



UNIVERSIDADE D  
**COIMBRA**

Alexandre Kazuya Dias Pereira  
Diogo Baptista Costa  
Tomás Pereira Rosa Leal Caçoete

# **HOSPITAL MANAGEMENT SYSTEM**

## **DATABASES PROJECT**

**Relatório do Projeto**

**Relatório do Projeto no âmbito da unidade curricular de Base  
de Dados da Licenciatura de Engenharia Informática.**

Maio de 2024

# ÍNDICE

MANUAL DE INSTALAÇÃO .....	<b>3</b>
REQUISITOS .....	3
CONFIGURAÇÃO DA APLICAÇÃO .....	3
CONFIGURAÇÃO DO POSTMAN .....	3
MANUAL DE UTILIZADOR .....	<b>4</b>
FINAL ER AND RELATIONAL DATA MODELS .....	<b>5</b>
PHYSICAL DIAGRAM.....	5
CONCEPTUAL DIAGRAM.....	5
DEVELOPMENT PLAN .....	<b>6</b>

# MANUAL DE INSTALAÇÃO

O primeiro passo para a utilização do sistema é a sua instalação. Neste breve manual irão ser apresentadas instruções passo a passo para conseguir replicar o sistema no seu dispositivo e usufruir das suas funcionalidades.

## REQUISITOS:

- PostgreSQL
- Python
- Postman

Configuração da Base de Dados:

Inicialmente será necessário fazer a criação das tabelas e estruturas utilizadas pela base de dados. Para tal basta apenas aceder a qualquer método de edição do PostgreSQL como, por exemplo, o pgAdmin. Posteriormente deve então criar uma base de dados bem como um novo utilizador e uma nova conexão. Por último será necessário executar o script “CriarBaseDeDados.sql” na plataforma que escolheu previamente.

**Não avance enquanto não tenha conseguido executar o script da criação da base de dados!**

## CONFIGURAÇÃO DA APLICAÇÃO (API):

Após a conclusão do script inicial iremos agora inicializar a API. Para tal abra um terminal do seu sistema operativo e dirija-se à pasta onde se encontra o ficheiro “requirements.txt”. Execute o comando “pip install -r requirements.txt” e deixe instalar as bibliotecas. Em caso de erro verifique se tem o Python e o Pip instalados corretamente. Depois de ter as bibliotecas instaladas pode então executar o seguinte comando no terminal: “python src/main.py”.

Caso tenha algum problema verifique se as bibliotecas foram instaladas corretamente.

**Não avance enquanto não tenha conseguido executar a API!**

## CONFIGURAÇÃO DO POSTMAN:

Por último, para fazer usar a API e a base de dados iremos configurar o Postman. Abra a aplicação e crie uma conta caso não tenha. Após criar uma conta, carregue no botão Import que se encontra no canto superior esquerdo da aplicação e irá reparar que é aberto um campo para onde pode arrastar ficheiros. No mesmo local onde se encontrava o ficheiro “requirements.txt” está também uma pasta com o nome “postman”. Arraste o único ficheiro que se encontra na pasta Postman para o campo aberto anteriormente.

# MANUAL DO UTILIZADOR

Este manual fornece instruções detalhadas sobre como interagir com o Postman para utilizar o sistema após a sua instalação e configuração. Toda a seguinte informação requer, do utilizador, um conhecimento básico do Postman.

Em primeiro lugar, um novo utilizador terá de executar o seu registo através dos pedidos “Register {role}”. Caso seja um utilizador existente pode passar para as instruções de login que se encontram mais abaixo. Cada tipo de utilizador necessita de utilizar o pedido destinado à sua função onde preenche informações diferentes. Um paciente apenas necessita de fornecer o seu nome, username, password e mail. Um assistente necessita de todos os anteriores mais o seu salário. Um enfermeiro todos os anteriores mais uma lista com os ids das *nurses* pelas quais é responsável caso existam. Um médico necessita dos mesmos dados que um assistente, porém, caso tenha alguma especialidade, deve também introduzir uma lista com o nome das mesmas.

Com o registo concluído o utilizador terá de executar o seu login na plataforma de modo a se autenticar perante o sistema. Caso contrário não terá acesso a qualquer funcionalidade. Para proceder a tal basta selecionar o pedido “Login User” e adicionar ao *payload* o seu username e a correspondente password. Em caso de erro quer pela não existência do utilizador ou pela password estar incorreta o sistema irá notificá-lo. Quando o login for efetuado com sucesso irá obter uma resposta com um *token* de autenticação. Este *token* será necessário em todos os pedidos seguintes para que o sistema consiga gerir as permissões dos seus utilizadores. A sua utilização é feita através da cópia do *token* para a secção “Headers” dos pedidos onde terá de adicionar um *Authorization Header* cujo valor será “Bearer {token}”.

Após a devida autenticação o utilizador está pronto a utilizar o sistema.

Abaixo encontram-se listadas as funcionalidades da aplicação assim como, dentro de parênteses, quem tem permissões para as utilizar:

## SCHEDULE APPOINTMENT: (pacientes)

Um paciente consegue marcar uma consulta através do *request* Schedule Appointment. Para isso basta preencher as informações sobre o médico, assistente e enfermeiros envolvidos na consulta, bem como a data de quando se realizará.

## SEE APPOINTMENTS: (assistentes, paciente alvo)

Um paciente consegue ver as suas consultas e os detalhes associados como, por exemplo, o médico, a data de ocorrência, etc. Além do paciente alvo, os assistentes conseguem ver as todas as consultas de qualquer paciente.

