

Universidade do Minho
Escola de Engenharia

Universidade do Minho

Licenciatura em Engenharia Biomédica

Simulação e Gestão de Clínica Médica

Relatório do Projeto de Algoritmos e Técnicas de
Programação

Realizado por:

Ana Matilde Fernandes Oliveira (A112014)

Matilde Laurido Arede (A110034)

Sofia Lopes Milhases Ferreira (A111094)

Índice

1	Introdução	1
2	Metodologia e Implementação	2
2.1	Funcionalidades Extra Implementadas	3
2.2	Execução da Simulação e Gestão de Filas de Espera	3
2.3	Funções Auxiliares	4
3	Análise Gráfica e Resultados	5
3.1	Análise Temporal	5
3.2	Análise de Estabilidade	6
3.3	Análise Estatística	7
4	Interface Gráfica e Interação com o Utilizador	9
4.1	Mecanismo de Autenticação	9
4.2	Painel Principal de Simulação	9
4.2.1	Separador Simulação	10
4.2.2	Separador Gráficos	11
4.3	Persistência e Usabilidade	12
4.4	Identidade Visual e Logótipo	13
5	Conclusão	14
	Referências Bibliográficas	15

1 Introdução

O trabalho desenvolvido consiste na criação de uma aplicação em Python seguindo uma Distribuição de Poisson que reproduz o funcionamento de uma clínica médica, considerando a chegada dos doentes ao longo do tempo, a atribuição de consultas consoante a disponibilidade dos médicos e a duração variável dos atendimentos seguindo distribuições estatísticas configuráveis (Exponencial, Normal ou Uniforme). O modelo permite recolher indicadores de desempenho fundamentais, como o tempo que os doentes aguardam por consulta, o grau de utilização dos médicos e o comportamento das filas de espera.

Neste relatório são apresentados o funcionamento do sistema desenvolvido, os principais resultados obtidos através de gráficos e uma análise dos dados recolhidos, permitindo avaliar a eficácia da solução implementada.

2 Metodologia e Implementação

A aplicação foi desenvolvida na linguagem Python, utilizando uma abordagem modular:

- **Motor de Simulação (simulacao.py):** Implementa a lógica de eventos discretos (chegada/saída), gera a fila de espera (FIFO) e atribui médicos livres aos pacientes.
- **Interface Gráfica (app.py):** Desenvolvida com a biblioteca FreeSimpleGUI, permitindo a configuração dinâmica de parâmetros e a visualização dos resultados.
- **Análise de Dados (analise.py):** Utiliza as bibliotecas NumPy e Matplotlib para cálculos estatísticos e geração de gráficos.

2.1 Funcionalidades Extra Implementadas

Para além dos requisitos base, foram implementadas as seguintes funcionalidades avançadas:

- **Configuração via JSON:** Possibilidade de carregar os parâmetros da simulação a partir de um ficheiro `config.json`.
- **Dataset de Pessoas Reais:** Integração de um ficheiro `pessoas.json`, permitindo simular doentes com nome, idade e sintomas específicos, em vez de identificadores genéricos.
- **Especialidades Médicas:** Implementação de um mecanismo de triagem no qual os médicos possuem especialidades (Clínica Geral, Cardiologia e Pediatria) e atendem os doentes de acordo com a sintomatologia e a idade.
- **Estatísticas Individuais:** Registo detalhado do número de doentes atendidos por cada médico ao longo da simulação.

2.2 Execução da Simulação e Gestão de Filas de Espera

O funcionamento principal da aplicação é assegurado pela função `simular_atendimento()`, responsável por coordenar todo o processo de simulação ao longo do tempo. Esta função gera a chegada de doentes, a atribuição de médicos, o atendimento clínico e a recolha de estatísticas relevantes.

Os eventos de chegada, saída e desistência dos doentes são representados por uma estrutura de eventos e processados cronologicamente através de uma fila ordenada. Para esse efeito, são utilizadas as funções `enqueue()` e `dequeue()`, que garantem que os eventos são tratados pela ordem correta de ocorrência temporal. A informação associada a cada evento é acedida através das funções `e_tempo()`, `e_tipo()` e `e_doente()`.

