

# Simulação epidemiológica

## Introdução

Modelos epidemiológicos computacionais (MEC) são utilizados para a compreensão da dinâmica de uma epidemia ou afim, e são, portanto, ferramentas essenciais na determinação de políticas públicas para combate e prevenção de doenças. A transmissão de uma doença infecciosa pode ocorrer por diversas formas, como contato direto entre indivíduos, por via aérea, por contato com superfícies contaminadas, via vetor de transmissão, dentre outros.

Um MEC bastante utilizado é conhecido como Suscetível-Infectado-Removido (SIR), descrito de forma simplificada a seguir. Considere 3 populações:

- **S**: número de indivíduos suscetíveis (que ainda não estão contaminados);
- **I**: número de indivíduos infectados (capazes de infectar indivíduos S);
- **R**: número de indivíduos removidos (que se recuperaram, tornaram-se imunes ou faleceram).

Nesse modelo, um indivíduo da população S pode permanecer em S ou ir para a população I, caso seja infectado. Um indivíduo de I pode, então, se recuperar ou falecer. Indivíduos que pertençam a R não mudam de população. A interação entre essas populações pode ser descrita pelas equações abaixo:

$$S(t) = S(t-1) - h.b.S(t-1).I(t-1)$$

$$I(t) = I(t-1) + h.[b.S(t-1).I(t-1) - k.I(t-1)]$$

$$R(t) = R(t-1) + h.k.I(t-1)$$

$$tempo(t) = tempo(t-1) + h$$

Em que:

- **h**: pequeno intervalo de tempo (em horas);
- **b**: facilidade de contágio de um indivíduo;
- **k**: probabilidade que um indivíduo se recupere;
- **tempo**: instantes de tempo nos quais o modelo é simulado (em horas).

Para estimar o parâmetro **b**, pode-se observar, durante um intervalo de tempo **T**, quantos indivíduos **N** se infectaram, considerando-se o número de pessoas suscetíveis e o número de pessoas infectadas no início da observação. A equação abaixo resume o cálculo:

$$b = \frac{N}{TS_0I_0}$$

Em que:

- **N**: número de pessoas suscetíveis que se infectaram em um intervalo de tempo **T**
- **S<sub>0</sub>**: número de pessoas suscetíveis no início da observação
- **I<sub>0</sub>**: número de pessoas infectadas no início da observação

Para se estimar o parâmetro **k**, pode-se observar, durante um intervalo de tempo **T'**, quantos indivíduos **m** se recuperaram de um total de **n** indivíduos. A equação abaixo resume o cálculo:

$$k = \frac{m}{nT'}$$

## Tarefa

Uma escola planeja organizar uma viagem de férias para seus alunos em um acampamento isolado. Com medo de uma epidemia viral, o diretor gostaria de traçar alguns cenários e contratou você para simulá-los. Utilizando o modelo SIR descrito acima, faça um programa que atenda às seguintes especificações:

Implemente o modelo SIR, isto é, encontre os valores de  $S(t)$ ,  $I(t)$  e  $R(t)$  ao longo de um determinado período de simulação. Para isso:

1. Leia os parâmetros da simulação a partir de um arquivo de texto:  $S(0)$ ,  $I(0)$ ,  $R(0)$ ,  $h$ ,  $N$ ,  $T$ ,  $S_0$ ,  $I_0$ ,  $m$ ,  $n$ ,  $T'$  e o tempo de simulação em dias.
2. Crie um arquivo de texto no formato csv contendo quatro colunas, cada coluna com os valores de  $S(t)$ ,  $I(t)$ ,  $R(t)$  e  $tempo(t)$ , respectivamente.
3. Construa os gráficos da evolução de cada uma das três populações simuladas.
4. Considerando que 2% dos indivíduos da população R não se recuperam e falecem em uma epidemia desse tipo, plote também a curva da estimativa de mortes relacionada à doença.
5. A partir do cenário definido pelos parâmetros de entrada do simulador (Cenário 0), rode também dois outros cenários com diferentes estratégias de contenção:
  - a. Cenário 1: Distanciamento/uso de máscaras  
Esta estratégia refere-se à facilidade/dificuldade de contágio na população, influenciando assim a variável  $b$  da equação. Para este cenário, suponha que o tempo de contaminação agora passa a ser maior (maior valor de  $T$ , embora os mais continuem constantes), como esta ação só ocorre depois da percepção do contágio, suponha que este cenário só começa a partir de um determinado tempo.
  - b. Cenário 2: Melhoria nos protocolos de atendimento  
Esta estratégia refere-se à capacidade de recuperação de pessoas já infectadas, influenciando assim a variável  $k$  da equação. Podemos simular esta modificação simplesmente reduzindo o tempo que leva para o paciente melhorar (menor valor de  $T'$ ). Da mesma forma que o cenário 1 este cenário só passa a ser válido depois de um determinado tempo.
  - c. Para implementar cada cenário você precisa modificar os parâmetros de entrada do programa para que, além dos dados já fornecidos, sejam fornecidos os novos valores para o cenário 1 e para o cenário 2.
6. Trace gráficos da evolução de  $S(t)$ ,  $I(t)$  e  $R(t)$  ao longo do tempo nos 3 cenários apresentados.

