

Modelagem Epidemiológica SIR e Extensões

Giovani Massayuki Miranda Nagano, Letícia Nunes de Souza Andrade, Lília Helena Gavazza Pessoa, Mateus de Jesus Mendes

Ilum Escola de Ciência, Centro Nacional de Pesquisa em Energia e Materiais (CNPEM), CEP 13083-970, Campinas, São Paulo, Brasil.

Descrição do Projeto

Este projeto tem como objetivo simular a propagação de uma doença infecciosa em uma população, utilizando o modelo epidemiológico SIR, considerando cenários de complexidade crescente. A proposta é fornecer uma ferramenta interativa, escrita em Python, que permita ao usuário inserir diferentes parâmetros epidemiológicos (como taxas de transmissão, recuperação, mortalidade, interação entre populações e renovação populacional) e observar como esses fatores afetam a evolução dos grupos Suscetíveis (S), Infectados (I) e Recuperados (R) ao longo do tempo.

Por meio da implementação numérica do modelo e da visualização dos resultados em gráficos, o projeto busca aprofundar o entendimento sobre como epidemias se comportam e como pequenas mudanças nos parâmetros podem impactar significativamente a disseminação de uma doença.

Uma das principais características deste trabalho é a possibilidade de o usuário escolher entre quatro diferentes modelos epidemiológicos, cada um incorporando novos parâmetros para representar fenômenos mais realistas da dinâmica populacional e da transmissão da doença. Além disso, o código serve como uma base didática, ideal para estudantes e pesquisadores interessados em explorar ou expandir modelos epidemiológicos com maior complexidade.

Modelos disponíveis no projeto:

- SIR: Modelo clássico (Suscetíveis-Infectados-Recuperados).
- SIRD: Inclui mortalidade por causa da doença, com o parâmetro μ (taxa de mortalidade).
- SIRD - Dupla População Interagente: Simula a interação entre duas populações distintas, com os parâmetros k_{ab} e k_{ba} , que controlam a taxa de contato entre os grupos.
- SIRD - Dinâmica Vital: Inclui o efeito de nascimentos e mortes naturais, com o parâmetro η , representando a taxa de renovação populacional.

A interface permite ao usuário escolher o modelo, inserir os parâmetros de forma personalizada e visualizar os resultados por meio de gráficos de fácil interpretação.

Sobre o modelo Epidemiológico SIR

O modelo SIR é um modelo matemático clássico usado para descrever a propagação de doenças infecciosas em uma população. Ele divide a população em três grupos:

- S (Suscetíveis): indivíduos que podem ser infectados;
- I (Infectados): indivíduos que estão atualmente infectados e podem transmitir a doença;
- R (Recuperados): indivíduos que se recuperaram da doença e ganharam imunidade.

A dinâmica entre esses grupos é representada por um sistema de equações diferenciais que descreve como os indivíduos passam de um grupo para outro ao longo do tempo. As principais taxas envolvidas são:

- β (beta): taxa de transmissão da doença;
- γ (gama): taxa de recuperação.

O modelo parte de algumas suposições simples:

- A população é fixa (não há nascimentos ou mortes naturais no modelo básico);
- A doença confere imunidade após a recuperação;
- A transmissão ocorre por contato entre suscetíveis e infectados.

Apesar de sua simplicidade, o modelo SIR é extremamente poderoso para entender o comportamento inicial de surtos e epidemias, como o crescimento de casos, o pico da infecção e o número total de pessoas afetadas.

Objetivos

O principal objetivo deste projeto é desenvolver uma ferramenta interativa e didática para simular diferentes cenários de propagação de doenças infecciosas em populações humanas, utilizando o modelo epidemiológico SIR e suas extensões.

