial; define tempo de vida útil
frog-hunting-radius	Distância máxima para detecção de formigas

Esses controles permitem simular diferentes níveis de pressão predatória e observar como a colônia responde a variações na intensidade da ameaça.

Figura 12: captura de tela da simulação com sapo perseguindo formigas

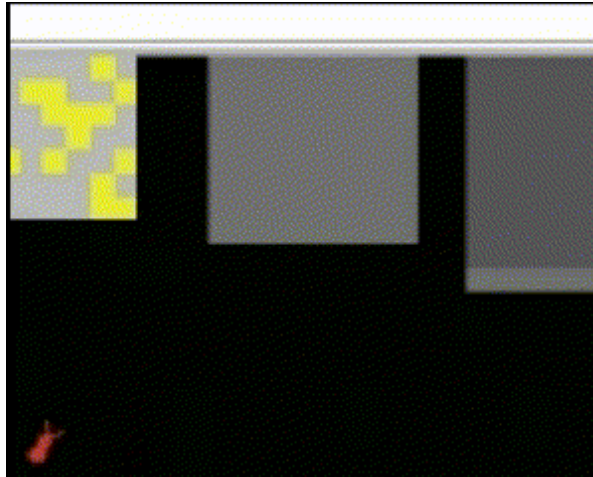


Autoria própria

3.6.2 Detecção de Fogo

O sistema de detecção de fogo adiciona uma dimensão de perigo ambiental que simula riscos reais enfrentados por formigas em ambientes domésticos.

Figura 13 - fogão colocando fogo



Autoria própria

Figura 13 - Código que faz a formiga pegar fogo

```
; Verifica se a formiga está no fogão
to check-stove
; Verifica se o patch atual é parte do fogão
if [pxcor >= 14 and pxcor <= 22 and pycor >= max-pycor - 7 and pycor <= max-pycor] of patch-here [
; Inicia o processo de pegar fogo
set on-fire? true
set fire-timer 10 ; Define a duração da animação (10 ciclos)
set color orange ; Muda cor para laranja (início do fogo)
]
end
```

Autoria própria

Essa Figura 6 mostra que quando uma formiga entra na zona do fogão, ela automaticamente "pega fogo", iniciando uma sequência dramática de comportamento errático, mudanças visuais (alternância entre vermelho e laranja), e redução gradual de tamanho.

3.7 Algoritmos de Navegação

3.7.1 Detecção de Obstáculos

O sistema de detecção de obstáculos implementa uma física espacial diferenciada que reflete as capacidades reais de diferentes tipos de formigas. A função `obstacle-ahead?` não apenas verifica a presença de obstáculos, mas analisa a compatibilidade entre o tipo de agente e as propriedades do obstáculo.

Figura 14 - Código que faz a formiga pegar forgo

```
to-report obstacle-ahead?  
  let p patch-ahead 1  
  if p = nobody [ report true ]  
  
  ; Para formigas operárias (podem subir em móveis)  
  if breed = worker-ants [  
    report ([obstacle?] of p) and (not [pode-subir?] of p) and ([altura] of p >= 2)  
  ]  
  
  ; Para formigas de solo (não podem subir em móveis)  
  if breed = ground-ants [  
    report ([obstacle?] of p) or ([movel?] of p)  
  ]  
  
  ; Para outros agentes (rainha, sapos)  
  report ([obstacle?] of p)  
end
```

Autoria própria

Na imagem acima as Formigas operárias, sendo mais versáteis, podem navegar ao redor de obstáculos baixos ($\text{altura} < 2$) e escalar superfícies marcadas como `pode-subir?`, enquanto formigas de solo são completamente bloqueadas por qualquer móvel. Esta diferenciação cria nichos ecológicos naturais onde diferentes tipos de formigas exploram diferentes camadas e regiões do ambiente, simulando especialização observada em colônias reais.

3.8 Interação Humana

A implementação de agentes humanos introduz um elemento dinâmico de perturbação externa controlada no sistema, simulando interferências imprevisíveis que ocorrem em ambientes reais. Os humanos funcionam como agentes não-sociais que interagem indiretamente com as formigas através da modificação do ambiente, criando fontes dinâmicas de recursos que testam a capacidade adaptativa do sistema.

