



Universidade do Minho  
Licenciatura em Engenharia Biomédica

---

RELATÓRIO FINAL  
Grupo 25

# PROJETO DE ALGORITMOS E TÉCNICAS DE PROGRAMAÇÃO

Francisca Peireira, a112168  
a112168@alunos.uminho.pt

Joana Cunha, a112128  
aa112128@alunos.uminho.pt

Maria Moura, a111681  
a111681@alunos.uminho.pt

Examiner: José Carlos Ramalho  
Universidade do Minho, Braga, Portugal

## Contents

<b>1. Introdução</b>	<b>1</b>
1.1 Descrição Informal do Problema . . . . .	1
<b>2. Estrutura da Base de Dados</b>	<b>2</b>
2.1 Funcionamento do Sistema . . . . .	2
2.2 Funcionalidades Implementadas no Sistema . . . . .	3
<b>3. Interface</b>	<b>5</b>
<b>4. Conclusão</b>	<b>10</b>
<b>5. Webgrafia</b>	<b>10</b>

## 1. Introdução

A partir da necessidade de compreender o funcionamento de uma clínica médica e o impacto da variabilidade no atendimento de doentes, desenvolvemos uma aplicação *ClinicaSimula* de simulação que modela o fluxo de doentes e a ocupação dos médicos ao longo do tempo. O projeto permite simular chegadas aleatórias de doentes, tempos de consulta variáveis e a gestão de filas de espera, possibilitando a análise do desempenho do sistema.

### 1.1 Descrição Informal do Problema

Neste projeto foi desenvolvido um sistema de simulação do atendimento numa unidade clínica, com o objetivo de analisar o funcionamento do serviço de urgência em diferentes cenários. Para a caracterização dos pacientes, é utilizado um ficheiro de dados em formato JSON, que permite armazenar informação de forma simples, estruturada e facilmente reutilizável ao longo da simulação. A aplicação integra um motor de simulação e uma interface gráfica interativa, através da qual o utilizador pode definir parâmetros operacionais, como a taxa de chegada de pacientes, o número de médicos disponíveis e o modelo estatístico aplicado aos tempos de consulta. O sistema gera automaticamente a fila de espera, atribui prioridades clínicas e encaminha os pacientes para médicos compatíveis com a sua especialidade.

Durante a execução da simulação, são recolhidas diversas métricas de desempenho, incluindo tempos médios de espera, ocupação dos médicos e distribuição dos atendimentos por especialidade. Os resultados são apresentados de forma visual através de gráficos e indicadores, permitindo ainda a realização de uma análise de sensibilidade ao número de pacientes que chegam ao sistema.

## 2. Estrutura da Base de Dados

O simulador clínico utiliza estruturas de dados baseadas em dicionários e listas, bem como um ficheiro externo em formato JSON, para representar a informação associada aos pacientes e ao funcionamento do sistema.

Cada paciente é representado por um dicionário, cujos pares chave-valor são:

- **id** (string) – identificador único do paciente no sistema
- **nome** (string) – nome do paciente
- **idade** (int) – idade do paciente
- **chegada** (float) – instante de chegada ao sistema (em minutos)
- **prioridade** (int) – nível de prioridade clínica do paciente, ajustado dinamicamente caso a espera seja longa
- **cor** (string) – cor da pulseira atribuída (Verde, Amarela ou Vermelha)
- **esp** (string) – especialidade médica necessária
- **ordem** (int) – número sequencial de chegada, usado para desempatar pacientes com a mesma prioridade

Os dados dos pacientes podem ser carregados a partir de um ficheiro JSON, sendo utilizados os campos **nome** e **idade** quando disponíveis. Na ausência deste ficheiro, os restantes dados são gerados automaticamente durante a simulação. A prioridade clínica de cada paciente pode aumentar ao longo do tempo de espera, garantindo que ninguém fique demasiado tempo na fila sem ser atendido.

A título de exemplo, apresenta-se um paciente representado na estrutura de dados utilizada:

```
{
  "id": "d1",
  "nome": "João Silva",
  "idade": 45,
  "chegada": 32.5,
  "prioridade": 1, # pode aumentar se o paciente esperar muito
  "cor": "Amarela",
  "esp": "GERAL",
  "ordem": 1
}
```

Para além dos dados dos pacientes, o sistema mantém estruturas internas para representar os médicos, eventos de simulação (chegada e saída), filas de espera e métricas estatísticas, permitindo a análise do desempenho global do serviço clínico.