Ao longo do desenvolvimento, buscamos atingir os seguintes objetivos específicos:

- Compreender a dinâmica de transmissão de doenças infecciosas por meio da implementação numérica de diferentes modelos epidemiológicos.
- Permitir a comparação entre diferentes cenários epidemiológicos, variando parâmetros como taxa de transmissão, recuperação, mortalidade, interação entre populações e dinâmica vital da população.
- Fornecer uma interface simples e intuitiva, que permita ao usuário escolher entre os quatro modelos implementados e inserir os parâmetros de forma personalizada.
- Promover a visualização gráfica dos resultados, facilitando a análise do comportamento dos grupos populacionais ao longo do tempo.
- Criar uma base de código modular e extensível, que possa ser utilizada por outros estudantes, pesquisadores ou interessados que desejem explorar novas variações ou incluir outros fenômenos epidemiológicos (como vacinação ou reinfecção).

- Incentivar o aprendizado prático de conceitos relacionados à epidemiologia matemática, ciência de dados e programação científica, integrando teoria e prática.

Tecnologias e Bibliotecas Utilizadas

O projeto foi desenvolvido utilizando a linguagem Python, por sua versatilidade e ampla aplicação em projetos de ciência de dados, modelagem matemática e análise de dados científicos.

Streamlit: Framework que permite transformar scripts Python em aplicações web interativas, facilitando a execução do projeto diretamente no navegador com uma interface amigável.

NumPy: Para realizar cálculos numéricos e manipulação de arrays de forma eficiente.

Matplotlib: Responsável pela geração dos gráficos que representam a evolução das populações ao longo do tempo.

SciPy: Utilizada para resolver as equações diferenciais ordinárias (EDOs) que descrevem a dinâmica dos modelos epidemiológicos.

Como Funciona a Simulação

A simulação é baseada na resolução numérica de equações diferenciais que descrevem a dinâmica de transmissão da doença, de acordo com o modelo epidemiológico escolhido pelo usuário.

O processo funciona da seguinte forma:

1. Escolha do modelo:

O usuário seleciona um dos quatro modelos disponíveis (SIR, SIRD, SIRD - Dupla População Interagente ou SIRD - Dinâmica Vital).

2. Definição dos parâmetros:

Em seguida, o usuário insere os valores para os parâmetros específicos de cada modelo, como taxa de transmissão, taxa de recuperação, mortalidade, taxas de interação entre populações ou taxa de renovação populacional.

3. Resolução das equações:

O programa utiliza métodos numéricos (com apoio da biblioteca SciPy) para resolver as equações diferenciais que descrevem o comportamento das populações ao longo do tempo.

4. Visualização dos resultados:

Após o cálculo, os resultados são apresentados de forma gráfica, mostrando a evolução de cada grupo populacional (S, I, R, e quando aplicável, D e demais categorias) ao longo dos dias simulados.

Parâmetros de Entrada

A cada execução, o usuário pode escolher qual modelo epidemiológico deseja simular. Após essa escolha, o programa solicitará os parâmetros específicos necessários para o modelo selecionado.

Parâmetros gerais (Comuns a todos os modelos):

Parâmetro	Descrição
Número de dias	Duração da simulação (tempo total em dias).
População inicial (S, I, R)	Quantidade inicial de indivíduos nos grupos Suscetíveis, Infectados e Recuperados.
β (beta)	Taxa de transmissão da doença.
γ (gama)	Taxa de recuperação dos infectados.

Parâmetros adicionais por modelo:

SIRD (Com mortalidade):

Parâmetro	Descrição
μ (mi)	Taxa de mortalidade causada pela doença (fração de infectados que morre por unidade de tempo).

SIRD - Dupla população Interagente:

Além dos parâmetros anteriores, o usuário também deve informar:

Parâmetro	Descrição
População inicial do segundo grupo (S_2 , I_2 , R_2)	Quantidade inicial de indivíduos na segunda população.
k_{AB}	Taxa de contato da população A com a população B.
k_{BA}	Taxa de contato da população B com a população A.

SIRD - Dinâmica Vital (com nascimentos e mortes naturais):

Parâmetro	Descrição
δ (delta)	Taxa de renovação populacional (taxa de nascimento e morte natural).