Sempre que ocorre um evento de chegada, o sistema tenta atribuir um médico disponível ao doente, recorrendo à função `procuraMedico()`. Esta função tem em consideração não apenas a disponibilidade dos médicos, mas também a especialidade clínica necessária, permitindo uma gestão mais realista do atendimento. Caso não exista um médico compatível livre, o doente é encaminhado para a fila de espera, sendo esta ordenada de acordo com o nível de prioridade atribuído.

Durante a simulação, são também tratados eventos de saída, que libertam médicos após o término das consultas, bem como eventos de desistência, aplicáveis a doentes de menor prioridade que aguardam demasiado tempo na fila. Este mecanismo contribui para tornar o modelo mais próximo do funcionamento real de uma clínica.

2.3 Funções Auxiliares

Para suportar o funcionamento da simulação, foram implementadas várias funções auxiliares que encapsulam tarefas específicas do sistema. As funções `gera_\intervalo_\tempo_\chegada()` e `gera_tempo_consulta()` são responsáveis pela geração de valores aleatórios associados à chegada de doentes e à duração das consultas, respeitando as distribuições estatísticas configuradas.

A atribuição de características clínicas aos doentes é realizada pelas funções `atribuir_\pulseira()` e `atribuir_especialidade()`, que determinam, respetivamente, o nível de prioridade e a especialidade médica adequada, tendo em conta a idade do paciente. Estas decisões influenciam diretamente a ordem de atendimento e a gestão da fila de espera.

Adicionalmente, são utilizadas funções específicas para a gestão do estado dos médicos, como `m0cupa()`, `mInicioConsulta()` e `mTempoOcupado()`, permitindo atualizar corretamente a ocupação, o tempo de trabalho e o doente em atendimento. Estas informações são fundamentais para o cálculo de métricas como a taxa de ocupação dos médicos e o tempo médio de consulta.

No final da simulação, a função principal devolve um conjunto de estatísticas agrupadas, incluindo tempos médios, tamanho máximo da fila, número de desistências e dados individuais por médico, possibilitando uma análise detalhada do desempenho do sistema.

3 Análise Gráfica e Resultados

O módulo de análise é responsável por transformar os dados brutos da simulação em visualizações que permitem interpretar o comportamento do sistema. Abaixo descrevem-se os cinco gráficos gerados:

3.1 Análise Temporal

- **Evolução da Fila de Espera:** Este gráfico utiliza uma representação em degraus com preenchimento para ilustrar o volume de doentes em espera ao longo do tempo. Permite identificar visualmente os picos de afluência (onde a curva sobe abruptamente) e a capacidade de recuperação da clínica (onde a curva desce). Uma tendência consistentemente ascendente indicaria que a taxa de chegada é superior à capacidade de atendimento (sistema instável).
- **Ocupação Instantânea dos Médicos:** Representa a percentagem de médicos ocupados em cada instante da simulação. Valores constantemente próximos de 100% sugerem um sistema saturado ou em risco de *burnout* dos recursos, enquanto valores baixos podem indicar que existem mais médicos do que o necessário para a procura atual.

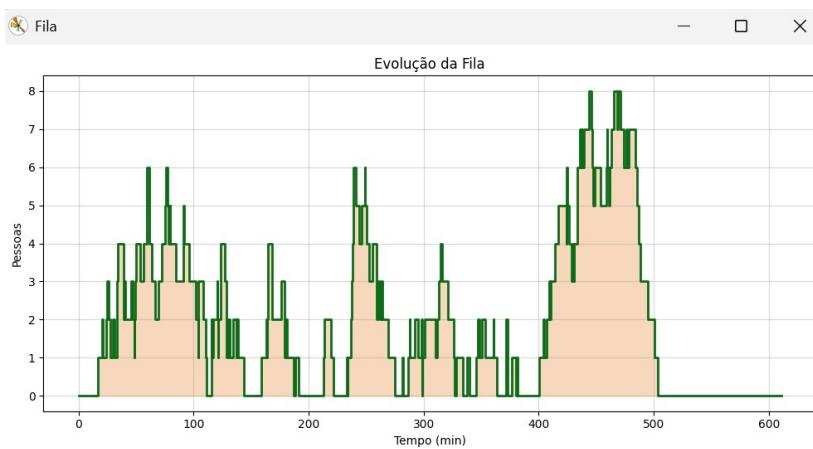


Figura 1: Evolução do tamanho da fila ao longo do tempo.

