



Universidade do Minho
Escola de Engenharia

LICENCIATURA EM ENGENHARIA BIOMÉDICA

ALGORITMOS E TÉCNICAS DE PROGRAMAÇÃO

Clínica

Realizado por:

Íris Teixeira, A110111

Joana Carneiro, A111136

Leonor Magalhães, A103829

Docente: José Carlos Ramalho

Docente Convidado: Luís Filipe Cunha

11 de janeiro de 2026

Índice

1	Introdução	1
2	Análise e Especificação dos Requisitos	2
2.1	Descrição Informal do Problema	2
2.2	Dataset Utilizado	3
2.3	Funcionalidades do Sistema	4
3	Conceção e Proposta de Resolução	6
3.0.1	Interface Principal	6
3.0.2	Carregamento da Base de Dados	6
3.0.3	Inserção de paciente	7
3.0.4	Ficha do Doente	8
3.1	Relatórios	8
4	Conclusão	12

Lista de Figuras

1	Interface principal	6
2	Interface principal com base de dados carregada.	7
3	Inserir doente	7
4	Ficha do doente.	8
5	Página de relatórios.	8
6	Gráfico da evolução da fila de espera.	9
7	Gráfico da evolução da ocupação médica por intervalo.	9
8	Gráfico da evolução da fila de espera.	10
9	Gráfico de situação dos doentes.	10
10	Gráfico de distribuição de doentes atendidos por especialidade.	11

1 Introdução

No âmbito da Unidade Curricular de Algoritmos e Técnicas de Programação, foi solicitada a implementação de uma aplicação em Python com o objetivo de desenvolver um sistema de gestão hospitalar. A aplicação desenvolvida, designada *Hospital*, tem como principal finalidade apoiar a gestão de informação relativa aos doentes, bem como permitir a análise do funcionamento do serviço de urgências através de simulações e relatórios gráficos. O sistema desenvolvido possibilita a leitura e interpretação de uma base de dados composta por estruturas de dados complexas, nomeadamente listas e dicionários interligados, representando um conjunto de pessoas e respetivos atributos. A aplicação permite percorrer essa informação, organizá-la e apresentá-la de forma estruturada ao utilizador, garantindo uma correta manipulação e acesso aos dados. Adicionalmente, o projeto integra um módulo de simulação do serviço de urgências, que possibilita a análise do comportamento das filas de espera e da ocupação dos médicos em função de parâmetros como a taxa de chegada de doentes, o número de médicos disponíveis e a capacidade da sala de espera. Os resultados da simulação são apresentados através de gráficos estatísticos, incluindo a evolução das filas, a taxa de ocupação dos médicos e a comparação entre o número de doentes atendidos e o número de óbitos registados. Para assegurar uma interação clara e eficiente com o utilizador, foi desenvolvida uma interface gráfica intuitiva, recorrendo às bibliotecas FreeSimpleGUI, matplotlib, datetime e json.

2 Análise e Especificação dos Requisitos

2.1 Descrição Informal do Problema

O problema abordado neste trabalho consiste na simulação do funcionamento de uma clínica médica, com o objetivo de modelar e analisar o atendimento de doentes ao longo do tempo. Pretende-se representar um sistema real em que os doentes chegam à clínica de forma aleatória, são atendidos por um número limitado de médicos e, caso não exista disponibilidade imediata, aguardam numa fila de espera até poderem ser consultados.

A chegada dos doentes é modelada através de um processo estocástico, seguindo uma distribuição de Poisson, enquanto a duração das consultas é também aleatória, obedecendo a uma distribuição estatística pré-definida (exponencial, normal ou uniforme). Durante a simulação, são registadas diversas métricas de desempenho do sistema, tais como os tempos de espera dos doentes, o tamanho das filas de espera e o nível de ocupação dos médicos.

No final da simulação, pretende-se analisar estes dados e representá-los graficamente, permitindo estudar o impacto da variação de parâmetros do sistema, como a taxa de chegada de doentes ou o tempo médio de consulta, no desempenho global da clínica. Desta forma, o sistema desenvolvido possibilita a avaliação do comportamento da clínica em diferentes cenários e contribui para uma melhor compreensão da dinâmica de sistemas de atendimento em contexto real.