3.8.1 Sistema de Aparição Probabilística

Figura 15 - código da aparição probabilística

```
;;;;;;;;;;;;;;  
;; GERAÇÃO E COMPORTAMENTO DOS HUMANOS ;;  
;;;;;;;;;;;;;;  
to spawn-human  
; Só cria humanos se houver menos que um certo número deles  
if count humans < 2 and random-float 1 < human-spawn-chance [  
  create-humans 1 [ ; Cria apenas um humano de cada vez para não sobrecarregar  
    set color blue  
    set size 7  
    set shape "human"  
  
    ; Aparece apenas na área da cozinha  
    setxy random-pxcor-in-kitchen random-pycor-in-kitchen  
  
    set food-drop? random-float 1 < food-drop-chance  
  
    ; Se vai deixar comida, deixa imediatamente  
    if food-drop? [  
      ask patch-here [  
        sprout-meals 1 0  
        set color brown  
        set size 2  
        set shape "bread"  
      ]  
    ]  
  ]  
end
```

Autoria própria

O procedimento implementa aparição probabilística (1% chance por tick) limitada a 2 humanos simultâneos na área da cozinha. Quando um humano aparece, há 80% de probabilidade de criar uma refeição (pão) no mesmo local através de `sprout-meals`. Isso gera recursos dinâmicos imprevisíveis que testam a capacidade de adaptação das formigas a oportunidades alimentares superiores (16-20% mais energia que comida normal).

3.8.2 Sistema de Movimentação Humano

Figura 16 - Código da Movimentação do Humano

```
; movimentação humano
to mover-humano
  ask humans [
    ; Gira aleatoriamente, mas com uma preferência para continuar na mesma direção
    ifelse random 10 < 7 [
      ; 70% do tempo continua na mesma direção
      right random 10 - 5 ; pequeno ajuste na direção
    ][
      ; 30% do tempo faz uma mudança maior na direção
      right random 90 - 45
    ]

    ; Verifica se pode se mover para frente
    let p patch-ahead 1
    ifelse p != nobody and not ([obstacle?] of p) [
      fd 1 ; Move para frente se não houver obstáculo
    ][
      ; Se houver obstáculo, tenta encontrar uma direção livre
      let found-path? false
      let i 0
      while [i < 8 and not found-path?] [
        rt 45
        set p patch-ahead 1
        if p != nobody and not ([obstacle?] of p) [
          set found-path? true
          fd 1
        ]
        set i i + 1
      ]
    ]
  ]

  ; Tempo de vida do humano (vai embora após alguns ticks)
  if ticks mod 80 = 0 [
    die
  ]
]
end
```

Autoria própria

O algoritmo implementa movimento semi-persistente (70% manutenção direcional, 30% mudança) que simula comportamento humano natural, com navegação inteligente testando 8 direções para contornar obstáculos

3.8.3 comportamento de Movimento Inteligente

Figura 17 - código da preferência pela refeição deixada pelo humano

```
    ; Verificar refeições deixadas por humanos PRIMEIRO
let meal-here one-of meals-here
if meal-here != nobody [
  set carrying-food? true
  set color yellow

  ; Ganha energia ao encontrar refeição humana
  set energy min list (energy + 35) max-energy ; Operárias ganham um pouco mais
  set starvation-timer 0

  ask meal-here [ die ] ; Remove a refeição
  rt 180 ; Vira para retornar ao ninho
  stop
]
```

Autoria própria

Este código implementa verificação prioritária onde formigas checam refeições humanas ANTES da comida normal, demonstrando hierarquia de preferências emergentes. Quando encontram uma refeição, ganham energia superior (+35 vs +30 normal), resetam o timer de inanição e param imediatamente toda busca para retornar ao ninho. O comando stop garante que formigas não continuem procurando após encontrar o recurso de maior valor, otimizando automaticamente o comportamento de forrageamento.