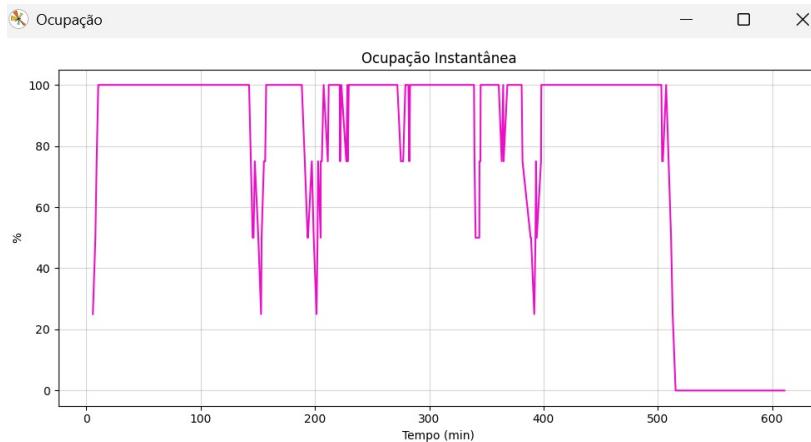


Figura 2: Ocupação instantânea dos Médicos

3.2 Análise de Estabilidade

- **Teste de Sensibilidade:** Relaciona a "Taxa de Chegada"(doentes/hora) com o "Tamanho Médio da Fila". Para gerar este gráfico, a simulação é executada iterativamente com diferentes valores de lambda, permitindo observar o ponto de ruptura do sistema. O crescimento exponencial da curva indica o limiar a partir do qual a clínica deixa de conseguir responder à procura.

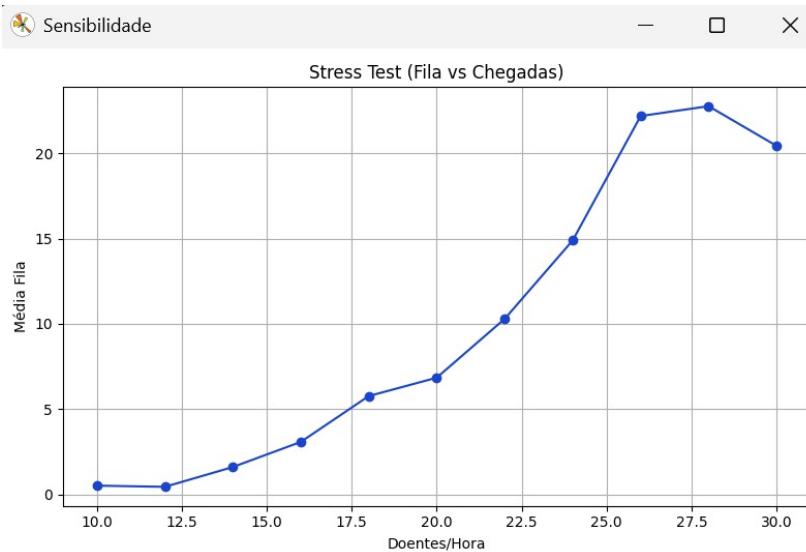


Figura 3: Teste de stress: Variação da fila média em função da taxa de chegada.

3.3 Análise Estatística

- **Histograma do Tempo na Clínica:** Apresenta a distribuição de frequências do tempo total que os doentes passam no sistema (desde a entrada até à saída). Ao contrário de uma média simples, este histograma permite ver a dispersão dos dados, identificando se a maioria dos atendimentos é rápida ou se alguns doentes acabam por ficar retidos demasiado tempo.
- **Distribuição das Desistências:** Este histograma foca-se especificamente nos doentes que abandonaram a fila (prioridade Verde). O gráfico mostra "quanto tempo" estes doentes esperaram antes de desistir. A concentração de barras num determinado intervalo de tempo revela a tolerância média dos pacientes e ajuda a calibrar os parâmetros de desistência do modelo.