### 2.1 Funcionamento do Sistema

O sistema funciona através de uma simulação por eventos discretos, considerando a chegada e a saída de pacientes como eventos principais. Os pacientes são encaminhados para médicos disponíveis e compatíveis com a sua especialidade, ou colocados em fila de espera quando não existe atendimento imediato.

A fila é gerida por prioridade clínica e, sempre que um médico fica livre, o próximo paciente elegível é chamado. Ao longo da simulação, são recolhidas métricas de desempenho que permitem avaliar o funcionamento do serviço clínico, sendo os resultados apresentados graficamente na interface da aplicação.

## 2.2 Funcionalidades Implementadas no Sistema

Com o objetivo de facilitar o acesso às diferentes funcionalidades da aplicação, criamos botões na interface gráfica, que permitem ao utilizador interagir com o simulador de forma intuitiva:

**Simular:** Permite executar a simulação com base nos parâmetros definidos pelo utilizador.

**Executar Teste:** Permite realizar uma análise de sensibilidade, avaliando o impacto da taxa de chegada de pacientes no tamanho médio da fila.

**Visualização Gráfica:** Apresenta gráficos de desempenho após a simulação.

**Sair:** Permite encerrar a aplicação de forma segura.

**Carregar:** A aplicação permite o carregamento de uma base de dados de pacientes em formato JSON. Este ficheiro contém informação como o nome e a idade dos pacientes, sendo lido no início da simulação. Caso o ficheiro não exista ou ocorra algum erro de leitura, o sistema gera automaticamente pacientes genéricos, assegurando o correto funcionamento da aplicação.

**Configuração da Simulação:** Antes da execução, o utilizador pode configurar os principais parâmetros do sistema, nomeadamente:

- Taxa de chegada de pacientes (pacientes por hora);
- Número de médicos disponíveis;
- Tipo de distribuição estatística dos tempos de consulta;
- Duração total da simulação.

Estes valores são posteriormente utilizados na simulação.

**Execução da Simulação:** Ao carregarmos no botão de simulação, o sistema inicia um processo de simulação por eventos discretos, considerando eventos de chegada e saída de pacientes. Cada paciente recebe uma prioridade clínica (através da pulseira) e uma especialidade médica, sendo encaminhado para um médico compatível ou colocado em fila de espera.

**Gestão da Fila e dos Médicos:** A fila de espera organiza os pacientes considerando a prioridade clínica (determinada pela cor da pulseira e pela idade) e a ordem de chegada.. O sistema assegura que os médicos apenas atendem pacientes compatíveis com a sua especialidade, respeitando a exclusividade do atendimento pediátrico a menores de idade. Sempre que um médico termina uma consulta, o próximo paciente elegível é automaticamente chamado.

**Consulta de Resultados:** Após iniciarmos a simulação, são apresentados alguns indicadores de desempenho, tais como:

- Número total de pacientes atendidos;
- Tempo médio de espera;
- Tempo médio total no sistema;
- Taxa média de ocupação dos médicos;
- Tamanho máximo da fila de espera.

Estes valores são exibidos diretamente na interface gráfica.

**Visualização Gráfica:** O sistema gera gráficos que permitem analisar visualmente o comportamento da clínica, tais como:

- **Evolução da Fila e Ocupação:** Criada pela função `mostrar_dashboard`, apresenta a evolução da fila de espera e a ocupação dos médicos ao longo do tempo.
- **Distribuição por Especialidade:** Criada pela função `mostrar_grafico_pizza`, mostra a percentagem de pacientes atendidos por especialidade.

- **Análise de Sensibilidade:** A função `mostrar_grafico_sensibilidade` apresenta o impacto da variação da taxa de chegada na média da fila.

**Análise de Sensibilidade:** O botão **Executar Teste** permite calcular o efeito da alteração da taxa de chegada de pacientes sobre a fila média, recorrendo à função `realizar_analise_sensibilidade`. Os resultados são apresentados num gráfico interativo que facilita a interpretação do desempenho do sistema sob diferentes cargas de trabalho.

