

# Relatório I - Modelagem e Simulação

**Aluno:** Matheus Henrique Ferreira Protásio

11521BCC020

**Professor:** Carlos Lopes

## Sumário

[Introdução](#)

[Requisitos para rodar o código](#)

[Instruções para rodar o programa](#)

[Detalhes gerais da Implementação](#)

[Considerações Finais](#)

## Introdução

O trabalho tinha como objetivo a implementação de um programa que permitisse a simulação de um sistema, onde deveria ser implementado o modelo M/M/1.

O código foi feito utilizando a linguagem Python e as bibliotecas auxiliares pandas e numpy.

Pandas é uma biblioteca muito utilizada para lidar com dados em Python, com ela é possível montar DataFrames que são estruturas que facilitam a manipulação e visualização dos dados.

Já a biblioteca numpy é uma biblioteca que possui uma grande coleção de funções matemáticas e facilita a manipulação de grandes arrays de dados.

## Requisitos para rodar o código

Para rodar o programa de simulação que utiliza o modelo M/M/1 é necessário ter duas bibliotecas instaladas: `pandas` e `numpy`.

É possível baixar as bibliotecas por meio do comando `pip` com:

```
pip install pandas
```

```
pip install numpy
```

A documentação das bibliotecas usadas pode ser encontrada em:

Pandas: <https://pandas.pydata.org/docs/>

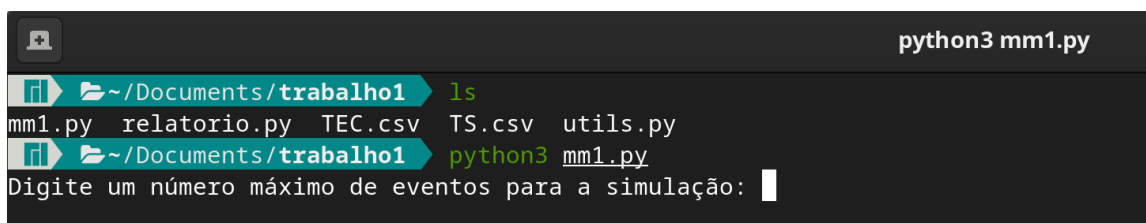
Numpy: <https://numpy.org/doc/stable/>

## Instruções para rodar o programa

Para iniciar o programa no Linux e no macOS é utilizado o comando: `python3 mm1.py` no terminal, dentro da pasta que contém a implementação. Já no caso do Windows deve ser utilizado o comando `python mm1.py`.

Após o início do script, o usuário deve inserir um inteiro que representa quantos eventos a simulação terá, esta foi a maneira escolhida para impor um limite no processo de simulação.

(cada chegada e cada saída correspondem a um evento)

A screenshot of a terminal window with a dark background. The title bar at the top right says "python3 mm1.py". The terminal shows a directory listing for ~/Documents/trabalho1, listing files mm1.py, relatorio.py, TEC.csv, TS.csv, and utils.py. Then, the command "python3 mm1.py" is entered. The prompt "Digite um número máximo de eventos para a simulação:" is displayed, followed by a cursor.

```
python3 mm1.py
~/Documents/trabalho1 ls
mm1.py relatorio.py TEC.csv TS.csv utils.py
~/Documents/trabalho1 python3 mm1.py
Digite um número máximo de eventos para a simulação: |
```

Depois de escolher o número máximo de eventos para a simulação, o usuário será perguntado se o valor do TEC é determinístico e deverá responder com `y` para sim ou `n` caso não seja determinístico. O mesmo procedimento irá acontecer para os valores de TS.

