



UNIVERSIDADE D  
**COIMBRA**

Faculdade de Ciências e Tecnologias da Universidade de Coimbra

*Hospicare*

**Unidade Curricular: Base Dados**

**Docente: Adorilson Bezerra de Araujo**

Tiago Rafael Cardoso Santos, 2021229679

Tomás Cunha da Silva, 2021214124

# Índice

<b>Introdução.....</b>	<b>2</b>
<b>Funcionalidades desenvolvidas na aplicação.....</b>	<b>3</b>
<b>Divisão de Tarefas.....</b>	<b>5</b>
<b>Conclusão.....</b>	<b>6</b>

# Introdução

Este relatório foi desenvolvido no âmbito da unidade curricular de Base de Dados, onde foi sugerida a realização de um trabalho prático visando o desenvolvimento de uma base de dados de simulação e gestão de um hospital.

Ao nível da base de dados, a aplicação gere utilizadores, pagamentos, agendamentos de consultas ou cirurgias. Existem diferentes tipos de utilizadores, desde pacientes a médicos, enfermeiras e assistentes.

# Funcionalidades desenvolvidas na aplicação

## 1. Registro de Usuários

Endpoints:

Paciente: POST <http://localhost:8080/dbproj/register/patient>

Assistente: POST <http://localhost:8080/dbproj/register/assistant>

Enfermeiro: POST <http://localhost:8080/dbproj/register/nurse>

Médico: POST <http://localhost:8080/dbproj/register/doctor>

Descrição:

Cada endpoint permite o registro de um novo usuário no sistema, incluindo detalhes específicos de cada tipo de usuário (e.g., especialidade para médicos). Os dados necessários incluem nome de usuário, e-mail, senha, entre outros atributos específicos.

## 2. Autenticação de Usuário

Endpoint: PUT <http://localhost:8080/dbproj/user>

Descrição:

Permite que o usuário faça login utilizando seu nome de usuário e senha, recebendo um token de autenticação (JWT) em caso de sucesso. Este token deve ser usado nas requisições subsequentes para autenticação.

## 3. Agendamento de Consulta

Endpoint: POST <http://localhost:8080/dbproj/appointment>

Descrição:

Permite que um paciente agende uma nova consulta com um médico. Requer o ID do médico, a data da consulta e outros detalhes relevantes. Uma fatura é gerada automaticamente usando triggers.

## 4. Visualização de Consultas

Endpoint: GET [http://localhost:8080/dbproj/appointments/{patient\\_user\\_id}](http://localhost:8080/dbproj/appointments/{patient_user_id})

Descrição:

Permite que assistentes e o paciente visualizem todas as consultas agendadas para um paciente específico, incluindo detalhes como nome do médico e data.

## 5. Agendamento de Cirurgia

Endpoints:

POST <http://localhost:8080/dbproj/surgery>

POST http://localhost:8080/dbproj/surgery/{hospitalization\_id}

Descrição:

Permite que assistentes agendem uma nova cirurgia. Pode-se associar a cirurgia a uma hospitalização existente ou criar uma nova. Inclui detalhes como ID do paciente, ID do médico, IDs dos enfermeiros e a data. Uma fatura é gerada ou atualizada automaticamente usando triggers.

#### 6. Obtenção de Prescrições

Endpoint: GET http://localhost:8080/dbproj/prescriptions/{person\_id}

Descrição:

Permite que pacientes e funcionários obtenham a lista de prescrições e seus detalhes para um paciente específico.

#### 7. Adição de Prescrições

Endpoint: POST http://localhost:8080/dbproj/prescription/

Descrição:

Permite que médicos adicionem uma nova prescrição associada a uma consulta ou hospitalização. Inclui detalhes sobre a validade da prescrição e a posologia dos medicamentos prescritos.

#### 8. Execução de Pagamento

Endpoint: POST http://localhost:8080/dbproj/bills/{bill\_id}

Descrição:

Permite que pacientes paguem suas faturas. A fatura é atualizada para o status "paga" após a conclusão do pagamento.

#### 10. Resumo Diário

Endpoint: GET http://localhost:8080/dbproj/daily/{year-month-day}

Descrição:

Permite que assistentes visualizem um resumo diário dos procedimentos realizados no hospital, incluindo hospitalizações, cirurgias, pagamentos e prescrições, todos obtidos através de uma única consulta SQL.

## Divisão de Tarefas

Tarefa	Responsável	Horas
Arquitetura da base de dados	Todos	10
Implementação da base de dados	Todos	15
Desenvolvimento dos serviços pretendidos	Todos	30
Inclusão da API REST	Todos	15
Relatório	Todos	4

Tiago Santos:

- Criar a função de registar um Utilizador
- Criar a função de fazer login de um Utilizador
- Criar a função de marcar consulta
- Criar a função de criar receita médica
- Criar a função de visualizar receitas médicas
- Criar a função de ver consultas marcadas

Tomás Silva:

- Criar a função de visualizar as consultas marcadas
- Criar a função de visualizar receitas médicas
- Criar a função de receita médica
- Criar a função de marcar Surgery
- Criar a função de Report Diário

O projeto estimou que cada membro dedicasse cerca de 30 horas para a conclusão. Optamos por utilizar uma pasta compartilhada na plataforma de armazenamento na drive para guardar diagramas, queries e scripts. Além disso, criamos um repositório no GitHub que ia sendo atualizado por um membro. Realizamos um total de quatro reuniões durante o projeto, e o restante foi dedicado ao desenvolvimento da aplicação, com todos os membros a trabalhar em conjunto.

# Conclusão

Ao longo da realização deste trabalho prático foi possível consolidar e pôr em prática diversos conceitos abordados nas aulas Teórico-práticas da cadeira de Base de Dados, estabelecendo uma base sólida para uma plataforma simplificada de gestão de um Hospital com capacidades robustas de gerenciamento de base de dados.

É importante referir que o desenvolvimento da base de dados da plataforma atende de forma eficaz às funcionalidades e restrições de negócio necessárias para armazenamento e processamento de informações.

Sendo assim, as noções desta cadeira estão, agora, mais claras.