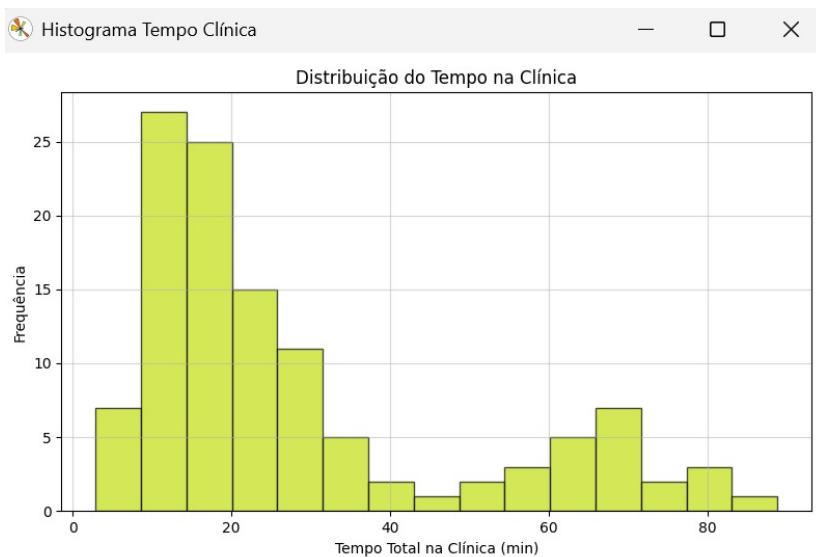


Figura 4: Tempo total na clínica

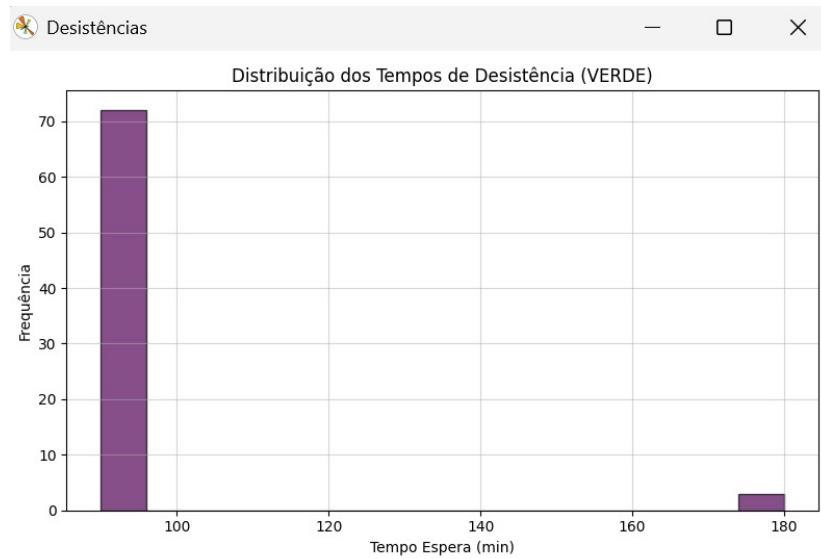


Figura 5: Distribuição dos tempos de desistência

4 Interface Gráfica e Interação com o Utilizador

A interação com o simulador é realizada através de uma interface gráfica desenvolvida com a biblioteca FreeSimpleGUI, estruturada em dois momentos principais.

4.1 Mecanismo de Autenticação

O acesso ao sistema é protegido por uma janela de login (“Entrada no Sistema da Clínica”). O código implementa uma validação de credenciais onde o utilizador deve introduzir o nome e a palavra-passe. O sistema verifica estes dados contra um dicionário de utilizadores registados. Em caso de erro, a interface fornece feedback visual imediato através de uma mensagem de erro a vermelho, sem fechar a aplicação, permitindo novas tentativas. A sessão só é iniciada se a validação for bem-sucedida (autenticacao = True).