Caso o usuário insira que o TEC não é determinístico, será solicitado o arquivo contendo os dados. O mesmo irá acontecer para TS, como é possível observar nas imagens abaixo.

```
O valor do Tempo entre Chegadas (TEC) é determinístico? [y/n] n
Insira nome do arquivo do TEC: TEC.csv
```

```
O valor do Tempo de Serviço (TS) é determinístico? [y/n] n
Insira nome do arquivo do TS: TS.csv
```

Caso o usuário informe que o TEC ou o TS são determinísticos, será solicitado o valor:

```
O valor do Tempo entre Chegadas (TEC) é determinístico? [y/n] y
Qual o valor do Tempo entre chegadas (TEC): █
```

Por fim, o usuário é questionado se a fila tem limite, caso a resposta seja **y** o usuário deverá informar o limite da fila. Já se a resposta for **n** a simulação e os resultados são mostrados no terminal.

```
A fila tem limite? [y/n] y
Qual o limite da fila? █
```

Abaixo segue um exemplo de saída do programa, para TEC e TS foram utilizados arquivos com dados iguais aos mostrados pelo professor em aula. A fila no exemplo abaixo possui limite de 2 pessoas.

```
matheus@manjaro:~/Documents/trabalho1

[8.275, 12.412500000000001] 0.144330 0.649485 [0.5051546391752578, 0.6494845360824741]
[12.412500000000001, 16.55] 0.164948 0.814433 [0.6494845360824743, 0.8144329896907215]
[16.55, 20.6875] 0.072165 0.886598 [0.8144329896907216, 0.8865979381443297]
[20.6875, 24.825] 0.030928 0.917526 [0.8865979381443299, 0.9175257731958761]
[24.825000000000003, 28.962500000000002] 0.051546 0.969072 [0.9175257731958762, 0.9690721649484535]
[28.962500000000002, 33.1] 0.030928 1.000000 [0.9690721649484536, 0.9999999999999999]

Tabela de simulação:
Evento Cliente TR ES TF HC HS
Início - 0.00000 0 0 0.00000 inf
Chegada 1 0.00000 1 0 14.48125 2.06875
Saída 1 2.06875 0 0 14.48125 inf
Chegada 2 14.48125 1 0 16.55000 37.23750
Chegada 3 16.55000 1 1 18.61875 37.23750
Chegada 4 18.61875 1 2 24.82500 37.23750
Chegada 5 24.82500 1 2 26.89375 37.23750
Saída 5 24.82500 1 2 26.89375 37.23750
Chegada 6 26.89375 1 2 45.51250 37.23750
Saída 6 26.89375 1 2 45.51250 37.23750
Saída 2 37.23750 1 1 45.51250 39.30625
Saída 3 39.30625 1 0 45.51250 45.51250
Saída 4 45.51250 0 0 45.51250 inf

Os clientes [5, 6] foram embora pois a fila estava cheia.

Relatório Final:
Número Médio de entidades na fila: 0.9090909090909092
Taxa Média de Ocupação dos Servidores: 0.7272727272727273
Tempo Médio de uma entidade na fila: 14.998437500000001
Tempo Médio no Sistema: 12.412500000000001
```

Como é possível observar no exemplo acima, são mostrados os clientes que foram embora caso encontrassem filas cheias (para casos onde a fila tem limite).

Em seguida é apresentado um relatório contendo dados referentes à simulação.

## Detalhes gerais da Implementação

O código efetua a simulação para um conjunto de dados TEC e TS não determinísticos utilizando a tabela resultante do Método de Monte Carlo (MMC). Neste caso é feito todo o processo utilizando o MMC para determinar a próxima chegada e o tempo de serviço para cada cliente. Esses dados são aplicados na simulação que segue o modelo M/M/1.

Para os casos de TEC e TS determinísticos, o processo de geração de número pseudo-aleatório seguindo o MMC não é feito, neste caso os valores de TEC e TS são fixos, obedecendo os valores indicados pelo usuário.

Nos casos de fila com limite, por exemplo, uma fila com limite de 3 pessoas, caso um novo cliente chegue para ser atendido e encontre o sistema com a fila cheia (neste caso, 3 pessoas), ele sai imediatamente e é considerado um cliente "desistente", pois quando entrou no sistema encontrou uma fila cheia. Assim, o próximo evento após a

sua chegada, já é a sua saída, pois ele desistiu. Sempre que houverem clientes desistentes eles serão mostrados antes do relatório de estatísticas finais.

## **Considerações Finais**

A implementação de um programa de simulação que segue o modelo M/M/1 foi interessante para entender melhor o processo de simulação como um todo. Com a implementação foi possível aplicar na prática conceitos que foram estudados ao longo do semestre, além de realizar alguns experimentos e diferentes simulações.