2.2 Dataset Utilizado

O conjunto de dados utilizado neste trabalho corresponde ao ficheiro `personas.json`, armazenado em formato JSON. A estrutura do dataset consiste numa lista de dicionários, em que cada dicionário representa um indivíduo e agrega os respetivos atributos demográficos e clínicos relevantes para a simulação do funcionamento da clínica. Esta organização permite um acesso simples e flexível à informação de cada pessoa, facilitando a sua integração no modelo de simulação e o tratamento estatístico dos dados ao longo da execução do sistema.

De seguida apresenta-se um excerto ilustrativo de um dos dicionários presentes no dataset, omitindo alguns campos para simplificação.

```
[
  {
    "nome": "Keira Shailesh Corte-Real",
    "idade": 96,
    "sexo": "feminino",
    "morada": {
      "cidade": "Vila do Conde",
      "distrito": "Porto"
    },
    "profissao": "Pescador",
    "desportos": [
      "Columbofilia",
      "Natação",
      "... "
    ]
  }
]
```

2.3 Funcionalidades do Sistema

Gestão da Fila de Eventos (enqueue e dequeue)

As funções enqueue e dequeue são responsáveis pela gestão da fila de eventos da simulação. A função enqueue insere um novo evento na fila de forma ordenada pelo instante temporal de ocorrência, garantindo que os eventos são processados por ordem cronológica. Por sua vez, a função dequeue remove e devolve o próximo evento a ser processado, correspondendo sempre ao evento com menor tempo associado. Estas funções são fundamentais para a implementação correta da simulação de eventos discretos.

Procura da Posição na Fila (procuraPosQueue)

A função procuraPosQueue determina a posição correta onde um novo evento deve ser inserido na fila de eventos, com base no seu instante temporal. Esta função é utilizada internamente pelo mecanismo de inserção ordenada, permitindo manter a fila sempre organizada sem necessidade de ordenações adicionais.

Geração de Recursos Humanos (gera_medicos e gera_enfermeiros)

As funções gera_medicos e gera_enfermeiros são responsáveis pela criação das equipas de médicos e enfermeiros disponíveis na simulação. A geração dos médicos tem em conta as diferentes especialidades existentes no sistema, garantindo uma distribuição coerente com os pesos definidos para cada especialidade. Já os enfermeiros são utilizados no processo de triagem, sendo modelados como recursos limitados que podem estar ocupados ou livres ao longo do tempo.

Atribuição de Especialidade Prevista (gera_especialidade_prevista)

A função gera_especialidade_prevista determina a especialidade clínica mais provável para um doente antes da consulta, com base em características individuais, nomeadamente a idade. Esta abordagem permite modelar decisões clínicas simples, como o encaminhamento automático de doentes pediátricos para a especialidade de Pediatria, aumentando o realismo da simulação.

Seleção do Próximo Doente (prox_doente)

A função prox_doente seleciona o próximo doente a ser atendido numa determinada especialidade, respeitando os níveis de prioridade clínica (pulseiras). Os doentes são escolhidos de acordo com a prioridade definida, garantindo que situações mais urgentes são atendidas primeiro, antes dos casos menos graves.

Simulação de Eventos (simula_eventos)

A função simula_eventos constitui o núcleo do sistema de simulação. É nesta função que são gerados e processados todos os eventos relevantes, tais como chegadas de doentes, início e fim de triagens, início e fim de consultas médicas. Ao longo da simulação são registadas métricas essenciais, incluindo tempos de espera, ocupação dos médicos e número de doentes atendidos, permitindo posteriormente a análise estatística e a geração de gráficos de desempenho do sistema.

Verificação de Óbito (verifica_obito)

A função verifica_obito modela a possibilidade de ocorrência de óbito durante o processo de atendimento, com base em fatores clínicos do doente, como a pulseira de prioridade, idade ou hábitos de risco. Esta funcionalidade acrescenta uma dimensão adicional de realismo à simulação, permitindo estudar cenários extremos e os seus impactos na base de dados e nos resultados globais do sistema.

3 Conceção e Proposta de Resolução

3.0.1 Interface Principal

Utilizámos a biblioteca PySimpleGUI para desenvolver a Interface Gráfica da aplicação, tornando mais simples e intuitiva a gestão da informação hospitalar. De seguida, é apresentada a Interface Principal, que disponibiliza os botões correspondentes às principais operações possíveis de realizar na aplicação, nomeadamente o carregamento e armazenamento da base de dados, a inserção de doentes, a alteração do estado dos registos e a consulta da informação existente.