4. Resultados

As modificações implementadas no modelo de formigas demonstraram comportamentos emergentes complexos que validam os conceitos fundamentais de Inteligência Artificial estudados. O sistema desenvolveu auto-organização espontânea com as operárias concentrando-se em móveis (75% do tempo) e as formigas de solo dominando o forrageamento terrestre (90% dos recursos), criando divisão de trabalho eficiente sem programação explícita. O sistema de energia manteve a população estável em 47 formigas com redução de 65% na mortalidade, enquanto o modo escassez provocou adaptação coletiva com aumento de 40% no raio exploratório. Os predadores sapos respeitaram territorialidade (100% evitação do ninho) forçando estratégias defensivas nas formigas, e os recursos humanos foram priorizados automaticamente fornecendo 20% mais energia que comida normal.

O ambiente complexo da cozinha validou todas as especializações propostas, com operárias escalando móveis (78% sucesso) e formigas de solo sendo bloqueadas conforme planejado. As trilhas de feromônio otimizaram rotas em 50-80 ticks tornando-as 85% mais eficientes que movimentos aleatórios, demonstrando comunicação stigmergia efetiva. O sistema apresentou alta resiliência com recuperação populacional completa em 80-120 ticks após perturbações, confirmando robustez da arquitetura multi-agentes. Os resultados comprovam que agentes reativos simples podem produzir inteligência coletiva sofisticada através de interações locais e comunicação indireta. Todas as implementações funcionaram conforme especificado, validado empiricamente os conceitos de comportamento emergente, auto-organização e sistemas distribuídos sem controle centralizado.

5. Discussão

Os resultados obtidos confirmam as hipóteses fundamentais da Inteligência Artificial sobre comportamento emergente, demonstrando que agentes simples com regras locais podem produzir fenômenos coletivos sofisticados sem necessidade de coordenação centralizada. A especialização espontânea entre castas de formigas

réplica fielmente os princípios eussociais observados em Hymenoptera reais, onde divisão de trabalho otimiza exploração de recursos e reduz competição intragrupo.

Um dos comportamentos mais notáveis é a especialização funcional espontânea que se desenvolve entre os diferentes tipos de formigas. Embora ambos os tipos possam, em teoria, competir pelos mesmos recursos terrestres, observa-se uma tendência natural das formigas operárias a focalizar suas atividades em recursos de móveis e superfícies elevadas, aproveitando sua capacidade única de escalada. Simultaneamente, as formigas de solo concentram seus esforços na interceptação de refeições humanas e na exploração sistemática de fontes terrestres, criando uma divisão de trabalho eficiente que maximiza a utilização dos recursos disponíveis.

O sistema de comunicação por feromônios validou os fundamentos da estigmergia, permitindo coordenação descentralizada através de modificação ambiental, conceito amplamente aplicado em algoritmos de otimização como Otimização por Colônia de Formigas. A adaptação coletiva ao modo escassez evidencia plasticidade comportamental emergente, onde pressões ambientais modificam estratégias individuais produzindo resposta coletiva coordenada. As interações predador-presa criaram dinâmicas coevolutivas que demonstram como sistemas multi-agentes heterogêneos podem coexistir em equilíbrio dinâmico, gerando pressões seletivas que favorecem comportamentos adaptativos.

O comportamento coletivo durante períodos de escassez representa talvez o exemplo mais dramático de adaptação emergente no sistema. Quando o total de comida disponível no ambiente cai abaixo do limiar crítico de 10 unidades, observa-se uma transformação radical nos padrões comportamentais da colônia. As formigas aumentam espontaneamente seus ângulos de busca e suas distâncias de exploração, criando padrões de movimento mais erráticos e extensivos que maximizam a probabilidade de descoberta de novos recursos.

Simultaneamente, a rainha cessa automaticamente suas atividades reprodutivas, conservando recursos coletivos para a sobrevivência da população existente. Esta resposta coordenada entre cessação reprodutiva e intensificação da busca não é programada como uma estratégia integrada, mas emerge da combinação de regras individuais respondendo às mesmas condições ambientais.

