

2023/2024



## Trabalho Prático 1: Agentes

### Engenharia Informática

*Inteligência Artificial*

**Corpo Docente:**  
**José Oliveira**  
**Eduardo Pires**



al70759 Cláudia Silva  
al71134 Pedro Barroso

## Objetivos

---

- Desenvolver competências fundamentais relativas à modelação e simulação computacional de sistemas com agentes racionais utilizando a ferramenta NetLogo.

## Índice

<b>Objetivos .....</b>	<b>1</b>
<b>Escolha do Vírus .....</b>	<b>2</b>
SARS-CoV-2 .....	2
Informações relevantes sobre a SARS-CoV-2 .....	2
<b>Interface – Etapa 1 .....</b>	<b>3</b>
<b>Interface – Etapa 2 .....</b>	<b>6</b>
<b>Conclusão .....</b>	<b>9</b>

---

## Escolha do Vírus

---

### SARS-CoV-2

A escolha de utilizar o COVID-19 como alvo da simulação num campus universitário é motivada pelo impacto significativo que a doença teve na comunidade académica pois a pandemia afetou a vida dos estudantes, professores e funcionários, alterando drasticamente a dinâmica do campus.

Assim sendo queremos simular algo que nos afetou diretamente.

### Informações relevantes sobre a SARS-CoV-2

Este vírus tem **um período de incubação** geralmente de 4 a 5 dias, podendo variar de 2 a 14 dias;

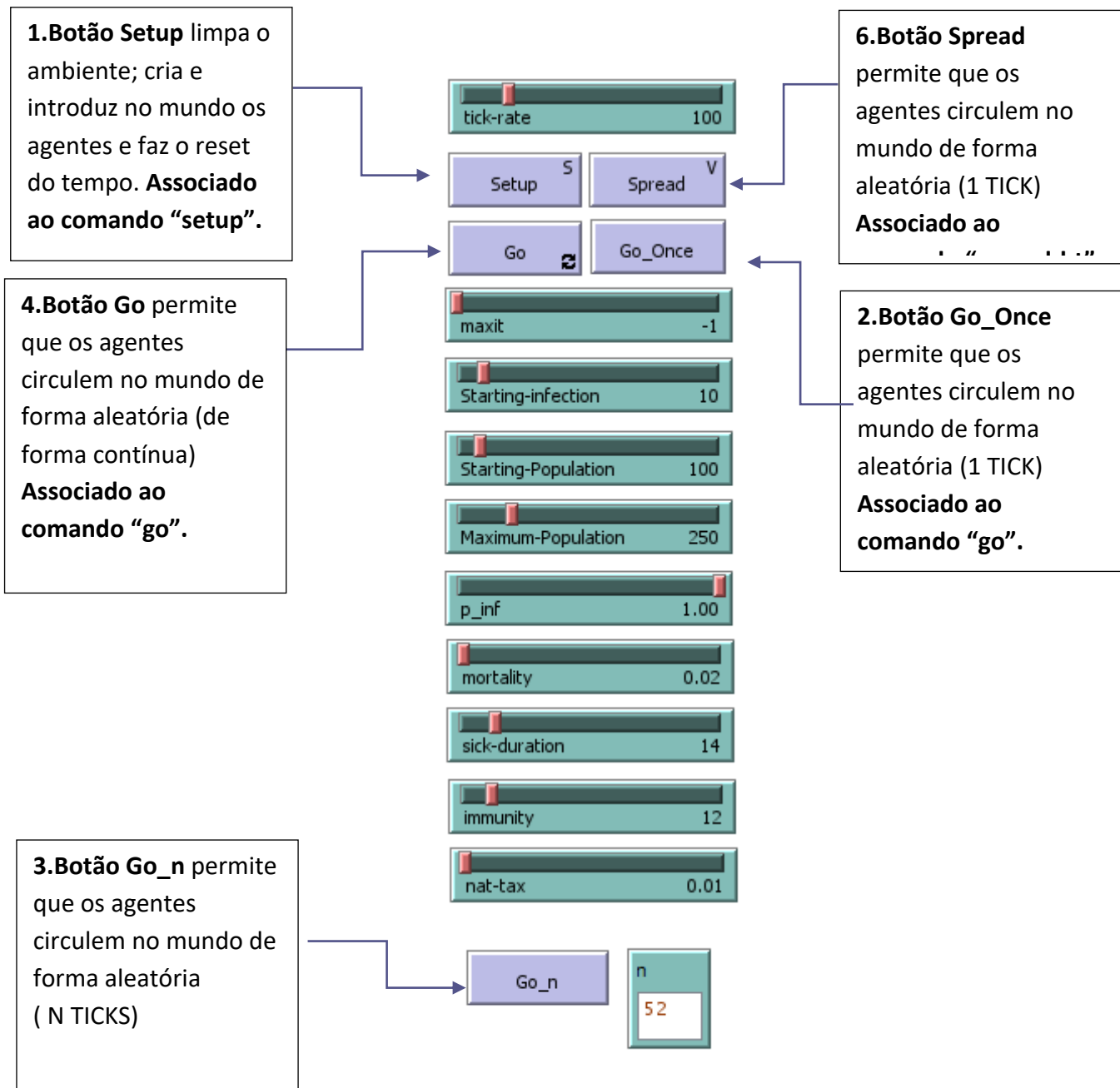
O período de doença é de 10 a 14 dias;

Tem um período de imunidade de 30 a 90 dias;

Atualmente em Portugal cerca **de 90% da população vacinada** com a vacina primária.