**Registo de Atividades:** Durante a execução da simulação, todas as ações relevantes são registadas num ficheiro de texto, permitindo ao utilizador consultar posteriormente o histórico detalhado dos atendimentos realizados.

### 3. Interface

A Interface Gráfica da aplicação foi desenvolvida utilizando a biblioteca FreeSimpleGUI (PySimpleGUI), garantindo uma interação intuitiva entre o utilizador e o sistema de simulação clínica. A interface principal permite configurar os parâmetros da simulação, executar o modelo e visualizar os resultados obtidos.

A interface organiza-se em separadores, facilitando a navegação entre as diferentes funcionalidades. No separador "Controlo", o utilizador pode definir os parâmetros da simulação, como a taxa de chegada de pacientes por hora, o número de médicos disponíveis, o tempo médio de atendimento e o tipo de distribuição estatística dos tempos de atendimento. Após a configuração destes parâmetros, a simulação é iniciada através do botão “Simular”.

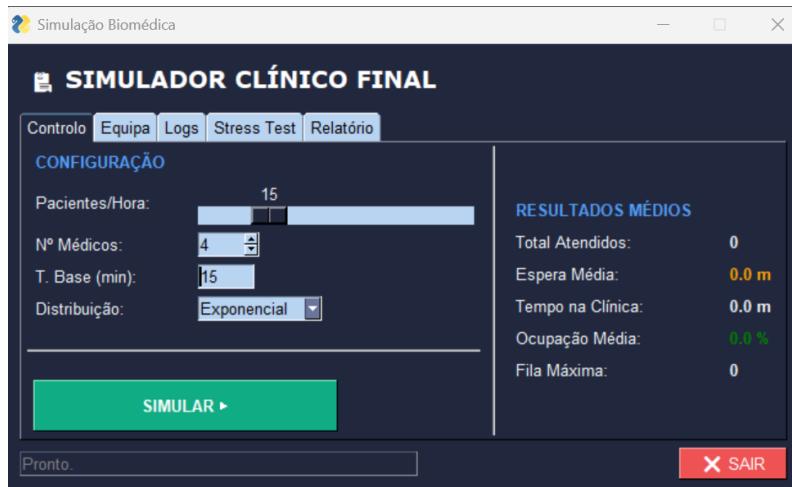


Figure 1: Interface Principal

No separador "Equipa", o utilizador pode visualizar a especialidade de cada membro da equipa assim como a percentagem de ocupação de cada um, fornecendo uma visão clara da distribuição da carga de trabalho e o tempo médio de consulta de cada médico.

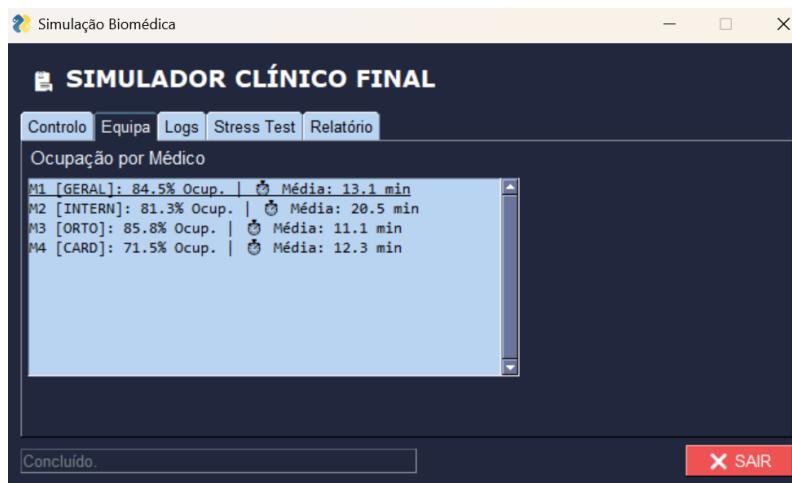


Figure 2: Ocupação por Médico

Neste seguimento, o separador "Logs" apresenta um registo detalhado das atividades de atendimento, mostrando informações como o nome do paciente, idade, número da pulseira e especialidade.

Este registo é atualizado automaticamente com cada simulação, permitindo ao utilizador acompanhar a execução em tempo real.

Simultaneamente, mostra se os utentes são logo atendidos ou se vão para uma fila de espera, aparecendo também o tempo de espera na fila de cada utente.