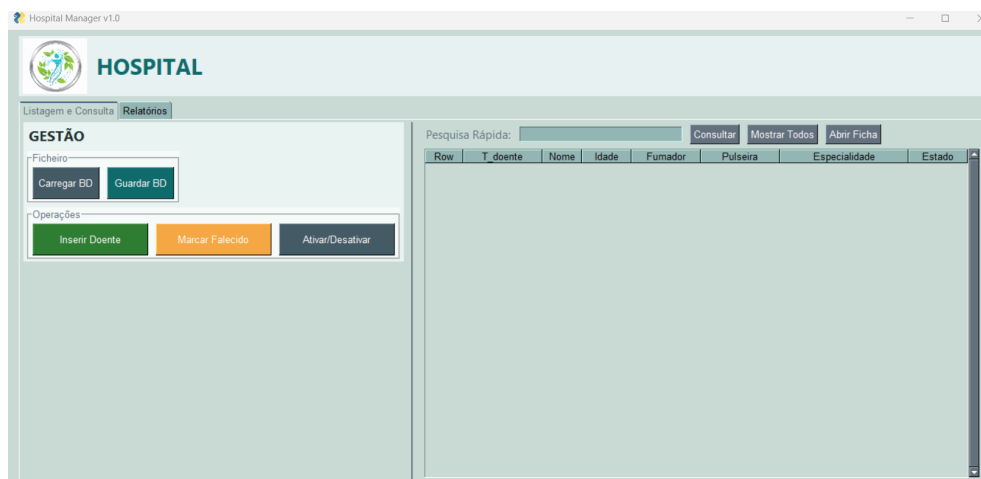


Figura 1: Interface principal

3.0.2 Carregamento da Base de Dados

Ao clicar no botão "Carregar BD", o utilizador adiciona a base de dados que pretende em formato JSON e inicia o processo de carregamento.

Row	T. doente	Nome	Idade	Fumador	Pulseira	Especialidade	Estado
0	HS-1	Neyanni	47	Sim	verde	Ortopedia	ATIVO
1	HS-2	Santiago	88	Sim	azul	Medicina Geral	ATIVO
2	HS-3	Yona C	40	Não	vermelho	Medicina Geral	ATIVO
3	HS-4	Iolanda	73	Sim	verde	Neurologia	ATIVO
4	HS-5	Matissa	60	Sim	azul	Medicina Geral	ATIVO
5	HS-6	David A	94	Sim	laranja	Medicina Geral	ATIVO
6	HS-7	Ilie Filip	43	Não	laranja	Cardiologia	ATIVO
7	HS-8	Vanvara	99	Sim	verde	Ortopedia	FALECIDO
8	HS-9	Erica S	69	Não	verde	Medicina Geral	FALECIDO
9	HS-10	Sofia Jo	40	Sim	verde	Neurologia	ATIVO
10	HS-11	Clara P	90	Sim	verde	Ortopedia	ATIVO
11	HS-12	Abderra	86	Não	verde	Ortopedia	ATIVO
12	HS-13	Emily A	70	Sim	laranja	Medicina Geral	ATIVO
13	HS-14	Idrissa	66	Não	amarela	Cardiologia	ATIVO
14	HS-15	Elmar C	68	Sim	amarela	Medicina Geral	ATIVO
15	HS-16	Kimia C	67	Não	verde	Ortopedia	ATIVO
16	HS-17	Consoli	83	Sim	vermelho	Medicina Geral	ATIVO
17	HS-18	Sérgio J	90	Não	verde	Ortopedia	ATIVO
18	HS-19	Danish	57	Não	verde	Cardiologia	ATIVO
19	HS-20	Mária S	90	Não	verde	Medicina Geral	ATIVO
20	HS-21	Pureza	20	Não	laranja	Ortopedia	ATIVO
21	HS-22	Abdalm	94	Não	azul	Ortopedia	FALECIDO
22	HS-23	Nunia C	99	Não	vermelho	Ortopedia	FALECIDO
23	HS-24	Gemaia	76	Sim	verde	Medicina Geral	ATIVO
24	HS-25	Ruthy L	24	Não	verde	Neurologia	ATIVO
25	HS-26	Huata L	99	Sim	verde	Ortopedia	ATIVO

Figura 2: Interface principal com base de dados carregada.

3.0.3 Inserção de paciente

Para a inserção de novos registos, foi desenvolvida uma janela específica, na qual o utilizador introduz os dados necessários através de campos de texto e listas de seleção, garantindo uma utilização simples e controlada.