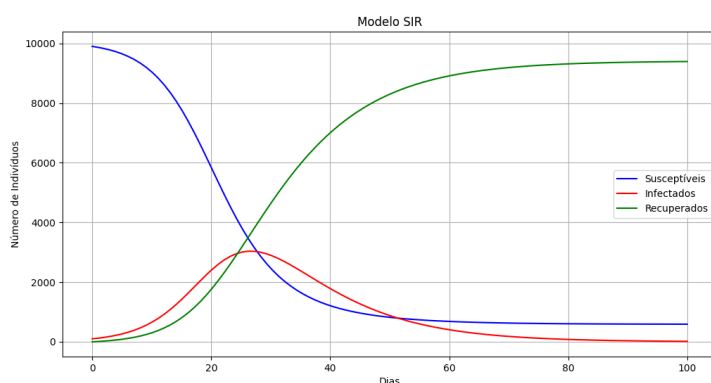
Exemplos de Saída (Gráficos)

Após a execução da simulação, o programa gera gráficos que mostram a evolução dos diferentes grupos populacionais ao longo do tempo, de acordo com o modelo e os parâmetros escolhidos pelo usuário.

Abaixo, apresentamos alguns exemplos de saída gráfica gerados pelos modelos implementados:

Modelo SIR (clássico)

O gráfico mostra o número de suscetíveis (S) diminuindo com o tempo, enquanto os infectados (I) aumentam inicialmente e depois diminuem, até que quase toda a população passe para o grupo de recuperados (R).



Parâmetros:

População total: 10.000

Infectados iniciais: 100

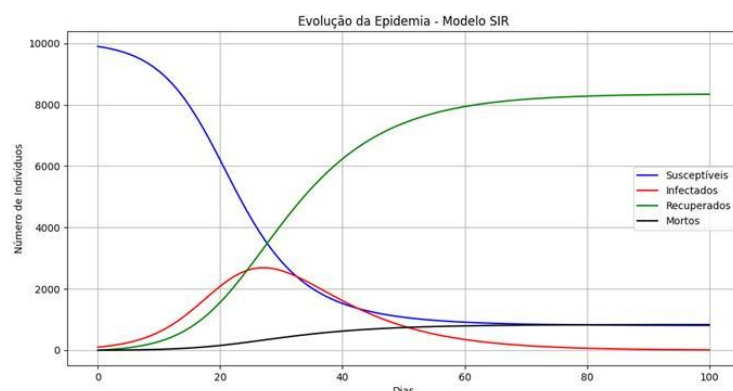
Taxa de transmissão: 0,30

Taxa de recuperação: 0,10

Período: 100

Modelo SIRD (com mortalidade)

Além das curvas S, I e R, é apresentada a curva D (óbitos), que cresce ao longo do tempo de forma proporcional à gravidade do surto.



Parâmetros:

População total: 10.000

Infectados iniciais: 100

Taxa de transmissão: 0,30

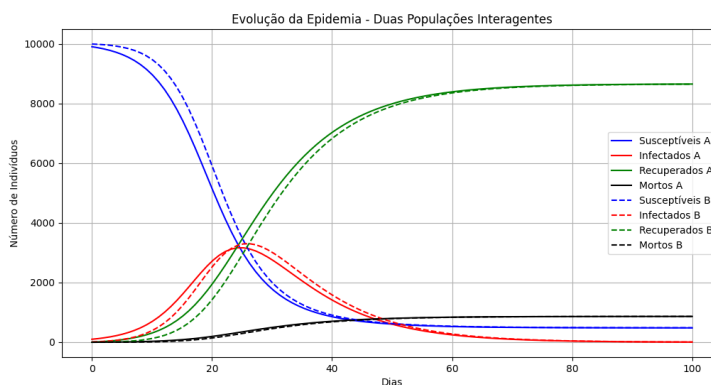
Taxa de recuperação: 0,10

Taxa de mortalidade: 0,01

Período: 100

Modelo SIRD – Dupla População Interagente

O gráfico apresenta as curvas S, I, R (e D, se aplicável) para duas populações distintas, permitindo visualizar como o contato entre grupos pode acelerar ou retardar a disseminação da doença.