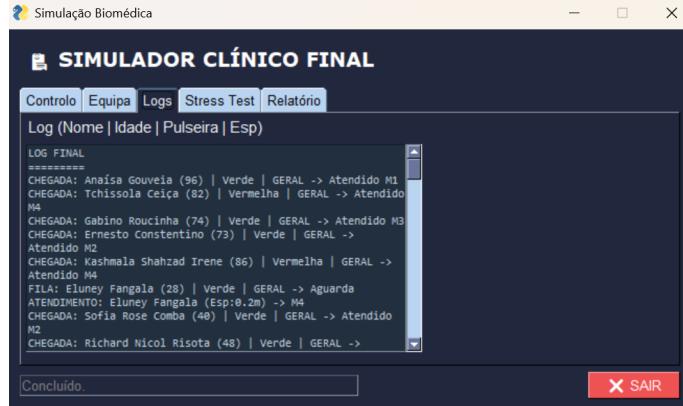


Figure 3: Registo de Eventos da Simulação

A funcionalidade de Análise de Sensibilidade encontra-se no separador “Stress Test”. Ao executar esta funcionalidade, a aplicação varia a taxa de chegada de pacientes e avalia o impacto dessa variação no tamanho médio da fila de espera.

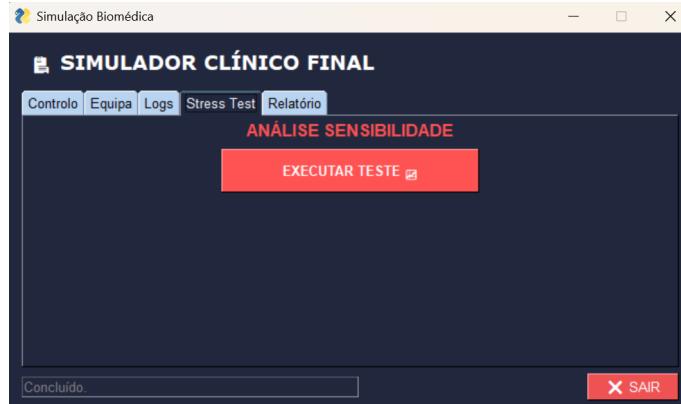


Figure 4: Análise de Sensibilidade do Sistema

Ao clicarmos no botão "Executar Teste", os resultados são apresentados sob a forma de um gráfico, permitindo analisar a robustez do sistema face ao aumento da procura.



Figure 5: Análise de Sensibilidade do Sistema

De seguida, ao fecharmos o gráfico voltamos para o separador e a aplicação apresenta um indicador de estado que informa o utilizador sobre o progresso das operações

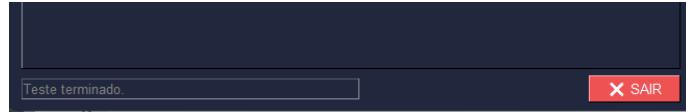


Figure 6: Análise de Sensibilidade do Sistema

No que toca à geração de relatórios gráficos, acessível através do separador “Relatório”.

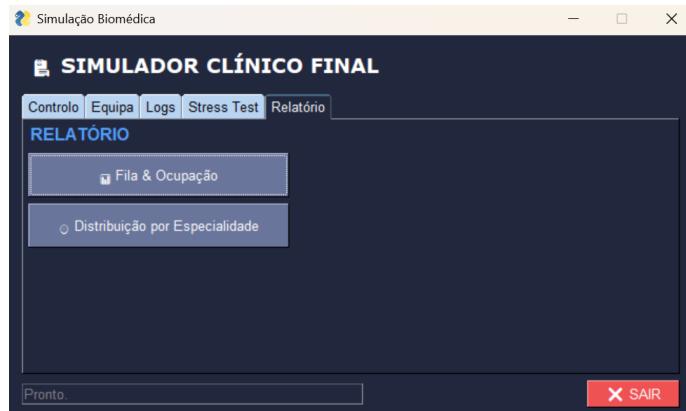


Figure 7: Relatório Gráfico

Nesta secção, ao clicar no botão "Fila & Ocupação", o utilizador pode gerar um gráfico que apresenta a evolução da fila de espera e a taxa de ocupação dos médicos ao longo do tempo.