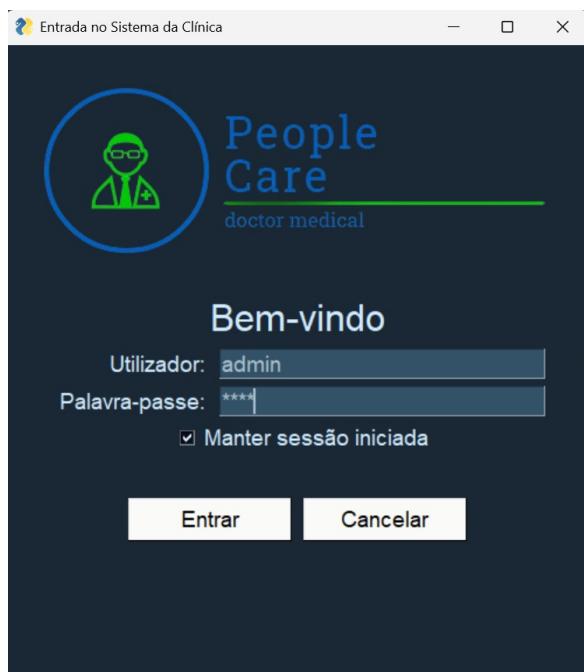


Figura 6: Janela de autenticação do sistema.

4.2 Painel Principal de Simulação

Após o login, o utilizador é direcionado para a janela principal, que está organizada através de um sistema de abas (TabGroup), separando a configuração da visualização gráfica.

4.2.1 Separador Simulação

Este é o painel de controlo central, dividido em duas colunas funcionais:

Configuração (Esquerda): Permite a parametrização dinâmica do modelo antes de cada execução. Utilizam-se componentes interativos específicos para garantir a integridade dos dados:

- **Spinners** para valores numéricos discretos (Taxa de Chegada, Número de Médicos);
- **Input Fields** para variáveis contínuas (Tempo Médio, Duração);
- **ComboBox** para a seleção da distribuição estatística (Exponencial, Normal, Uniforme).

Execução e Visualização de Resultados (Direita): O botão “Executar Simulação” atua como o controlador central da aplicação. Ao ser acionado, desenca-deia uma rotina que valida os parâmetros, carrega a base de dados de pacientes (`pessoas.json`) e invoca o algoritmo de simulação. Após o processamento, os resultados são apresentados de duas formas:

- **Tabela Detalhada:** Apresenta os dados individuais de cada doente (Nome, Prioridade, Tempos), formatados com uma casa decimal para garantir uma leitura limpa.
- **Quadro de Estatísticas:** Um *Frame* dedicado aos KPIs (*Key Performance Indicators*) agregados, como a média de espera, a ocupação dos médicos e a taxa de desistências.

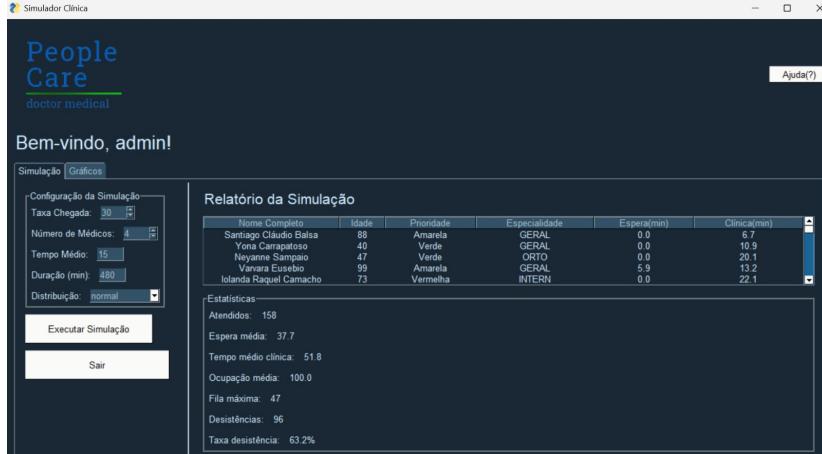


Figura 7: Relatório da Simulação

4.2.2 Separador Gráficos

Para evitar a sobrecarga visual no ecrã principal, a geração de visualizações avançadas foi segregada nesta aba. O utilizador dispõe de botões dedicados que invocam as funções do módulo de análise, gerando janelas flutuantes com os gráficos de:

- Análise de Sensibilidade;
- Evolução da Fila e Ocupação dos Médicos;
- Histogramas de tempos na clínica e desistências.

Do ponto de vista técnico, a implementação destes botões inclui mecanismos de verificação de estado que impedem a geração de gráficos sem dados prévios (evitando erros de execução). Além disso, o processamento foi configurado, em modo não-bloqueante (`block=False`), permitindo ao utilizador manter várias janelas de análise abertas simultaneamente sem congelar a interação com o painel principal, assegurando assim a estabilidade e fluidez do sistema.