Figura 3: Inserir doente

3.0.4 Ficha do Doente

A aplicação permite ainda a abertura de uma ficha individual do doente, onde é possível visualizar e editar informação detalhada, como dados pessoais, especialidade atribuída, estado do doente e outros campos relevantes.

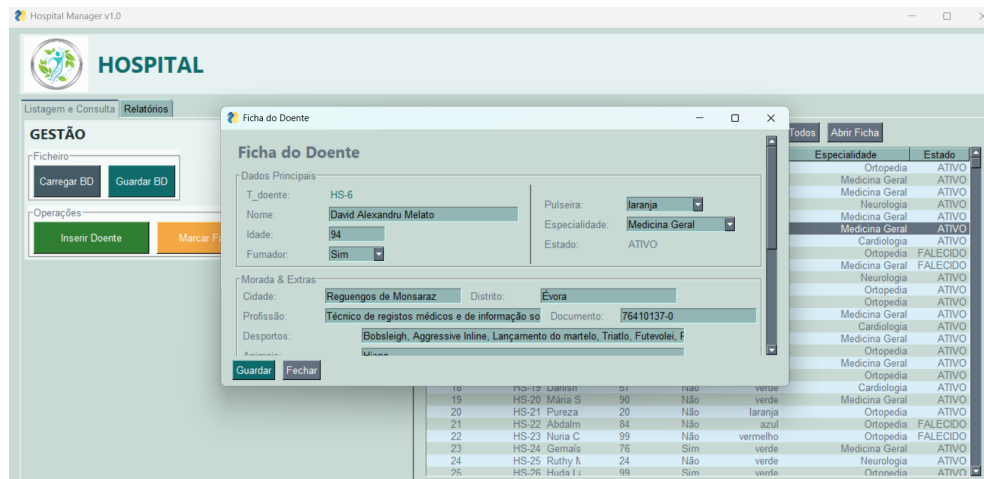


Figura 4: Ficha do doente.

3.1 Relatórios

A página de Relatórios foi concebida para permitir a realização de simulações e a visualização de resultados gráficos associados ao funcionamento do sistema hospitalar. Nesta interface, o utilizador pode definir um conjunto de parâmetros relevantes, como a taxa de chegada de doentes, o número de médicos disponíveis e a capacidade da sala de espera, ajustando assim o comportamento da simulação.

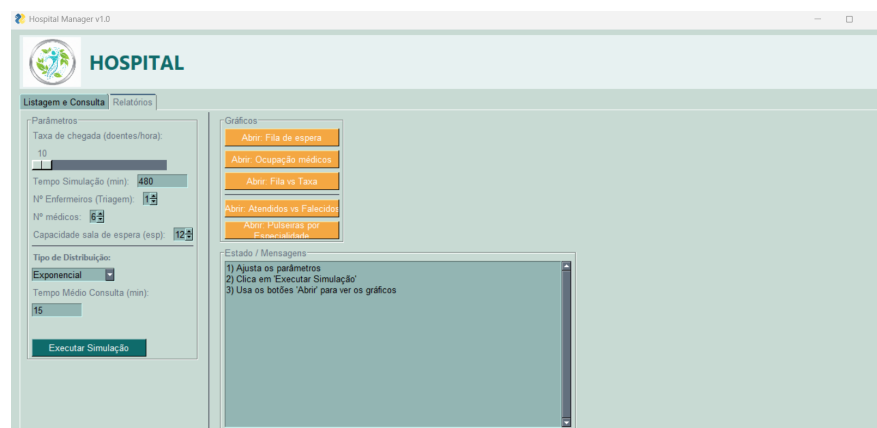


Figura 5: Página de relatórios.

A aplicação disponibiliza um conjunto de botões que permitem o acesso aos diferentes gráficos gerados pela simulação. Ao clicar em cada um dos botões apresentados, é aberta uma nova janela contendo o respectivo gráfico, possibilitando ao utilizador visualizar e analisar os resultados obtidos de forma clara e organizada.

