

---

User manual

# Hospital Management System

---

Gonçalo José Carrajola Gaio 2022224905  
João Ricardo Teixeira Gaspar Madeira 2022200648  
Rodrigo Carvalho dos Santos 2022218283

Coimbra, maio de 2024



Licenciatura em Engenharia Informática  
Base de Dados

# Índice

1	Introdução	2
2	Atualização das componentes	2
3	Descrição do projeto	3

# 1 Introdução

O **Sistema de Gestão Hospitalar** é uma solução desenvolvida para coordenar um amplo conjunto de atividades no âmbito da administração hospitalar.

Para poder interagir eficazmente com esta solução é necessário conhecer o funcionamento dos seus componentes, descritos no presente manual.

## 2 Atualização das componentes

Durante o desenvolvimento do trabalho, deparamo-nos com a inevitabilidade de realizar alterações no mesmo. Estas mudanças foram o culminar de uma análise contante das necessidades do projeto, por isso é essencial que estas estejam descritas de forma clara.

- **Diagrama**

1. **registration:**

O valor *default* da data é **CURRENT\_DATE**

2. **prescription:**

O valor *default* da data é **CURRENT\_DATE**

- **Physical**

1. A tabela `nurse_nurse` foi renomeada para:

**sup\_nurse.**

2. `nurse_employee_person_cc1` na tabela `sup_nurse` foi renomeado para:

**low\_nurse\_employee\_person\_cc.**

3. `specialization_specialization` foi renomeado para:

**sup\_specialization.**

4. `specialization_spec_id1` na tabela `sup_specialization` foi renomeado para:

**low\_specialization\_spec\_id.**

- "sur\_date" and "hosp\_date", no código, apenas têm o nome "name"
- Para ajudar no processamento da **bill** foi criado um *trigger* que a cada nova consulta, hospitalização ou cirurgia a fatura é atualizada.
- Para pesquisas mais céleres considerando grandes quantidades de dados, foram implementados *indexes* nas colunas mais utilizadas.

### 3 Descrição do projeto

Para testar o sistema de armazenamento e gestão de dados utilizamos 12 endpoints que podem ser consultados no ficheiro **Hospital\_Man\_Sys.py**.

Nos *endpoints* que assim o exigem é executada a receção do Payload, com campos de preenchimento comuns a todos os *users* e a validação dos campos comuns de preenchimento;

- **landing\_page**

A *landing page* apresenta as informações sobre o projeto e sua autoria.

- **insert\_type**

Este *endpoint* é utilizado para registar novos *users* do Sistema de Gestão Hospitalar. Podendo estes serem pacientes, enfermeiros, médicos ou assistentes.

1. Validação dos campos específicos de cada *user*;
2. Criação das *queries* para inserção dos novos dados nas tabelas da base de dados;
3. Execução das *queries*, caso não seja bem sucedida, é executado um *rollback*, para garantir a integridade dos dados;
4. Confirmação do sucesso da tarefa;

- **authenticate\_user**

Este *endpoint* é utilizado para executar a autenticação dos *users* do Sistema de Gestão Hospitalar. Verifica as credencias fornecidas pelo *user* e se estiverem corretas devolve o token de acesso.

1. Criação de uma *query* para verificar a senha e obter o número do Cartão de Cidadão, executa esta consulta para os vários tipos de *user*;
2. Para cada *user* verifica se os dados estão corretos, caso estejam retorna o token de acesso;

- **schedule\_appointment**

Este *endpoint* é utilizado para agendar consultas *users* no Sistema de Gestão Hospitalar. Recebe os dados e cria um registo de consulta na base de dados.

1. Receção do Payload, e validação dos campos **assistant\_id**, **cost**, **appoint\_date**, **doctor\_id**, **nurse\_id**;
2. Criação de registos nas respetivas tabelas dos dados fornecidos pelo *user*;

- **get\_patient\_appointments**

Este *endpoint* é utilizado para listar todas as consultas de um paciente em específico no Sistema de Gestão Hospitalar. Devolva uma lista de consultas associado ao fornecido.

