🏥 Simulação de uma Clínica Médica

Projeto laboratorial

Idealizado e coordenado por: José Carlos Ramalho e Luís Filipe Cunha

Data de execução: 7 de Novembro de 2025 a 11 de Janeiro de 2026

- Instituição: Universidade do Minho / Escola de Engenharia
- Curso: Licenciatura em Engenharia Biomédica (2ºano)
- UC: Algoritmos e Técnicas de Programação

Objetivos

Os alunos deverão construir uma aplicação que simula o atendimento de doentes numa cllínica médica de acordo com o seguinte fluxo:

- Os doentes chegam à clínica de acordo com uma **distribuição de Poisson**;
- Os médicos atendem os doentes e o respetivo tempo de consulta é aleatório mas dentro de certos parâmetros (segue também uma distribuição: exponencial, uniforme ou normal);
- A simulação irá registar tempos de espera, tamanhos das filas de espera, e o tempo de ocupação dos médicos;
- No fim, deverão ser produzidos gráficos que mostrem a evolução dos vários parâmetros ao longo da duração da simulação;
- A aplicação deverá ser usada também para testar a variação dos resultados em função da variação dos parâmetros, exemplos:
    1. O que acontece à fila de espera se a taxa de chegada de doentes aumentar?
    2. O que acontece à ocupação dos médicos se a taxa de chegada de doentes aumentar ou se o tempo de consulta seguir outros parâmetros?
    3. Sê criativo e pensa em mais resultados interessantes que se poderão obter.

Objetivos da aprendizagem

- Modelação de sistemas do mundo real recorrendo a processos estocásticos;
- Implementação de simulações de eventos discretos em Python;
- Compreender as distribuições de Poisson e Exponencial e a sua aplicação prática;
- Armazenamento e análise de dados estatísticos de desempenho;
- Escrita de código estruturado e bem documentado.

Descrição do funcionamento básico do sistema

- Os doentes chegam aleatoriamente à clínica de acordo com uma distribuição de Poisson com parâmetro λ;
- Se houver um médico disponível, o doente é atendido imediatamente;
- Caso contrário, irá esperar pela sua vez (entra numa fila de espera - `queue`);
- A duração de uma consulta segue uma distribuição aleatória (exponencial, normal, or uniforme);

- Terminada a consulta, o doente abandona a clínica.

Parâmetros de configuração

Parâmetro	Descrição	Valor exemplo
`lambda_rate`	Taxa de chegada de doentes (doentes/hora)	10
`num_doctors`	Número de médicos disponíveis	3
`service_distribution`	tipo de distribuição (`exponential`, `normal`, `uniform`)	`"exponential"`
`mean_service_time`	Tempo médio de consulta (minutos)	15
`simulation_time`	Duração da simulação (minutos)	480

Resultados esperados

Indicador	Descrição
Tempo médio de espera	Tempo médio de espera para os doentes
Tempo médio de consulta	
Tempo média na clínica	Quando tempo em média esteve cada doente na clínica
Tamanho da fila de espera (média & máximo)	
Ocupação dos médicos	Percentagem de tempo em que os médicos estão ocupados
Doentes atendidos	Número total de doentes atendidos

Gráficos a gerar

- Evolução do tamanho da fila de espera ao longo do tempo da simulação;
- Evolução da taxa de ocupação dos médicos ao longo do tempo da simulação;
- Gráfico mostrando a relação do tamanho médio da fila de espera com a taxa de chegada de doentes (fazer variar a taxa entre 10 e 30);
- Outros que possas achar interessantes e úteis.

Interface da aplicação

A aplicação deverá ter uma interface gráfica implementada com o módulo `simpleGUI` onde deverá ser possível:

1. Executar uma simulação;
2. Alterar os parâmetros da simulação;
3. Visualizar os gráficos pedidos;
4. (extra) Visualizar graficamente a fila de espera e a ocupação dos consultórios.

Kit inicial

Dada a complexidade do projeto, optou-se por fornecer aos alunos um kit inicial a funcionar. Com este kit podem testar-se os vários parâmetros da simulação e perceber rapidamente como tudo funciona. Há muito espaço para melhorias e alterações.

O objetivo não é os alunos ficarem "agarrados" a este kit, é apenas terem um exemplo a funcionar. Podem alterar tudo o que quiserem e até implementar outras abordagens.

