

**ATIVIDADE AULA 07 – MODELAGEM E SIMULAÇÃO**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**  
**UNIVERSIDADE FRANCISCANA – UFN. 2025-02.**

PROFESSOR: André F. dos Santos.

**Nome do aluno:** \_\_\_\_\_.

**Data:** \_\_\_\_/\_\_\_\_/\_\_\_\_.

### **OBJETIVO**

Criar um programa Python que analise um sistema de filas real, calculando  $\lambda$  (taxa de chegada),  $\mu$  (taxa de atendimento),  $\rho$  (ocupação) e interpretando os resultados.

### **SUA TAREFA**

Vocês devem criar um programa Python completo que:

- Analise ‘DOIS’ dos serviços listados abaixo
- Colete dados de 30 minutos de observação
- Calcule automaticamente  $\lambda$ ,  $\mu$  e  $\rho$
- Interprete se o sistema está funcionando bem ou mal
- Apresente os resultados de forma clara

**ESCOLHA ‘DOIS’ SERVIÇO PARA ANALISAR (códigos separados, um para cada tipo)**

- RU da faculdade (horário de almoço)
- Caixa do supermercado
- Fila do ônibus (ponto movimentado)
- Atendimento na secretaria acadêmica
- Laboratório de informática

### **DADOS QUE VOCÊS DEVEM INCLUIR**

Seu programa deve trabalhar com:

- **Nome do serviço** (ex: "RU da Faculdade")
- **Horário de observação** (30 minutos - ex: "12h às 12h30")
- **Número de servidores** (quantos atendentes/caixas/computadores)
- **30 dados de chegadas** (pessoas que chegam por minuto)
- **30 tempos de atendimento** (tempo em minutos para cada atendimento)

## CÁLCULOS QUE O PROGRAMA DEVE FAZER

Seu código deve calcular automaticamente:

- $\lambda$  (**lambda**) = Total de chegadas ÷ 30 minutos
- $\mu$  (**mi**) = 1 ÷ Tempo médio de atendimento
- $\rho$  (**rô**) =  $\lambda$  ÷ (Número de servidores ×  $\mu$ )

## INTERPRETAÇÃO AUTOMÁTICA

O programa deve interpretar automaticamente:

- Se  $\rho < 0.5$ : Sistema subutilizado
- Se  $0.5 \leq \rho < 0.8$ : Sistema eficiente
- Se  $0.8 \leq \rho < 1.0$ : Sistema no limite
- Se  $\rho \geq 1.0$ : Sistema em colapso

## EXEMPLO DE SAÍDA ESPERADA

Seu programa deve imprimir algo como:

ANÁLISE DE FILAS - RU da Faculdade

Horário: 12h às 12h30

---

### DADOS COLETADOS:

- Total de chegadas em 30 min: 96 pessoas
- Número de servidores: 3
- Tempo médio de atendimento: 1.95 minutos

### CÁLCULOS:

- $\lambda$  (taxa de chegada): 3.20 pessoas/minuto
- $\mu$  (taxa de atendimento): 0.51 atendimentos/minuto
- $\rho$  (ocupação): 2.09

### INTERPRETAÇÃO:

- Sistema EM COLAPSO - Fila cresce infinitamente!
- Recomendação: URGENTE! Adicione vários servidores

## O QUE ENTREGAR

Arquivo .py (código Python completo e funcionando se possível adicionar gráficos)

Print da execução (screenshot mostrando os resultados)

Análise pessoal (1-2 frases: o serviço funciona bem? Por quê?)

## DICAS IMPORTANTES

- Use dados realistas baseados na sua experiência
- Lembrem-se: import numpy as np para cálculos
- Testem o código antes de entregar
- Se  $\rho > 1$ , o sistema está sobrecarregado

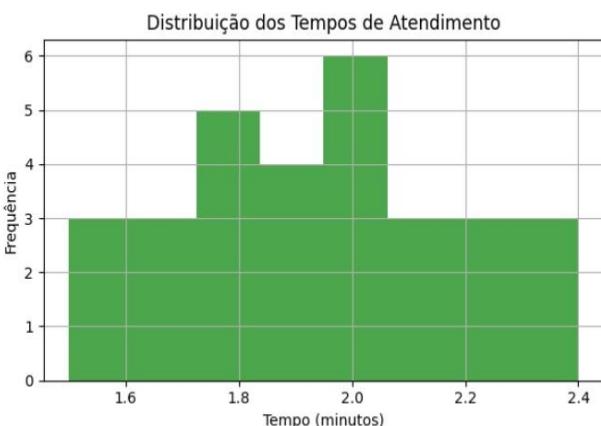
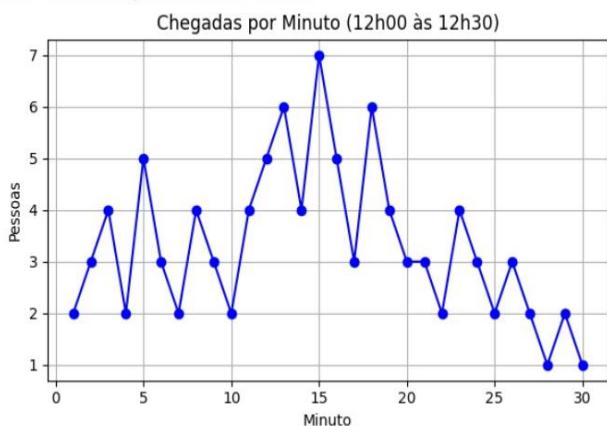
## CRITÉRIOS DE AVALIAÇÃO

- Código Python funcionando corretamente (50%)
- Dados realistas e cálculos corretos (30%)
- Interpretação adequada dos resultados (20%)

## Tipos de gráficos que podem ser utilizados:

ANÁLISE DETALHADA:

- Em 1 hora, chegam aproximadamente: 200 pessoas
- Em 1 hora, conseguimos atender: 93 pessoas
- PROBLEMA: 1.79 pessoas/min se acumulam na fila!
- Em 1 hora: 107 pessoas extras na fila!



ANÁLISE CONCLUÍDA!

**PRAZO: ATÉ O FINAL DA AULA (08-09-2025). Entregas com atrasos terão desconto de nota.**