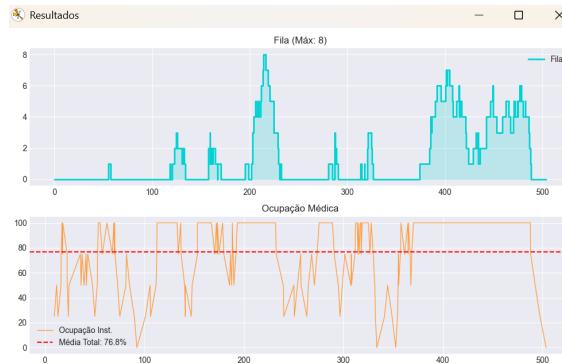


Figure 8: Relatório Gráfico - Fila e Ocupação

Adicionalmente, ao clicarmos em "Distribuição por Especialidade", é possível visualizar um gráfico circular que representa a distribuição dos pacientes atendidos por especialidade médica, permitindo identificar a percentagem associada a cada especialidade.



Distribuição de Pacientes por Especialidade ORTO

Figure 9: Distribuição de Pacientes por Especialidade

Para finalizar, a aplicação permite o encerramento seguro do sistema através do botão "Sair", garantindo a correta finalização da simulação.

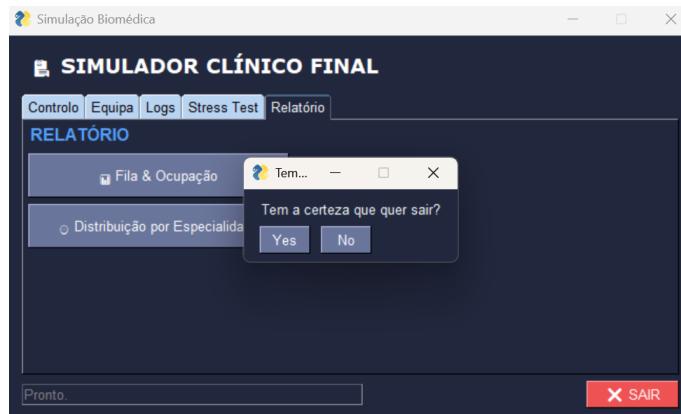
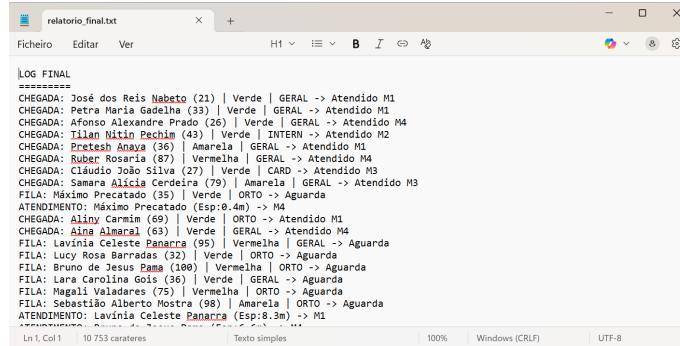


