



## **Pesquisa sobre Ferramentas de Modelagem e Simulação – Prof. André**

**Guilherme da Silva Scher**

### **SimPy**

#### **a) Descrição geral: finalidade e principais áreas de aplicação.**

SimPy é uma biblioteca em Python para simular sistemas de eventos discretos. Ele ajuda a modelar situações onde eventos acontecem em pontos específicos no tempo, como chegada de clientes em uma lavagem de carros (como a que eu fiz). Ele é super útil para simular esse tipo de processo em áreas como logística e sistemas de filas.

#### **b) Classificação: tipo de sistema que pode modelar (discreto, contínuo, baseado em agentes, multimétodo).**

Simulação de eventos discretos: Isso significa que ele lida com sistemas onde os eventos acontecem de forma pontual no tempo, como no exemplo de clientes chegando na lavagem.

#### **c) Licenciamento: gratuita, paga, versão acadêmica ou open-source.**

Gratuito e open-source, sob a licença MIT.

#### **d) Recursos de Inteligência Artificial: se possui integração ou suporte a IA/ML.**

Não tem suporte direto para IA/ML, mas como é Python, tem como usar bibliotecas de Machine Learning pra adaptar como quiser.

#### **e) Aplicação prática:**

- Exemplo real: Eu pensei como exemplo uma lavagem de carros e simulei uma fila de espera pro serviço de lavagem. Carros chegam a diferentes horários e são atendidos na ordem de chegada, mas o tempo de lavagem pode variar.

- Código de exemplo:

Primeiro tem que instalar o SimPy no Colaboratory (ou na máquina que tu estiver).

```
[2] !pip install simpy
```

```
Collecting simpy
  Downloading simpy-4.1.1-py3-none-any.whl.metadata (6.1 kB)
  Downloading simpy-4.1.1-py3-none-any.whl (27 kB)
Installing collected packages: simpy
Successfully installed simpy-4.1.1
```

Agora já pode rodar o código.

```
import simpy
import random

# processo de lavagem do carro
def lavar_carro(env, nome, fila):
    print(f'{nome} chegou no tempo {env.now:.1f}')

    # só pode lavar um carro por vez
    with fila.request() as pedido:
        yield pedido # espera até que possa usar a vaga de lavagem

    tempo_lavagem = random.randint(5, 15)
    print(f'{nome} começou a ser lavado no tempo {env.now:.1f} (vai levar {tempo_lavagem} min)')
    yield env.timeout(tempo_lavagem) # simula o tempo de lavagem
    print(f'{nome} terminou no tempo {env.now:.1f}')

# função para gerar carros chegando
def chegada_carros(env, fila):
    for i in range(5):
        env.process(lavar_carro(env, f'Carro {i+1}', fila))
        yield env.timeout(5) # novo carro chega a cada 5 min

# criação do ambiente (environment que depois é rodado por env.) e recurso
env = simpy.Environment()
fila = simpy.Resource(env, capacity=1) # apenas uma vaga de lavagem

# inicia o processo de chegada dos carros
env.process(chegada_carros(env, fila))

# roda
env.run()
```

```
Carro 1 chegou no tempo 0.0
Carro 1 começou a ser lavado no tempo 0.0 (vai levar 6 min)
Carro 2 chegou no tempo 5.0
Carro 1 terminou no tempo 6.0
Carro 2 começou a ser lavado no tempo 6.0 (vai levar 5 min)
Carro 3 chegou no tempo 10.0
Carro 2 terminou no tempo 11.0
Carro 3 começou a ser lavado no tempo 11.0 (vai levar 8 min)
Carro 4 chegou no tempo 15.0
Carro 3 terminou no tempo 19.0
Carro 4 começou a ser lavado no tempo 19.0 (vai levar 10 min)
Carro 5 chegou no tempo 20.0
Carro 4 terminou no tempo 29.0
Carro 5 começou a ser lavado no tempo 29.0 (vai levar 12 min)
Carro 5 terminou no tempo 41.0
```

Usei `random.randint` para definir o tempo de lavagem dos carros de forma aleatória entre 5 e 15 minutos.

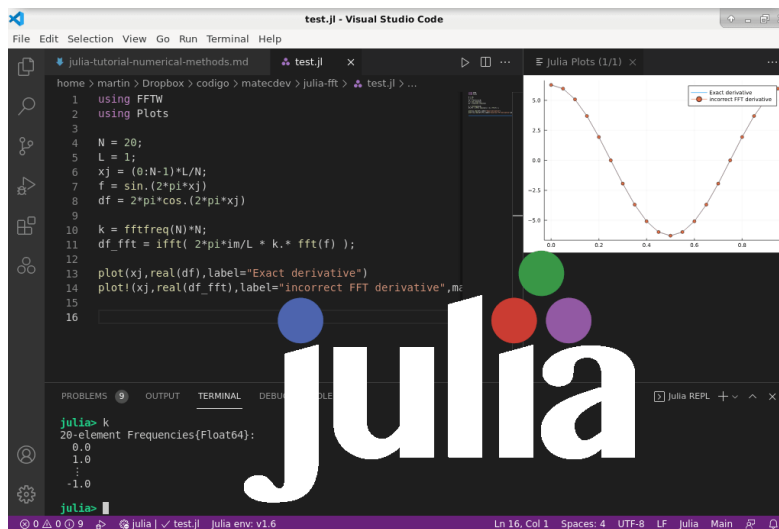
A função `lavar_carro` simula a chegada dos carros e a fila de espera. Quando o posto de lavagem estiver livre, o carro é atendido e o tempo de lavagem é processado. O ambiente `simpy.Environment()` gerencia o tempo e os eventos que acontecem durante a simulação.