Cerca de 75-80% da população em Portugal usou máscara diariamente.

## Interface - Etapa 1

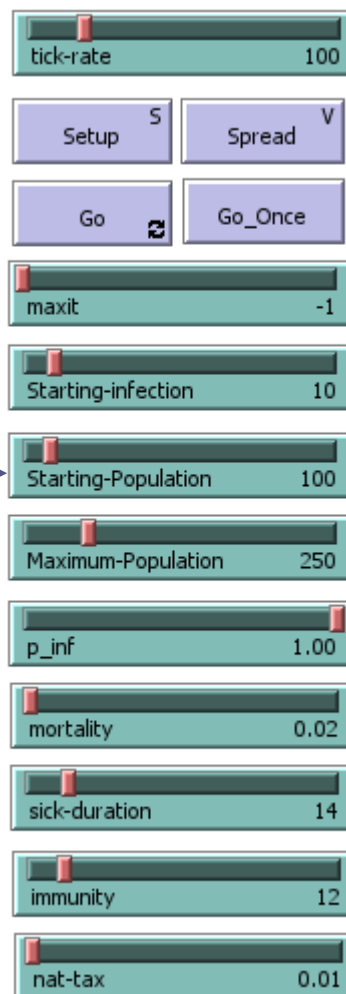


7. Um agente saudável fica contaminado se estiver na mesma célula de um agente contaminado, desde que o teste probabilístico seja favorável.

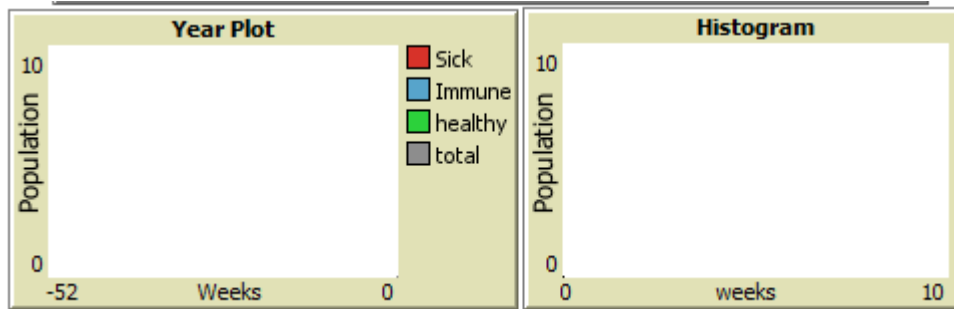
```
to spread-sick ;pessoas procedure
  ask other pessoas-here with [virus? = false and immune? = false] [
    if random-float 1 < p_inf [ get-sick ]
  ]
end
```

**8.Deslizador Starting-Population** permite alterar o número de agentes iniciais.

**9.Deslizador maxit** permite definir o número N de iterações.



10. Um gráfico que permita visualizar a evolução dos agentes contaminados.



11. Um contador que apresente o número de agentes infectados;

Infected_Total 10	sick% 10	Sick_Now 10
Recovered_Total 0	immune% 0	Immune_Now 0
Mortality_Total 0	healthy% 90	Healthy_Now 90
born 0		Idade Media 24.7
time-years 0		

12. Ao fim de X dias (a definir conforme o vírus) um agente contaminado pode ficar recuperado.

```
to get-healthy ;pessoas procedure
  set virus? false
  set color blue
  set immune-duration immunity - random (immunity / 5) + random (immunity / 5)
end
```

13. Um agente contaminado pode morrer por complicações associadas.

```
to immune-or-die ;pessoas procedure
  ifelse random-float 1 < mortality [ set dead dead + 1 die ] [ set recovered recovered + 1 get-healthy ]
end
```

---

## Interface - Etapa 2

---

Para tornar a nossa simulação mais enriquecida decidimos implementar algumas ideias como:

### 14. Utilização de máscaras;

```
to reset-mask
  ask pessoas [set mask? false]
  ask n-of (count pessoas * (mask% / 100)) pessoas [set mask? true]
end

to spread-sick ;pessoas procedure
  ask other pessoas-here with [virus? = false and immune? = false] [
    ifelse mask? [if random-float 1 < p_inf * 0.25 [ get-sick ]] [if random-float 1 < p_inf [ get-sick ]]
  ]
end
```

Com a introdução de uma percentagem de pessoas que usam máscara diariamente a propagação da doença diminui em cerca 75% logo a probabilidade de infetar um agente “**p\_inf**” é multiplicada por 0,25. Portanto se um agente estiver a usar mascara o p\_inf será o novo valor enquanto se não estiver a usar máscara será a probabilidade definida no início da simulação.