Figure 10: Terminar Simulação

Durante a execução da simulação, é gerado automaticamente o ficheiro `relatorio_final.txt`, que contém um registo detalhado das atividades do sistema. Cada entrada do log descreve eventos como a chegada de pacientes, a entrada na fila de espera, o início e o término do atendimento por cada médico, incluindo a especialidade e a prioridade do paciente. Este ficheiro é criado diretamente pelo módulo de simulação, sem necessidade de intervenção do utilizador, garantindo que todas as ações ocorridas durante a simulação fiquem registadas de forma cronológica. Posteriormente, o conteúdo do ficheiro é carregado na interface gráfica, no separador “Logs”, permitindo ao utilizador acompanhar o histórico completo da simulação de forma visual e organizada.



```

relatorio_final.txt
Ficheiro Editar Ver H1 ≡ B I ↻ Aº
LOG FINAL
=====
CHEGADA: José dos Reis Nabete (21) | Verde | GERAL -> Atendido M1
CHEGADA: Petra Maria Gadelha (33) | Verde | GERAL -> Atendido M1
CHEGADA: Afonso Alexandre Prado (26) | Verde | GERAL -> Atendido M4
CHEGADA: Ana Paula Oliveira Peceli (43) | Verde | INTERNA -> Atendido M2
CHEGADA: Pratish Anayu (36) | Amarela | GERAL -> Atendido M3
CHEGADA: Ruber Reisaria (87) | Vermelha | GERAL -> Atendido M4
CHEGADA: Cláudio João Silva (27) | Verde | CARD -> Atendido M3
CHEGADA: Samara Alcicio Cerdeira (79) | Amarela | GERAL -> Atendido M3
CHEGADA: Máximo Precatado (35) | Verde | ORTO -> Aguarda
ATENDIMENTO: Máximo Precatado (Esp:8.4m) -> M1
CHEGADA: Ana Paula Oliveira (69) | Vermelha | ORTO -> Atendido M1
CHEGADA: Síria Alvesal (51) | Verde | GERAL -> Atendido M4
FILA: Lavinia Celeste Panarra (95) | Vermelha | GERAL -> Aguarda
FILA: Lucy Rosa Barradas (32) | Verde | ORTO -> Aguarda
FILA: Bruno de Jesus Pama (108) | Vermelha | ORTO -> Aguarda
FILA: Lara Carolina Gois (36) | Verde | GERAL -> Aguarda
FILA: Magali Valadares (75) | Vermelha | ORTO -> Aguarda
FILA: Sebastião Alberto Mostra (98) | Amarela | ORTO -> Aguarda
ATENDIMENTO: Lavinia Celeste Panarra (Esp:8.3m) -> M1
=====

Linha, Coluna 10 753 caracteres Texto simples 100% Windows (CRLF) UTF-8

```

Figure 11: Visualização do Log de Simulação

## 4. Conclusão

De modo geral, através das bibliotecas PySimpleGUI, matplotlib e NumPy, e com a linguagem de programação Python, foi possível desenvolver uma aplicação de simulação clínica que permite ao utilizador gerir e acompanhar o atendimento de pacientes numa clínica virtual. Acedendo a uma base de dados em JSON, o sistema foi capaz de gerar entradas de pacientes com diferentes idades, especialidades e prioridades, simulando o fluxo de atendimento de acordo com os recursos disponíveis.

O utilizador consegue definir parâmetros da simulação, como o número de médicos, taxa de chegada de pacientes, distribuição dos tempos de consulta e duração média de atendimento. Após a execução da simulação, a aplicação apresenta gráficos de evolução da fila de espera, ocupação dos médicos e distribuição de pacientes por especialidade, permitindo uma análise visual clara e imediata do desempenho do sistema. Adicionalmente, é gerado automaticamente o ficheiro `relatorio_final.txt`, que regista todas as atividades ocorridas durante a simulação, podendo ser consultado diretamente no separador *Logs* da interface.

Para além dos requisitos básicos, foram implementadas:

- Atendimento a pessoas reais (doentes têm nome e idade);
- Utentes com prioridades (relacionada com a idade dos utentes) devido à implementação de pulseiras;
- Implementação de especialidades: Geral, Medicina Interna, Ortopedia, Cardiologia, Pediatria (nunca atende adultos), e cada especialidade tem tempos médios de consulta diferentes;
- Gráfico circular da distribuição dos pacientes atendidos por especialidade médica.

Desta forma, a realização deste trabalho permitiu um maior aprofundamento nos conceitos de simulação de filas e gestão de recursos em saúde, bem como a familiarização com bibliotecas gráficas e de análise de dados em Python. De uma forma geral, consideramos que alcançamos o objetivo do projeto, visto que a aplicação é funcional, interativa e consegue executar as operações previstas.

Todavia, há sempre aspectos que podem ser melhorados, como:

- Adição de mais métricas estatísticas;
- Melhorias na prioridade da fila, aproximando ainda mais de uma realidade clínica;
- Adicionar tempo de descanso dos médicos para tornar a simulação mais realista.

Por fim, tendo em atenção a complexidade e o alcance das tarefas desenvolvidas, este projeto permitiu aumentar os nossos conhecimentos, desafiando-nos a encontrar soluções criativas e a melhorar competências em programação, análise de dados e desenvolvimento de interfaces gráficas.

## 5. Webgrafia

Website do manual do PySimpleGUI [site](#).

Website criado pelo professor José Carlos Ramalho *ATP2021 programação* [site](#).