



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
**COIMBRA**

# Bases de Dados

**Licenciatura em Engenharia Informática**  
**2024**

**Relatório | Projeto**

## **Sistema de Gestão Hospitalar**

**Projeto realizado por:**

**Gonçalo Leitão | 2022222043 | uc2022222043@student.uc.pt**

**João Filipe Valente Cardoso | 2022222301 | uc2022222301@student.uc.pt**

**Sérgio Lopes Marques | 2022222096 | uc2022222096@student.uc.pt**

# **Índice**

Descrição do Projeto.....	2
Manual de Instalação .....	2
Manual do Utilizador .....	3
Plano de Desenvolvimento.....	7

## Descrição do Projeto

O projeto a ser desenvolvido tem como objetivo implementar um sistema de gestão hospitalar que seja capaz de otimizar e garantir o correto funcionamento de todos os processos inerentes à administração de um hospital. Para isso, será criada uma base de dados que se comunicará com uma REST API, através de um servidor web.

Esta comunicação requer um cuidado especial e terá de ser feita de forma correta, de modo a assegurar a eficiência de todos os processos que serão realizados. Nestes processos estão incluídos os agendamentos de consultas, cirurgias ou internamentos, prescrições de medicamentos, pagamentos de possíveis despesas, bem como a gestão de recursos humanos (nomeadamente funcionários).

## Manual de Instalação

### Introdução

Este manual descreve o processo de instalação e execução do software desenvolvido, cujo objetivo é conectar um código Python a uma base de dados através do uso do Postman.

### Instalação

Antes de tentar arrancar o nosso software, certifique-se de ter os seguintes pré-requisitos instalados no seu computador:

- IDE e Interpretador de Python
- Postman (ferramenta para testes de APIs)
- PostgreSQL (sistema para gerar a base de dados)

### Correr o Software

Após verificar que a conexão entre o código Python e base de dados foi estabelecida, crie as tabelas necessárias através dos scripts fornecidos pelo , conforme a informação presente no enunciado. Será ainda, necessário guardar as

informações da conexão (user, password, host, port, database) que serão usadas no código Python.

Por fim, abra o Postman e crie uma nova requisição. Configure o método HTTP e o URL do endpoint que deseja testar. Deverá fornecer ainda, os parâmetros necessários no corpo da requisição, se aplicável. Envie a requisição e verifique se a API está a comunicar corretamente com a base de dados, garantindo que todas as operações estejam a funcionar conforme o esperado.

## Conclusão

Ao seguir as instruções indicadas acima, deverá ser capaz de implementar e executar o nosso software que conecta um código Python a uma base de dados usando o Postman. Certifique-se de testar o funcionamento do sistema para garantir que tudo esteja conforme o planeado.

# Manual do Utilizador - Sistema de Gestão Hospitalar

Bem-vindo ao nosso Sistema de Gestão Hospitalar! Este manual fornece instruções detalhadas sobre como utilizar as funcionalidades disponíveis da aplicação.

## Autenticação do Utilizador

Para autenticar um utilizador, utilize o seguinte comando:

- PUT `http://localhost:8080/dbproj/user`

Após o registo, poderá efetuar o login fornecendo o nome de utilizador e a palavra-passe. Após o sucesso da autenticação, receberá um token de reconhecimento que deve ser incluído no cabeçalho de todas as solicitações subsequentes.

## Adicionar Paciente

Para adicionar um paciente, utilize o seguinte comando:

- POST `http://localhost:8080/dbproj/register/patient`

Ao enviar a solicitação de registo, forneça as informações necessárias, como o nome e a palavra-passe. O sistema irá retornar um ID de utilizador após o registo bem-sucedido.

## Adicionar Assistente

Para adicionar um assistente, utilize o seguinte comando:

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

Ao enviar a solicitação de registo, forneça as informações necessárias, como o nome e a palavra-passe. O sistema irá retornar um ID de utilizador após o registo bem-sucedido.