Para facilitar a verificação dos resultados de seu programa, apresentamos o gráfico da evolução de  $S(t)$ ,  $I(t)$  e  $R(t)$  a partir do seguinte cenário:

- O número de alunos participantes do acampamento é 70. Destes, dois alunos estão infectados com o vírus e nenhum deles ainda se recuperou ou tornou-se imune.
- A partir do início da simulação ( $tempo(0)=0$ ), a discretização do tempo é de 6 minutos. Ou seja, a cada 6 minutos novos valores de  $S(t)$ ,  $I(t)$  e  $R(t)$  são calculados. Isso equivale a dizer que  $h=0,1$  (ou seja, 1/10 de 1h).
- Em uma excursão no passado, observou-se que em um grupo de 60 alunos, dentre os quais havia 10 infectados, apenas 38 alunos não estavam infectados após 24 horas de

contato. Esses alunos foram imediatamente isolados. Entre os infectados, 6 se recuperaram nas 24h seguintes.

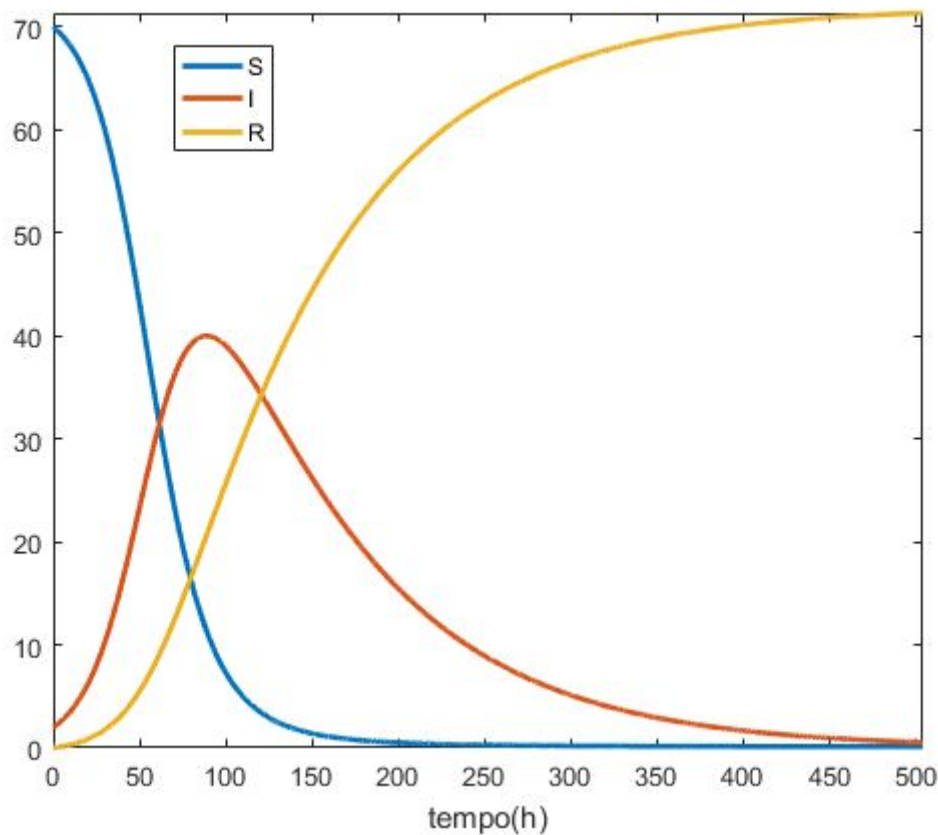


Figura 1 - Simulação de  $S(t)$ ,  $I(t)$  e  $R(t)$  a partir do cenário apresentado após 500 horas.