É importante salientar que a interatividade destes botões implementa mecanismos de verificação de estado. Caso o utilizador tente gerar gráficos sem antes executar uma simulação, o sistema interceta o evento e exibe um alerta (*popup*), impedindo erros de execução e assegurando a estabilidade do programa.

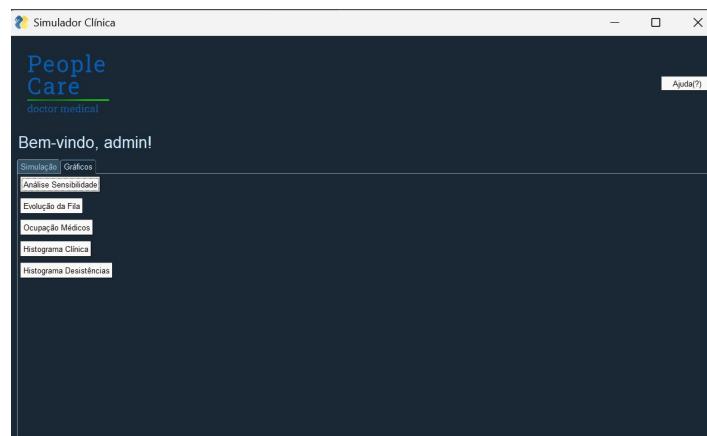


Figura 8: Menu dedicado à geração de gráficos de análise.

4.3 Persistência e Usabilidade

A aplicação integra ainda funcionalidades de suporte à usabilidade, como botões de “Ajuda” (que abrem *popups* com instruções) e janelas de confirmação ao sair. Adicionalmente, o sistema carrega configurações prévias e listas de pacientes a partir de ficheiros JSON (`config.json` e `pessoas.json`), permitindo a persistência de dados entre sessões.

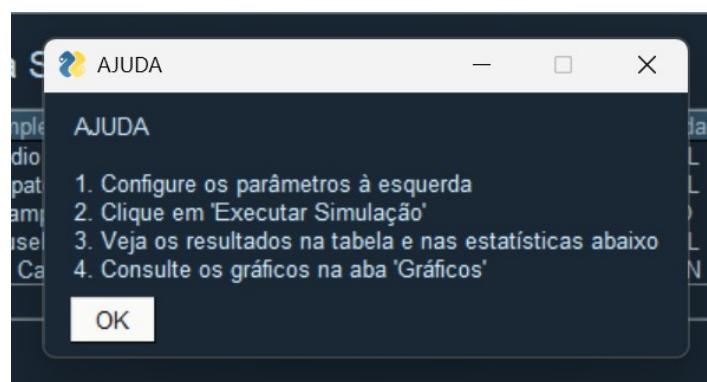


Figura 9: Botão Ajuda

4.4 Identidade Visual e Logótipo

Para conferir uma identidade profissional e coesa à aplicação, foi definida uma identidade visual própria para a clínica simulada. O logótipo desenvolvido atua como o elemento central desta identidade, tendo sido desenhado para transmitir, de forma imediata, o contexto hospitalar e a natureza tecnológica do projeto.



Figura 10: Logotipo

5 Conclusão

Neste projeto, foi desenvolvido um sistema de simulação de eventos discretos aplicado ao contexto de uma clínica médica, integrando um motor lógico robusto com uma interface gráfica interativa baseada em `FreeSimpleGUI`. O sistema cumpriu os objetivos propostos, permitindo não só a gestão operacional (triagem, filas e atendimento), mas também uma análise estatística detalhada do desempenho da clínica.

A análise dos resultados, suportada pelos gráficos gerados (como o teste de sensibilidade e os histogramas), permitiu concluir que o equilíbrio entre a *taxa de chegada* e o *número de médicos* é o fator crítico para a estabilidade do sistema. Observou-se que a implementação de mecanismos de desistência para doentes de menor prioridade (Verdes) aproxima a simulação da realidade, evitando filas infinitas em cenários de saturação.

Referências

- [1] Tutorialspoint. *PySimpleGUI Tutorial*. Disponível em: <https://www.tutorialspoint.com/pysimplegui/index.htm>