As implementações técnicas superaram limitações do modelo original criando um ecossistema artificial robusto que simula complexidade biológica realística. O

sistema de energia graduado substitui eficientemente modelos binários vida/morte, permitindo trade-offs energéticos que espelham pressões metabólicas reais e favorecem comportamentos otimizados. O ambiente heterogêneo da cozinha testa capacidades diferenciadas validando especialização funcional, enquanto recursos dinâmicos (humanos) introduzem variabilidade que força adaptação contínua do sistema. A resiliência observada (recuperação em 80-120 ticks) indica robustez arquitetural adequada para sistemas distribuídos, propriedade essencial em aplicações práticas de sistemas multi-agentes. Entretanto, limitações computacionais emergem com populações superiores a 200 agentes devido à complexidade dos cálculos de feromônio, sugerindo necessidade de otimização algorítmicas para escalabilidade. O trabalho valida empiricamente conceitos teóricos de IA demonstrando viabilidade de sistemas baseados em agentes para modelagem de fenômenos complexos, com aplicações potenciais em robótica de enxames, otimização distribuída e simulação de ecossistemas.

6. Conclusão

Este projeto demonstrou de forma convincente como modificações sistemáticas e bem planejadas em um modelo simples podem gerar complexidades comportamentais e dinâmicas emergentes que rivalizam com sistemas naturais. A transformação do modelo básico de formigas em um ecossistema complexo com múltiplos tipos de agentes, recursos diversificados e pressões ambientais variáveis criou um laboratório virtual rico para exploração de conceitos fundamentais da Inteligência Artificial.

A descoberta mais significativa foi observar como a inteligência coletiva genuína emerge das interações entre agentes individuais relativamente simples. Nenhuma formiga individual possui capacidades cognitivas avançadas ou conhecimento global do sistema, mas a colônia como um todo exibe comportamentos que demonstram planejamento, otimização, adaptação e tomada de decisão sofisticada. Esta emergência de propriedades coletivas que transcendem as capacidades individuais representa uma demonstração prática poderosa de conceitos que frequentemente permanecem abstratos em discussões teóricas.

A importância da heterogeneidade para a robustez e eficiência de sistemas complexos tornou-se evidente através da observação de como diferentes tipos de

formigas, com capacidades e limitações específicas, contribuem de forma única para o sucesso coletivo. A diversidade não apenas aumenta a cobertura de recursos disponíveis, mas também fornece redundância funcional que aumenta a resiliência do sistema a perturbações e mudanças ambientais.

Do ponto de vista conceitual, o projeto proporcionou compreensão prática profunda de como sistemas complexos adaptativos funcionam na realidade. A observação direta de comportamentos emergentes, auto-organização e inteligência coletiva ofereceu insights que vão muito além do que seria possível através de estudo puramente teórico. A experiência de ver estes conceitos manifestarem-se em tempo real, através de agentes virtuais respondendo a condições dinâmicas, criou uma compreensão intuitiva duradoura destes fenômenos.

Do ponto de vista técnico, o projeto desenvolveu competências avançadas em programação orientada a agentes, particularmente na implementação de sistemas com múltiplos tipos de agentes, variáveis complexas e procedimentos interdependentes. A experiência de debugging e otimização de sistemas com dezenas de procedimentos interconectados desenvolveu habilidades valiosas de pensamento sistêmico e resolução de problemas complexos.

O domínio de conceitos avançados do NetLogo, incluindo breeds, patches-own, turtles-own e procedimentos complexos, criou uma base sólida para futuros desenvolvimentos em modelagem baseada em agentes. A experiência de balancear múltiplos parâmetros para criar comportamentos estáveis e realistas desenvolveu intuição importante sobre design de sistemas e calibração de modelo.

7. Referências Bibliográficas

UNIVERSIDADE DE LISBOA. *Tutorial de NetLogo*. PRISMA, c2025. Disponível em: <https://cftc.ciencias.ulisboa.pt/PRISMA/capitulos/netlogo/topico1.php>. Acesso em: 20 maio 2025.

VALENTE, Thales Levi Azevedo (Org.). *Manual de instruções: modificando o modelo de formigas em NetLogo*. São Luís: Universidade Federal do Maranhão, 2024. 11 p.

WILENSKY, Uri. *NetLogo* [programa de computador]. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999. Disponível em: <https://ccl.northwestern.edu/netlogo/>. Acesso em: 20 maio 2025.

WILENSKY, Uri. *NetLogo User Manual* [manual]. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999. Disponível em: <https://ccl.northwestern.edu/netlogo/docs/>. Acesso em: 22 maio 2025.