

# (IIA) Teórica NetLogo

From WikiNote

## Contents

- 1 O que é o NetLogo
  - 1.1 Interface do NetLogo
- 2 Agente
  - 2.1 Propriedades dos Agentes(turtles)
- 3 Patch
- 4 Estrutura do NetLogo
- 5 Alguns comandos
  - 5.1 Inicializações
  - 5.2 Movimentação
  - 5.3 Instrução ask
  - 5.4 Procedimentos
  - 5.5 Funções
  - 5.6 Declaração de variáveis
  - 5.7 Comando with
  - 5.8 Criar espécies
  - 5.9 Condições

## O que é o NetLogo

O NetLogo é um ambiente de programação para modelação de sistemas multiagente. O NetLogo é baseado no conceito de agente. Durante a simulação existem interações entre os agentes e o ambiente, que são devidas ao comportamento dos agentes.

## Interface do NetLogo

- Interface
  - Onde se interage com a simulação do modelo
- Info
  - Onde se coloca a documentação do modelo
- Code
  - Onde se faz a codificação das funções do modelo

**Os componentes (Switch, slides, etc) são variáveis**

**Os botões estão associados a procedimentos**

### NOTA

*No code estamos sempre ao nível do programador, se quisermos alterar um agente temos de entrar como agente usando uma instrução.*

## Agente

Um agente é uma entidade autónoma com comportamento próprio.

## Propriedades dos Agentes(turtles)

Aqui em baixo vão estar algumas propriedades úteis dos agentes.

- No heading (**valor entre 0° e 360°**)
- Cor ( **RGB ou escrever a cor** )
- Shape ( **String válida** )

Para ver as strings válidas basta ir a **Tools > Turtle Shape Editor**

## Patch

Uma patch é basicamente uma célula do mundo

As patches têm propriedades e para as manipular basta carregar em cima de uma com o botão do lado direito do rato e faz-se **inspect path** podendo fazer exatamente o mesmo a um agente.

## Estrutura do NetLogo

O NetLogo têm 3 tipos de elementos:

- Turtles
  - Agentes que se movem no ambiente 2D
- Patches
  - Células que constituem o ambiente 2D
- Observador
  - Observa o mundo e pode atuar sobre ele (programador)

## Alguns comandos

Normalmente começamos com **ca** para limpar o mundo sempre que existe uma nova iteração.

### Inicializações

*Apaga todos os agentes*

```
clear-turtles ou ct
```

*Limpa o ambiente 2D*

```
clear-patches ou cp
```

*Criar n agentes*

```
create-turtles n ou crt n
```

*Atribuir valores a uma variavel e/ou a uma propriedade*

```
set var value
```

Para atribuir valores a uma variavel e/ou a uma propriedade usamos o **set**

**Exemplo : Atribuir +1 a uma variavel chamada "energia"**

```
set energia energia +1
```

### Movimentação

*O agente avança n unidades segundo a sua orientação*

```
forward n ou fd n
```

*O agente recua n unidades*

```
back n ou bk n
```

*Virar para a direita ou para a esquerda*

```
left n ou lt n  
right n ou rt n
```

*Move-se até à posição do agente de nome agent*

```
move-to agent
```

*O agente vai para uma posição aleatória não saindo das dimensões do mundo*

```
setxy random-xcor random-ycor
```

## Instrução ask

Para alterar uma propriedade de um agente e/ou patch usamos o comando **ask**

*Todos os agentes avançam uma unidade*

```
ask turtles [fd 1]
```

*Todas as patches são pintadas de vermelho*

```
ask patches [set pcolor red]
```

*Apenas o agente com o identificador 4 vira 90° à direita*

```
ask turtle 4 [rt 90]
```

## Procedimentos

**Não retornam nada.**

```
to NomeProc [parametro 1 parametro 2]
```

```
Comandos ...  
Comandos ...  
Comandos...
```

```
end
```

**Não Retorna nada**

## Funções

**Retornam um valor**

T to-report NomeFunc [parametro 1 parametro 2]

Comandos...  
Comandos...  
Comandos...

Retorna um valor

report valor  
end

## Declaração de variáveis

Podemos dar propriedades novas aos agentes e aos patches, no entanto devemos colocar sempre no topo do código estas instruções.

*Variáveis específicas a cada patch*

```
patches-own [var1 ...]
```

*Variáveis específicas a cada agente*

```
turtles-own [var1 ...]
```

## Comando with

Podemos alterar ou comandar apenas alguns agentes e para isso usamos o comando with

*Todos os agentes com coordenada x maior que zero*

```
turtles with [xcor > 0]
```

**Exemplo mais prático**

```
ask turtles with [color = red]
[
  set color blue
]
```

Vê os agentes com cor vermelha e muda a sua cor para azul

## Criar espécies

Em vez de trabalhar apenas com agentes do tipo turtle, podemos especificar espécies com outros nomes usando o comando "breed"

*Syntax*

```
breed [nomePlural nomeSingular]
```

*Cria agentes do tipo male*

```
breed [males male]
```

`ask male 3 [set color red]` } O male 3 é pintador de vermelho

`ask males [set color green]` } Todos os males (agentes) são pintados de verde

## Condições

*Se a condição for verdadeira executa os comandos*

```
if condição [comandos]
```

*Se o if for verdade executa o comando 1, se o if for falso executa o comando2*

```
ifelse condicao [comando1] [comando 2]
```

*Enquanto for verdade executa comandos*

```
while [condicao][comandos]
```

*Se a patch onde está o agente for verde, põe a patch a preto e aumenta a energia dele*

```
ask turtles
[
  if pcolor = green
  [
    set pcolor black
    set energia (energia + 10)
  ]
]
```

Retrieved from "[http://zebisnaga.pt/wiki/index.php?title=\(IIA\)\\_Teórica\\_NetLogo&oldid=136](http://zebisnaga.pt/wiki/index.php?title=(IIA)_Teórica_NetLogo&oldid=136)"