## Avaliação

O trabalho será avaliado sob os seguintes critérios: (i) utilização correta e abrangente dos conteúdos vistos nas aulas; (ii) a corretude da execução do programa implementado, que deve apresentar saída em conformidade com a especificação e as entradas de dados fornecidas, e (iii) a aplicação correta de boas práticas de programação, incluindo legibilidade, organização e documentação de código fonte. A presença de mensagens de aviso (*warnings*) ou de erros de compilação e/ou de execução, a modularização inapropriada e a ausência de documentação são faltas que serão penalizadas.

## Checkpoints

A avaliação do projeto ocorrerá em duas etapas (checkpoints), descritas a seguir:

### Checkpoint 1

- **Pontuação:** 3,5
- **Objetivos:**
  - a. Implementação do modelo SIR e saída no terminal;

- b. Leitura dos parâmetros iniciais de simulação à partir de um arquivo de texto (formato do arquivo definido pela equipe);
- c. Implementação do modelo SIR com saída em arquivo de texto csv;
- d. Identificar e elencar modificações no código que serão cobradas no checkpoint 2. Exemplos:
  - Modularização
  - Modificação de parâmetros de funções (valor para referência)
  - Uso de alocação dinâmica
- **Entregáveis:**
  - a. Implementação do modelo com entrada e saída no terminal (50% do checkpoint);
  - b. Formato de dados do arquivo txt de configuração e leitura do arquivo de configuração (+25% do checkpoint);
  - c. Saída de dados no arquivo de csv (+25% do checkpoint).

## Checkpoint 2

- **Pontuação:** 3,5
- **Objetivos:**
  - a. Plotagem dos gráficos de SIR após saída (para isso, é necessário usar uma ferramenta extra que deve ser definida à escolha da equipe);
  - b. Implementação da simulação dos dois cenários de contenção (cenários 1 e 2);
  - c. Implementação das modificações sugeridas no Checkpoint 1.
- **Entregáveis:**
  - a. Implementação dos cenários de simulação e formato de dados do arquivo txt de configuração e leitura do arquivo de configuração (40%)
  - b. Plotagem dos gráficos dos três cenários em função do arquivo de entrada (20%)
  - c. Entrega das modificações sugeridas no CP1 (40%)

Além dos 2 checkpoints, o código fonte entregue ao final da apresentação será avaliado segundo os seguintes parâmetros:

- **Pontuação:** 3,0

### (A) Readme

- Documento contendo uma descrição concisa do trabalho mostrando quais itens foram entregues em cada checkpoint.
- Se compilado no Windows/Linux, o documento deve conter uma seção explicando como compilar e executar o programa **usando um terminal!**
- O documento deve conter uma seção explicando como escrever um arquivo de configuração de entrada, contendo um exemplo concreto de como esse arquivo deve ser.
- O documento deve conter uma seção explicando o passo à passo de como fazer para plotar os dados gerados pelo simulador

### (B) Legibilidade

- A legibilidade do código será avaliada com base no uso de indentação adequada e nomes de funções e variáveis que façam sentido

(C) Organização e documentação de código fonte

- A documentação do código será avaliada com base na documentação inserida através de comentário que precede cada função. A documentação pode usar o [formato do doxygen](#), mas fiquem à vontade para escolher qualquer formato que melhor agrade.
- A organização do código será avaliada com base na escolha de divisão de arquivos de código (.h e .c) com base no significado envolvido na escrita de cada .h (o código precisa obrigatoriamente ser dividido em .h e .c, porém as divisões precisam fazer sentido).

(D) Modularização (separação adequada de arquivos e funções)

- A modularização do código fonte será avaliada com base na divisão do programa em funções e da utilização de structs.
- A modularização do código fonte será avaliada ainda com base na divisão em arquivos .h e .c dos códigos.
- Para o projeto, os códigos devem ser **obrigatoriamente** divididos em ao menos 3 arquivos (main.c, um .h e um .c).

(E) Uso dos conteúdos da linguagem abordados durante o semestre

- O código será avaliado com base no uso das estruturas apresentadas, principalmente: Ponteiros, Alocação Dinâmica, Structs e Manipulação de arquivos (leitura e gravação de dados).

## Entrega

Em cada checkpoint, cada aluno do grupo deverá submeter um único arquivo compactado no formato .zip contendo, de forma organizada, todos os códigos fonte resultantes da implementação deste projeto, sem erros de compilação e devidamente testados e documentados através da opção *Tarefas* na Turma Virtual do SIGAA. Seu arquivo compactado deverá incluir também um arquivo texto README contendo:

- a identificação completa de cada aluno do grupo;
- a descrição de como compilar e rodar o programa (README);
- a descrição das limitações (caso existam) do programa e quaisquer dificuldades encontradas.