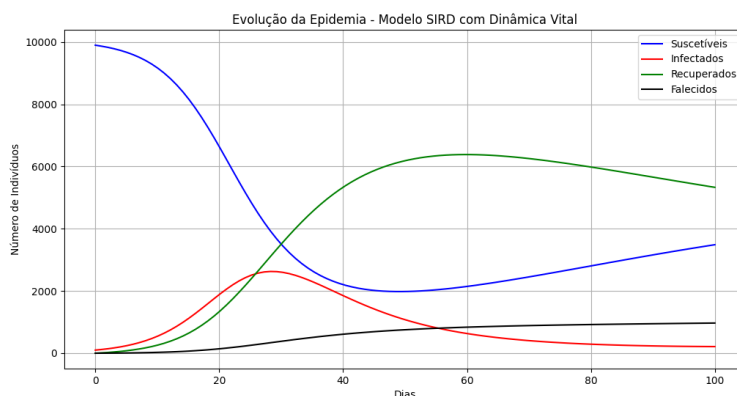


Parâmetros:

População	A	População	B	Interação e Duração
População total:	10.000	População total:	10.000	Fator de transmissão A ->
Infectados iniciais:	100	Infectados iniciais:	100	B: 0,05
Taxa de transmissão:	0,30	Taxa de transmissão:	0,30	Fator de transmissão B ->
Taxa de recuperação:	0,10	Taxa de recuperação:	0,10	A: 0,05
Taxa de mortalidade:	0,01	Taxa de mortalidade:	0,01	Período: 100

Modelo SIRD – Dinâmica Vital

Inclui o efeito de nascimentos e mortes naturais na evolução da população. Isso pode levar à estabilização das curvas ao longo do tempo, simulando situações de equilíbrio epidemiológico.



Parâmetros:

População total:	10.000
Infectados iniciais:	100
Taxa de transmissão:	0,30
Taxa de recuperação:	0,10
Taxa de natalidade/mortalidade natural:	0,01
Taxa de mortalidade:	0,01
Período:	100

Como executar o Projeto

Para acessar projeto final diretamente pela internet, acesse: [Modelo SIR](#).

Para executar o projeto no Visual Studio Code, siga os seguintes passos:

1. Baixe os arquivos do Modelo-SIR disponíveis no repositório do [GitHub](#) (se desejar apenas os arquivos essenciais, baixe SIR.py, SIRD.py, SIRD_duplo.py, SIRD_vital.py, main.py e mantenha-os no mesmo diretório);
2. Abra o Visual Studio Code;
3. No canto superior esquerdo, clique em Arquivo e depois clique em Abrir pasta;

4. Selecione a pasta onde estão os arquivos do Modelo-SIR;
5. Na esquerda, deve haver vários arquivos de extensão .py; selecione o **main.py**;
6. Com a aba main.py aberta, clique Ctrl + ' (a tecla abaixo do Esc);
7. Digite **streamlit run main.py** ou **python -m streamlit run main.py**;
8. Após isso, uma página deve abrir no seu navegador com o programa rodando.

Dificuldades encontradas

Durante o desenvolvimento do projeto, enfrentamos alguns desafios que limitaram a implementação de certas ideias inicialmente previstas. Um dos principais foi a tentativa de criar uma animação da evolução da doença que varia ao longo do período estabelecido. Apesar de termos desenvolvido parte da lógica (imagem 1), a implementação visual se mostrou complexa dentro do tempo disponível.