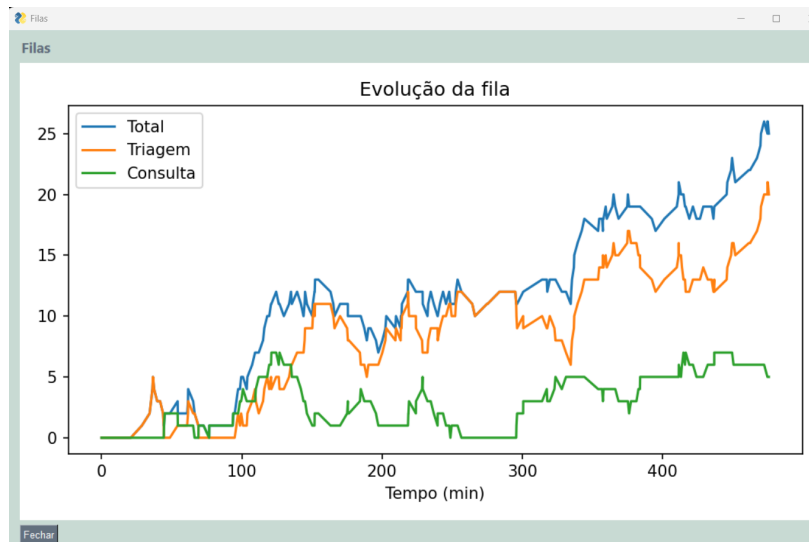


Figura 6: Gráfico da evolução da fila de espera.

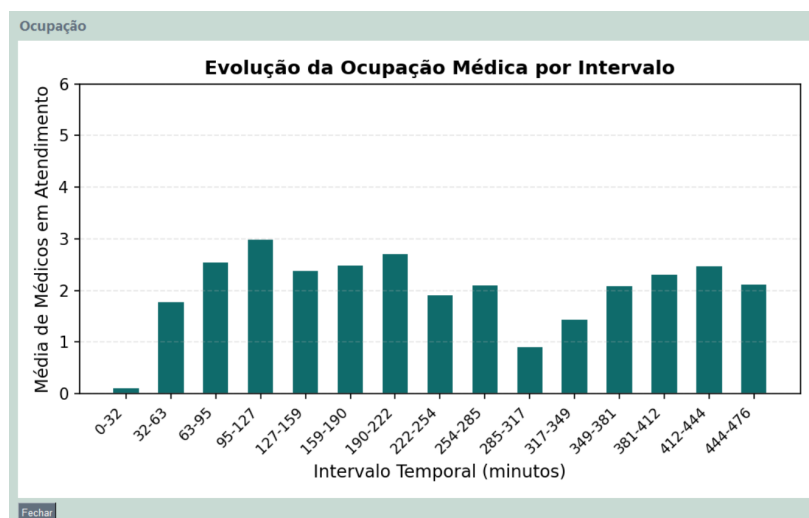


Figura 7: Gráfico da evolução da ocupação médica por intervalo.

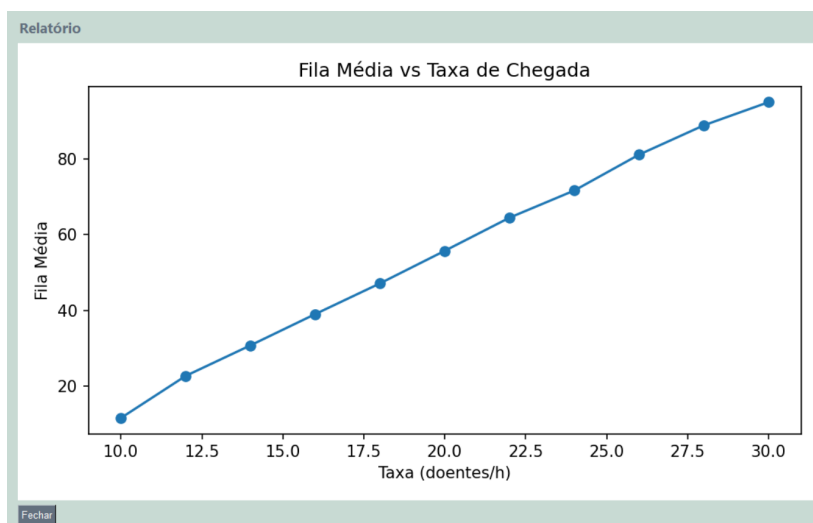


Figura 8: Gráfico da evolução da fila de espera.