Daí o código vai exibir os carros que chegam e o tempo que cada um vai ser lavado, mostrando o processo de fila e atendimento.

## Julia

### a) Descrição geral: finalidade e principais áreas de aplicação.

O Julia é uma linguagem de programação muito rápida e eficiente, feita especialmente para cálculos pesados e simulações científicas. É uma ótima escolha para quem precisa de desempenho, mas também tem uma sintaxe bem simples e próxima das matemáticas que a galera de engenharia e ciência usa.



### b) Classificação: tipo de sistema que pode modelar (discreto, contínuo, baseado em agentes, multimétodo).

Contínuo, discreto, baseado em agentes e multimétodo. Ou seja, depende do que tu precisa. Você pode usar Julia para simular coisas que envolvem tempo contínuo (como o movimento de partículas) ou sistemas com eventos discretos (como uma fila de atendimento).

### c) Licenciamento: gratuita, paga, versão acadêmica ou open-source.

Gratuito e open-source.

**d) Recursos de Inteligência Artificial: se possui integração ou suporte a IA/ML.**

Julia tem suporte para IA e ML. Também tem como usar diversos pacotes para melhorar ainda mais essa integração.

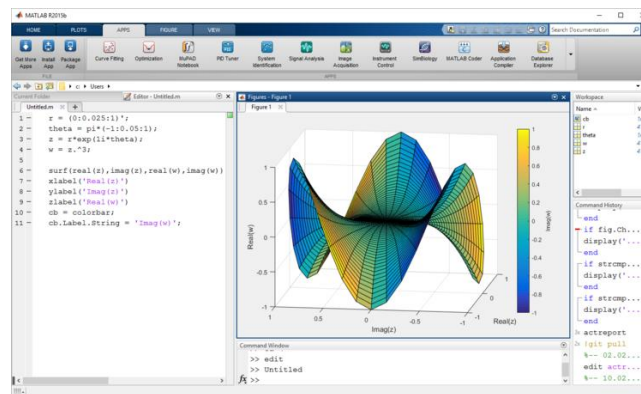
**e) Aplicação prática:**

- Exemplo real: Imagina que você quer estudar o tempo que as pessoas passam esperando na fila do caixa. Com Julia, dá pra modelar quantas pessoas chegam, quanto tempo cada uma leva para ser atendida e quantos caixas estão disponíveis. Isso ajuda a descobrir se é melhor abrir mais caixas ou melhorar o atendimento para reduzir a espera.

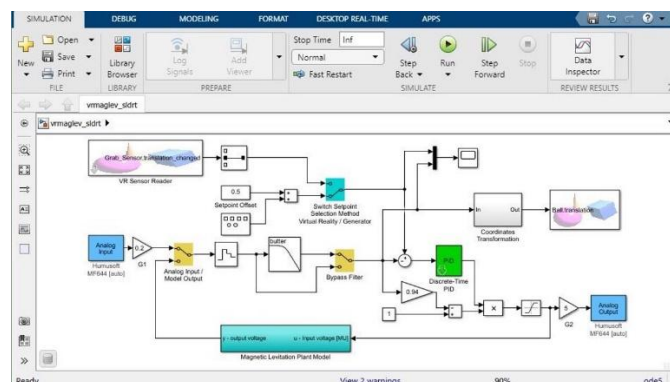
## MATLAB/Simulink

**a) Descrição geral: finalidade e principais áreas de aplicação.**

O MATLAB é uma ferramenta super popular, principalmente em áreas como engenharia e matemática aplicada. Ele é usado para modelar e simular sistemas dinâmicos, fazer análise de dados e resolver equações diferenciais. O Simulink é a versão gráfica, onde você monta os modelos com blocos.



*Imagem do matlab*



*Imagem do matlab com o simulink*

**b) Classificação: tipo de sistema que pode modelar (discreto, contínuo, baseado em agentes, multimétodo).**

Sistemas contínuos e sistemas de controle. O Simulink é focado principalmente em sistemas dinâmicos e de controle, como o controle de temperatura de um forno, ou controle de velocidade de um motor.

**c) Licenciamento: gratuita, paga, versão acadêmica ou open-source.**

Paga, mas existe versão acadêmica com desconto.

**d) Recursos de Inteligência Artificial: se possui integração ou suporte a IA/ML.**

O MATLAB tem integração com IA e ML. Tem ferramentas prontas para usar redes neurais, algoritmos de aprendizado supervisionado e não supervisionado, e até otimização para ajustar os parâmetros dos modelos.

**e) Aplicação prática:**

Exemplo real: No MATLAB com Simulink, é muito comum modelar sistemas que controlam a temperatura, tipo um forno industrial. Tem como criar um modelo que simula como a temperatura varia e ajusta um controlador para manter a temperatura estável mesmo quando a carga ou o ambiente muda.