## 15. Introdução de vacinação;

```
to vaccination
  ifelse (count pessoas * (Vax/day / 100)) > count pessoas with [not (virus? and not incubation?) and not vac?]
  [
    ask pessoas with [not (virus? and not incubation?) and not vac?]
    [
      set vac-duration (- (Incubation-time - random (Incubation-time / 3) + random (Incubation-time / 3)))
      set color violet
      set vaccinated vaccinated + 1
    ]
  ]
  ask n-of (count pessoas * (Vax/day / 100)) pessoas with [not (virus? and not incubation?) and not vac?]
  [
    set vac-duration (- (Incubation-time - random (Incubation-time / 3) + random (Incubation-time / 3)))
    set color violet
    set vaccinated vaccinated + 1
  ]
end

set vac-duration vac-duration + 1
if vac-duration = 0 [get-healthy]
end
```

Com a introdução da vacina um valor chamado "vac-duration" é definido como -7, o que indica que a pessoa foi vacinada. A cor da pessoa muda para indicar isso.

Depois, a cada iteração, o valor de "vac-duration" aumenta em 1 unidade, representando o tempo decorrido desde a vacinação. Quando "vac-duration" atinge 0, isso significa que a proteção da vacina acabou, e a pessoa se torna saudável novamente.

Essas pessoas podem ser vacinadas novamente após 90 dias. O ciclo continua permitindo que as pessoas sejam vacinadas repetidamente para manter a proteção ao longo do tempo. O "vac?" é definido como falso após 90 dias para permitir que a pessoa seja vacinada novamente.



---

## 16.Introdução de períodos de incubação;

```
to get-sick ;pessoas procedure
  set virus? true
  set color orange
  set incubation-duration Incubation-time - random (Incubation-time / 3) + random (Incubation-time / 3)
  set virus-duration sick-time - random (sick-time / 2) + random (sick-time / 2)
  set immune-duration 0
  set infected infected + 1
end
```

Quando um agente entra no patch de outro agente não fica logo infetado ele entra num estado de incubação e como em situação real nem todos têm o mesmo período de incubação acrescentou-se set a um random fazendo o período variar de 4 a 10 dias. Após esse período o agente passa a estar doente.

## 17.Introdução de períodos de imunidade;

## 18.Implicação da idade na mortalidade;

```
to progress ;pessoas procedure
  ifelse incubation? [ set incubation-duration incubation-duration - 1 ]
  [
    ifelse sick? [ set virus-duration virus-duration - 1 set color red ] [ immune-or-die ]
  ]
end

to immune-or-die ;pessoas procedure
  ifelse age < 20 * 52 * 7
  [
    ifelse random-float 100 * ( age / 20 * 52 * 7 ) < mortality [ set dead dead + 1 die ] [ set recovered recovered + 1 get-healthy ]
  ]
  [
    ifelse random-float 100 < mortality [ set dead dead + 1 die ] [ set recovered recovered + 1 get-healthy ]
  ]
end

to get-healthy ;pessoas procedure
  set virus? false
  set incubation-duration 0
  set color blue
  set immune-duration immunity - random (immunity / 3) + random (immunity / 3)
end
```

Após o tempo de doença acabar o agente tem duas opções ou ele torna-se imune durante um período de tempo ou morre.

Tendo em conta que em Portugal a esperança media de vida é de 80 anos os agentes têm life span de número aleatório até 80, portanto quando esse número definido chegar a 0 o agente morre ou seja um agente que tenha a idade  $20(\text{life span restante}) * 52(\text{semanas}) * (\text{dias})$  tem 60 anos e se o agente tiver então mais de 60 anos a probabilidade de ele morrer é aumentada por pertencer a um grupo de risco.

Caso o agente não morra ele entra num estado de imunidade que pode ser definida por um slider na interface do usuário, mas que por nós está definida para 90 dias como é o consultado para a nossa doença escolhida. Durante esse período o agente não consegue ficar doente.

## 19.Introdução de reprodução.

```
to reproduce ;pessoas procedure
  if random-float 1 < nat-tax and count other pessoas-here > 0 and count turtles < Maximum-Population
  [ hatch 1
    [
      set age 80 * 52 * 7 + random 10 * 52 * 7 - random 10 * 52 * 7
      lt 90 fd 1
      get-healthy
      set immune-duration 0
      set vac-duration 90
      set color green
      set born born + 1
    ]
  ]
end
```

## Conclusão

---

Após termos feito várias simulações com variáveis diferentes chegamos á conclusão de que o uso de softwares de simulação é essencial para a resolução de certos problemas como a propagação de um vírus.