Também criamos uma versão do código que simulava a propagação da doença entre múltiplas cidades, considerando o fluxo de pessoas entre elas. Essa abordagem incorpora mobilidade populacional como fator de disseminação (imagem 2), mas acabou sendo deixada de lado por conta da dificuldade de integração com a interface existente.

```

# ANIMAÇÃO COM BOTÕES
st.subheader("Animação da evolução da epidemia")
frame_placeholder = st.empty() # Espaço para o gráfico animado

# Inicializa o estado da animação
if "animar" not in st.session_state:
    st.session_state.animar = False

# Botão para iniciar a animação
if st.button("Iniciar Animação"):
    st.session_state.animar = True

# Botão para parar
if st.button("Parar Animação"):
    st.session_state.animar = False

# Se o estado da animação estiver ativado, roda a animação
if st.session_state.animar:
    st.subheader("Animação da evolução da epidemia")
    frame_placeholder = st.empty()

    for dia in range(1, len(t), 2):
        fig, ax = plt.subplots(figsize=(10, 6))
        ax.plot(t[dia], S[dia], 'b', label='Suscetíveis')
        ax.plot(t[dia], I[dia], 'r', label='Infetados')
        ax.plot(t[dia], R[dia], 'g', label='Recuperados')
        ax.plot(t[dia], D[dia], 'y', label='Falecidos')

        ax.set_title(f'Evolução da Epidemia até o Dia {int(t[dia])}')
        ax.set_xlabel('Dias')
        ax.set_ylabel('Número de Indivíduos')
        ax.legend()
        ax.grid(True)

        frame_placeholder.pyplot(fig)
        time.sleep(0.05)

```

Figura 1 – Trecho de código que realiza a animação.

```

Quantas cidades participarão da simulação? 3

Nome da cidade 1: São Paulo
Parâmetros para São Paulo:
Número de suscetíveis: 11000000
Número de infectados: 50000
Número de recuperados: 10000

Nome da cidade 2: Rio de Janeiro
Parâmetros para Rio de Janeiro:
Número de suscetíveis: 6500000
Número de infectados: 30000
Número de recuperados: 5000

Nome da cidade 3: Recife
Parâmetros para Recife:
Número de suscetíveis: 1540000
Número de infectados: 8000
Número de recuperados: 2000

Insira o valor de beta (taxa de transmissão, entre 0 e 1): 0.25
Insira o valor de gamma (taxa de recuperação, entre 0 e 1): 0.1

Agora informe quantas pessoas infectadas viajam diariamente de uma cidade para outra:
São Paulo -> Rio de Janeiro: 500
São Paulo -> Recife: 100
Rio de Janeiro -> São Paulo: 450
Rio de Janeiro -> Recife: 80
Recife -> São Paulo: 120
Recife -> Rio de Janeiro: 90
Quantos dias deseja simular? 30

```

Figura 2 – Resultados do código não implementado de n-cidades.

Essas tentativas, mesmo não implementadas na versão final, contribuíram para nosso aprendizado e abrem caminhos para futuras melhorias do projeto.

Conclusões

O desenvolvimento deste projeto proporcionou uma oportunidade prática de aplicar conceitos de modelagem matemática, ciência de dados e programação científica, além de aprofundar o entendimento sobre a dinâmica de propagação de doenças infecciosas.

Ao implementar diferentes versões do modelo epidemiológico SIR, com níveis crescentes de complexidade, conseguimos observar na prática como pequenas variações nos parâmetros podem gerar grandes impactos na evolução de uma epidemia. A inclusão de aspectos como mortalidade, interação entre populações e dinâmica vital trouxe ao

projeto um caráter mais próximo de situações reais, permitindo uma análise mais rica e detalhada.

Além disso, a criação de uma interface interativa em Streamlit tornou a experiência do usuário mais acessível e intuitiva, permitindo que pessoas com diferentes níveis de familiaridade com programação pudessem explorar os modelos.