1. O *endpoint* recebe como parâmetro o id do paciente;
2. Executa uma consulta para selecionar as consultas associadas ao paciente;
3. Caso as informações sejam encontradas serão convertidas para uma lista de dicionários;

4. Caso não existam consultas associadas ou paciente não exista, será indicado um erro;

- **schedule\_new\_surgery\_nh**

Este *endpoint* é utilizado para agendar uma cirurgia de um paciente, que não esteja hospitalizado, no Sistema de Gestão Hospitalar.

1. Verificação dos custos e dos id dos enfermeiros;
2. Execução de consultas para inserir os dados nas tabelas;
3. Caso a cirurgia seja agendada com sucesso, devolve o seu id;

- **schedule\_new\_surgery**

Este *endpoint* é utilizado para agendar uma cirurgia de um paciente, que já esteja associado a uma hospitalização, no Sistema de Gestão Hospitalar.

1. Verificação da hospitalização associada ao paciente;
2. Verificação dos custos e dos id das enfermeiras;
3. Execução de consultas para inserir os dados nas tabelas;
4. Caso a cirurgia seja agendada com sucesso, devolve o seu id;

- **get\_patient\_prescriptions**

Este *endpoint* é utilizado para listar as prescrições associadas a respetivo paciente, no Sistema de Gestão Hospitalar.

1. Verifica se o *user* é um paciente;
2. Executa uma consulta para procurar as prescrições associadas ao paciente;
3. Caso sejam encontradas com sucesso serão devolvidas as prescrições resultantes das consultas e hospitalizações daquele paciente;
4. Caso não existam prescrições será indicado um erro.

- **add\_prescriptions**

Este *endpoint* é utilizado para adicionar uma novas prescrição a uma paciente, no Sistema de Gestão Hospitalar. Esta tarefa apenas pode ser executada por médico.

1. Verifica e valida todos os campos necessários presentes na *request*;
2. Verifica se o id da consulta/hospitalização existe na base de dados;
3. Insere os dados da prescrição, os dados dos medicamentos prescritos e associa os medicamentos à prescrição;
4. Caso ocorra algum erro será devolvida uma mensagem.

- **pay\_bill**

Este *endpoint* é utilizado para executar o pagamento de um determinado paciente, no Sistema de Gestão Hospitalar. Esta tarefa apenas pode ser executada por um paciente.

1. Valida os campos **amount**, **type** e verifica se o **amount** é positivo.
2. Verifica se a **bill** existe e pertence ao paciente que está a aceder à base de dados;
3. No processamento do pagamento, caso o valor da soma do que já foi pago mais o pagamento for superior ao valor a pagar, o valor do pagamento é ajustado;
4. Se o valor total pago for igual ao valor a pagar, o **status** da **bill** será atualizado.

- **get\_top3\_patients**

Este *endpoint* é utilizado para listar os três pacientes, que mais dinheiro gastaram no mês, no Sistema de Gestão Hospitalar.

1. O *endpoint* executa uma query para calcular os três pacientes que mais gastaram em hospitalizações e consultas;
2. A *query* utiliza o comando "WITH" para criar uma *Common Table Expression*, de forma a calcular os custos totais de todos os pacientes da base de dados;
3. Os dados serão organizados no formato lista de dicionários, onde cada dicionário representa o pacientes e os procedimentos a que foi submetido;

- **list\_daily\_summary**

Este *endpoint* é utilizado para listar todos os acontecimentos associados com hospitalizações num determinado dia , no Sistema de Gestão Hospitalar. Esta tarefa apenas pode ser executada por assistente.

1. Verifica e se a data introduzida está no formato correto;
2. Constroi uma *query* para devolver o total de contas a pagar, pagamentos efetuados, número de cirurgias calendarizadas e total de prescrições;
3. Caso um destes parâmetros se encontre inicializado como **None** estão serão enviados como 0;

- **generate\_monthly\_report**

Este *endpoint* é utilizado para listar os médicos com mais cirurgias, em cada mês nos últimos 12 meses, no Sistema de Gestão Hospitalar. Esta tarefa apenas pode ser executada por um assistente.

1. A *query* utiliza uma *Common Table Expression* para poder calcular para cada médico o número de cirurgias realizadas em cada mês;
2. No processamento dos dados é contruída uma lista de dicionários, que contêm o mês e ano, nome do médico e número de cirurgias realizadas;