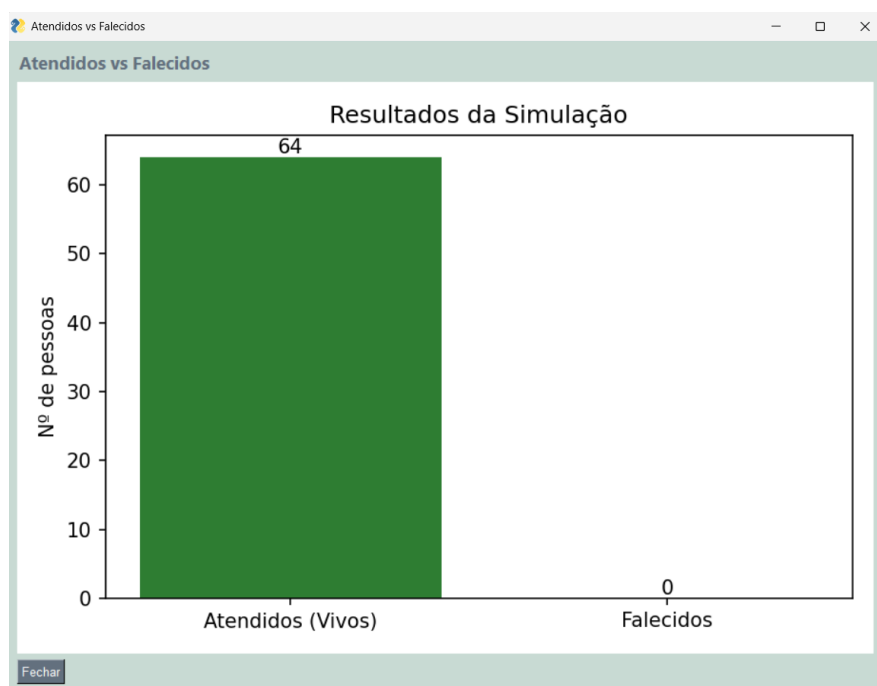


Figura 9: Gráfico de situação dos doentes.

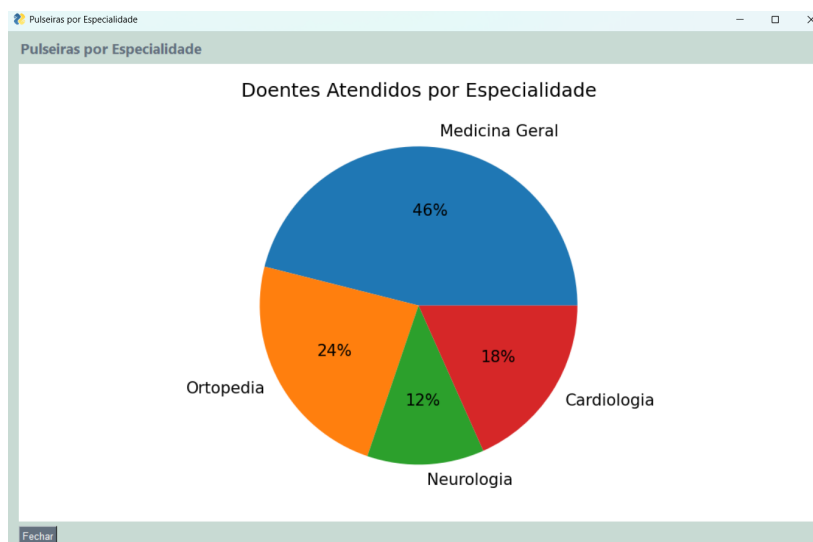


Figura 10: Gráfico de distribuição de doentes atendidos por especialidade.

4 Conclusão

A realização deste projeto permitiu consolidar os conhecimentos adquiridos na Unidade Curricular de Algoritmos e Técnicas de Programação, através do desenvolvimento de uma aplicação em Python capaz de carregar, manipular e visualizar uma base de dados estruturada em formato JSON.

O sistema desenvolvido possibilita a leitura e interpretação de uma base de dados composta por estruturas de dados complexas, nomeadamente listas e dicionários, representando um conjunto de pessoas e respetivos atributos. A aplicação permite percorrer essa informação, organizar os dados e apresentá-los de forma estruturada ao utilizador, garantindo uma correta manipulação e acesso à informação.

A utilização de uma Interface Gráfica (GUI) contribuiu para uma interação mais intuitiva com o sistema, enquanto a integração da biblioteca matplotlib permitiu a representação gráfica de dados simples, facilitando a visualização da informação existente na base de dados.

De uma forma geral, considera-se que o projeto cumpre os requisitos definidos, demonstrando uma correta aplicação dos princípios abordados na unidade curricular. Como trabalho futuro, poderão ser implementadas melhorias, nomeadamente ao nível da interface gráfica e da organização da informação, mantendo a coerência com a estrutura da base de dados utilizada.