Possíveis extensões futuras

Durante o desenvolvimento deste projeto, surgiram diversas ideias de como o trabalho pode ser expandido e aprimorado em versões futuras. Algumas possibilidades incluem:

Inclusão de novos modelos epidemiológicos:

Adicionar variações mais complexas, como o modelo SEIR (Suscetível-Exposto-Infectado-Recuperado), SIRS (com reinfeção) ou modelos com vacinação.

Análise com dados reais:

Permitir a importação de dados epidemiológicos de surtos reais, possibilitando a comparação entre os resultados da simulação e os dados observados.

Simulação geoespacial:

Adicionar recursos que considerem distribuição geográfica das populações, possibilitando a análise de propagação em diferentes regiões ou cidades.

Contribuidores

Giovani Massayuki Miranda Nagano: Responsável pelo código de resolução das equações diferenciais e pela organização das contribuições na função do modelo SIR Base. Também auxiliei no desenvolvimento inicial da interface gráfica do projeto.

Letícia Nunes de Souza Andrade: Contribuí com a implementação da parte do código relacionada aos parâmetros de entrada. Também desenvolvi parte de uma versão de código que considerava o impacto de viajantes infectados entre cidades (não utilizada na versão final). Além disso, fui responsável pela elaboração do README e pela redação do relatório final.

Lília Helena Gavazza Pessoa: Contribuí com a implementação da parte do código relacionada à parte matemática do modelo, definindo as equações base. Também desenvolvi a parte matemática do código que não foi utilizado na versão final. Ademais, fui responsável pela elaboração dos slides da apresentação.

Mateus de Jesus Mendes: Contribuí com a integração entre as funcionalidades matemáticas do código, a obtenção dos parâmetros numéricos e a exibição dos resultados de maneira dinâmica, por meio do desenvolvimento da interface gráfica da aplicação.

Orientadores

Daniel Roberto Cassar

Doutorado: Ciência e Engenharia de Materiais (UFSCar) – Pós-doutorado: UFSCar. Área de atuação: Informática de materiais.

James Moraes de Almeida

Doutorado: Nanociências e Materiais Avançados (UFABC). Pós-Doutorado: UFABC, Ecole Polytechnique Fédérale de Lausanne, USP. Área de atuação: Computação de alto desempenho.

Leandro Nascimento Lemos

Doutorado: Bioinformática e Ecologia de Microrganismos (USP). Pós-Doutorado: Laboratório Nacional de Computação Científica (LNCC), Unicamp e Universidade de Viena. Área de Atuação: Bioinformática e Ecologia Numérica.

Vinícius Francisco Wasques

Doutorado: Matemática Aplicada (Unicamp). Área de atuação: Equações Diferenciais, Teoria de Conjuntos Fuzzy e Biomatemática.

Referências Bibliográficas

Analytics Vidhya. Mathematical Modelling: Modelling the Spread of Diseases with SIRD Model. Disponível em: <https://www.analyticsvidhya.com/blog/2021/08/mathematical-modelling-modelling-the-spread-of-diseases-with-sird-model/>. Acesso em: junho de 2025.

ARINO, J.; VAN DEN DRIESSCHE, P. A multi-city epidemic model. Mathematical Population Studies, Philadelphia, v. 10, n. 3, p. 175–193, 2003. DOI: 10.1080/08898480390200962.

HETHCOTE, Herbert W. The mathematics of infectious diseases. SIAM Review, Philadelphia, v. 42, n. 4, p. 599–653, 2000. DOI: 10.1137/S0036144500371907.

Scientific Python. Epidemic Model – SIR. Disponível em: https://scientific-python.readthedocs.io/en/latest/notebooks_rst/3_Ordinary_Differential_Equations/02_Examples/Epidemic_model_SIR.html. Acesso em: junho de 2025.

TAVARES, João Nuno. Modelo SIR em epidemiologia. Revista de Ciência Elementar, v. 5, n. 2, 2017. Disponível em:

<https://rce.casadasciencias.org/rceapp/static/docs/artigos/2017-020.pdf>.

Acesso em:

junho de 2025.