A seguir apresenta-se a função principal, o kit completo encontra-se em ficheiro anexo.

```
def simula():
    tempo_atual = 0.0
    contadorDoentes = 1
    queueEventos = [] # Lista de eventos que vão acontecer, ordenada por
tempo de ocorrência do evento
    queue = [] # Fila de espera - doentes à espera de médico disponível
    # --- Geração da lista de médicos
    medicos = [[f"m{i}", False, None, 0.0, 0.0] for i in
range(NUM_MEDICOS)]
    # ---
    # --- Geração das chegadas de doentes
    chegadas = {} # dicionário de suporte para a geração das consultas
    tempo_atual = tempo_atual + gera_intervalo_tempo_chegada(TAXA_CHEGADA)
    while tempo_atual < TEMPO_SIMULACAO:
        doente_id = "d" + str(contadorDoentes)
        contadorDoentes += 1
        chegadas[doente_id] = tempo_atual
        queueEventos = enqueue(queueEventos, (tempo_atual, CHEGADA,
doente_id))
        tempo_atual = tempo_atual +
gera_intervalo_tempo_chegada(TAXA_CHEGADA)
    # ---
    # ---
    # --- Tratamento dos eventos
    doentes_atendidos = 0

    while queueEventos != []:
        evento, queueEventos = dequeue(queueEventos)
        print(e_tipo(evento), evento)
        tempo_atual = e_tempo(evento)

        if e_tipo(evento) == CHEGADA:
            medico_livre = procuraMedico(medicos)
            if medico_livre:
                medico_livre = mOcupa(medico_livre) # o médico ficou
ocupado
                medico_livre = mInicioConsulta(medico_livre, tempo_atual)
                tempo_consulta = gera_tempo_consulta()
                medico_livre = mDoenteCorrente(medico_livre,
e_doente(evento)) # médico fica a atender o doente que acabou de chegar
                queueEventos = enqueue(queueEventos, (tempo_atual +
```

```
tempo_consulta, SAIDA, e_doente(evento)))
            else:
                queue.append((evento[2], tempo_atual)) # doente fica à
espera
                print(f"Fila de Espera({len(queue)}): ", queue)
        elif evento[1] == SAIDA:
            doentes_atendidos += 1
            # Vamos libertar o médico e despachar o doente
            i = 0
            encontrado = False
            while i < len(medicos) and not encontrado: # vou procurar o
médico que está a atender o doente cuja consulta terminou
                if m_doente_corrente(medicos[i]) == e_doente(evento): # se
encontrei o médico que está a atender o doente deste evento
                    medicos[i] = mOcupa(medicos[i]) # o médico ficou livre
                    medicos[i] = mDoenteCorrente(medicos[i], None)  # não
está a atender nenhum doente
                    medicos[i] = mTempoOcupado(medicos[i],
m_total_tempo_ocupado(medicos[i]) + tempo_atual -
m_inicio_ultima_consulta(medicos[i])) # incremento o tempo da consulta que
terminou
                    encontrado = True
                i = i + 1

            medico = medicos[i-1]

            if queue != []: # se há doentes à espera vou ocupar o médico
que ficou livre...
                ev, queue = dequeue(queue)
                prox_doente, tchegada = ev
                medico = mOcupa(medico)
                medico = mInicioConsulta(medico, tempo_atual)
                medico = mDoenteCorrente(medico, prox_doente)
                tempo_consulta = gera_tempo_consulta()
                queueEventos = enqueue(queueEventos, (tempo_atual +
tempo_consulta, SAIDA, prox_doente))

    print(f"Doentes atendidos: {doentes_atendidos}")
```

Extras

- Adicionar mais estatísticas: por exemplo, métricas por médico;
- Implementar especialidades médicas e vários tipos de doente (quadro clínico, idade, sexo, ...);
- Suportar a configuração de parâmetros na linha de comando ou num ficheiro de configuração em JSON;
- Em lugar de atender doentes que são meramente um número, atender pessoas (encontra-se em anexo um dataset em JSON de pessoas que pode ser usado).

Resultados a entregar

1. Código Python do projeto;

2. `README.md` — relatório técnico com o setup, descrição de parâmetros, e resultados;
3. (Opcional) Relatório técnico mais elaborado em Markdown ou LaTeX com os resultados e gráficos.

Bibliotecas que poderão utilizar

- `numpy`
- `matplotlib`
- `simpleGUI`

Este projeto não é um *projeto académico*, é um projeto de engenharia. Exige amadurecimento de ideias, discussão em grupo, procura de soluções para os problemas apresentados e alguma investigação.

Têm de dedicar-lhe o tempo necessário.

Bom trabalho e boa criatividade

José Carlos Ramalho e Luís Filipe Cunha