## Adicionar Enfermeiro

Para adicionar um enfermeiro, utilize o seguinte comando:

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

Ao enviar a solicitação de registo, forneça as informações necessárias, como o nome e a palavra-passe. O sistema irá retornar um ID de utilizador após o registo bem-sucedido.

## Adicionar Médico

Para adicionar um médico, utilize o seguinte comando:

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

Ao enviar a solicitação de registo, forneça as informações necessárias, como o nome e a palavra-passe. O sistema irá retornar um ID de utilizador após o registo bem-sucedido.

## Marcar Consultas

Apenas pacientes podem marcar consultas. Utilize o seguinte comando:

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

Ao enviar a solicitação de marcação de consulta, inclua o ID do médico e a data desejada. O sistema retornará um ID de consulta após a marcação bem-sucedida.

## Ver Consultas

Para visualizar todas as consultas marcadas para um paciente específico, utilize o seguinte comando:

- GET `http://localhost:8080/dbproj/appointments/{patient_user_id}`

Assistentes e pacientes podem utilizar esta funcionalidade enviando o ID do paciente para receber uma lista de todas as consultas marcadas.

## Marcar Cirurgia

Apenas assistentes podem marcar cirurgias. Utilize o seguinte comando:

- POST `http://localhost:8080/dbproj/surgery`

Ao solicitar a marcação de uma cirurgia, forneça o ID do paciente, do médico, dos enfermeiros envolvidos e a data da cirurgia.

## Ver Prescrições

Para ver as prescrições, utilize o seguinte comando:

- GET `http://localhost:8080/dbproj/prescriptions/{person_id}`

Pacientes podem enviar solicitações para visualizar as prescrições médicas. As prescrições contêm o tipo de evento (consulta ou hospitalização), o ID do evento, a validade da prescrição e os detalhes dos medicamentos prescritos.

## Adicionar Prescrição

Apenas médicos podem adicionar prescrições. Utilize o seguinte comando:

- POST `http://localhost:8080/dbproj/prescription`

Médicos podem adicionar prescrições após consultas ou hospitalizações.

## Efetuar Pagamento

Para realizar um pagamento numa conta, utilize o seguinte comando:

- POST `http://localhost:8080/dbproj/bills/{bill_id}`

Pacientes devem fornecer o valor e o método de pagamento utilizado na solicitação.

## Ver Top 3

Para visualizar um relatório dos três utilizadores que gastaram mais dinheiro durante um determinado mês, utilize o seguinte comando:

- GET `http://localhost:8080/dbproj/top3`

Apenas assistentes podem acessar esta funcionalidade. O relatório detalha os procedimentos realizados por esses pacientes.

## Resumo Diário

Para visualizar informações sobre as atividades de um determinado dia, utilize o seguinte comando:

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

Apenas assistentes podem acessar o relatório diário que resume todas as hospitalizações, cirurgias, pagamentos e prescrições realizadas nesse dia.

## Resumo Mensal

Para visualizar informações sobre os médicos com o maior número de cirurgias realizadas em cada mês dos últimos 12 meses, utilize o seguinte comando:

- GET `http://localhost:8080/dbproj/report`

Esta funcionalidade ajuda na análise do desempenho dos médicos e na gestão de recursos hospitalares.

## **Plano de Desenvolvimento**

Nesta meta do projeto, vimos-nos obrigados a fazer algumas alterações no diagrama ER de modo a termos a que as tabelas contivessem as informações pretendidas. De seguida, o foco passou a estar totalmente nos endpoints e, por fim, desenvolvemos tanto o manual de instalação como o manual do utilizador. Apesar de na primeira meta termos definido tarefas para cada membro do grupo, todos contribuíram de algum modo para cada tópico que foi realizado nesta entrega final e por isso, o número de horas foi distribuído de certa forma igual para todos.