## SCHEDULE SURGERY: (assistentes)

Um assistente pode marcar cirurgias com este pedido que, conseqüentemente, irá criar uma hospitalização e uma nova fatura associadas. Para que tal seja sucedido o assistente necessita de preencher as informações do médico responsável, do assistente, do paciente em questão e uma lista com os ids das enfermeiras bem como o cargo que elas terão na cirurgia.

## SCHEDULE SURGERY WITH HOSPITALIZATION: (assistentes)

Este pedido é bastante semelhante ao anterior, porém, em vez de criar uma hospitalização e fatura quando a cirurgia é marcada, apenas associamos uma cirurgia a uma hospitalização já existente.

## ADD PRESCRIPTION: (médicos)

Um médico consegue passar uma prescrição através deste pedido. O médico precisa de inserir informações acerca da receita como o nome, dosagem e frequência dos medicamentos em questão. Além disso, o médico necessita também de preencher dados sobre o evento que originou a prescrição (*Appointment/Hospitalization*).

## GET PRESCRIPTION: (empregados, paciente alvo)

Os empregados do hospital conseguem executar este pedido para conseguir informações sobre todas as prescrições de um paciente específico. Além dos empregados, os próprios pacientes também conseguem verificar as prescrições que já lhes tenham sido passadas.

## EXECUTE PAYMENT: (paciente alvo)

Com este pedido os pacientes conseguem pagar as suas faturas, quer na totalidade, quer em prestações. O pedido regista um novo pagamento com a informação sobre a data, a quantidade e a despesa a que este pagamento está associado.

## LIST TOP 3 PATIENTS: (assistentes)

Um assistente consegue verificar os três pacientes que mais gastaram no mês atual. Na resposta deste pedido, além dos clientes, estão também as informações sobre os eventos de onde os seus gastos advêm.

## DAILY SUMMARY: (assistentes)

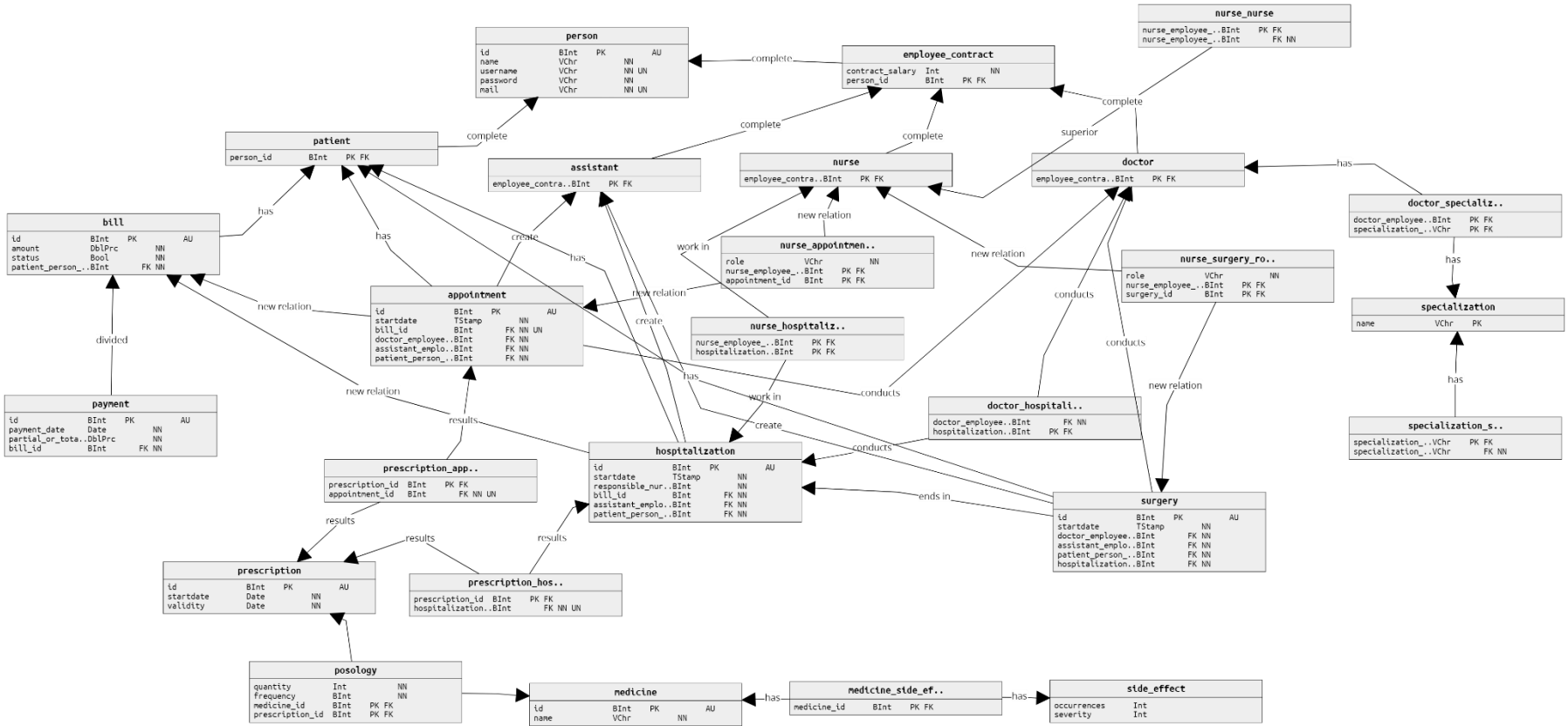
O pedido do Daily Summary devolve informação sobre as hospitalizações que começaram num determinado dia e também do número de cirurgias e prescrições associadas às referidas hospitalizações. Informa ainda o total faturado pelo hospital nos referidos eventos.

## MONTHLY REPORT: (assistentes)

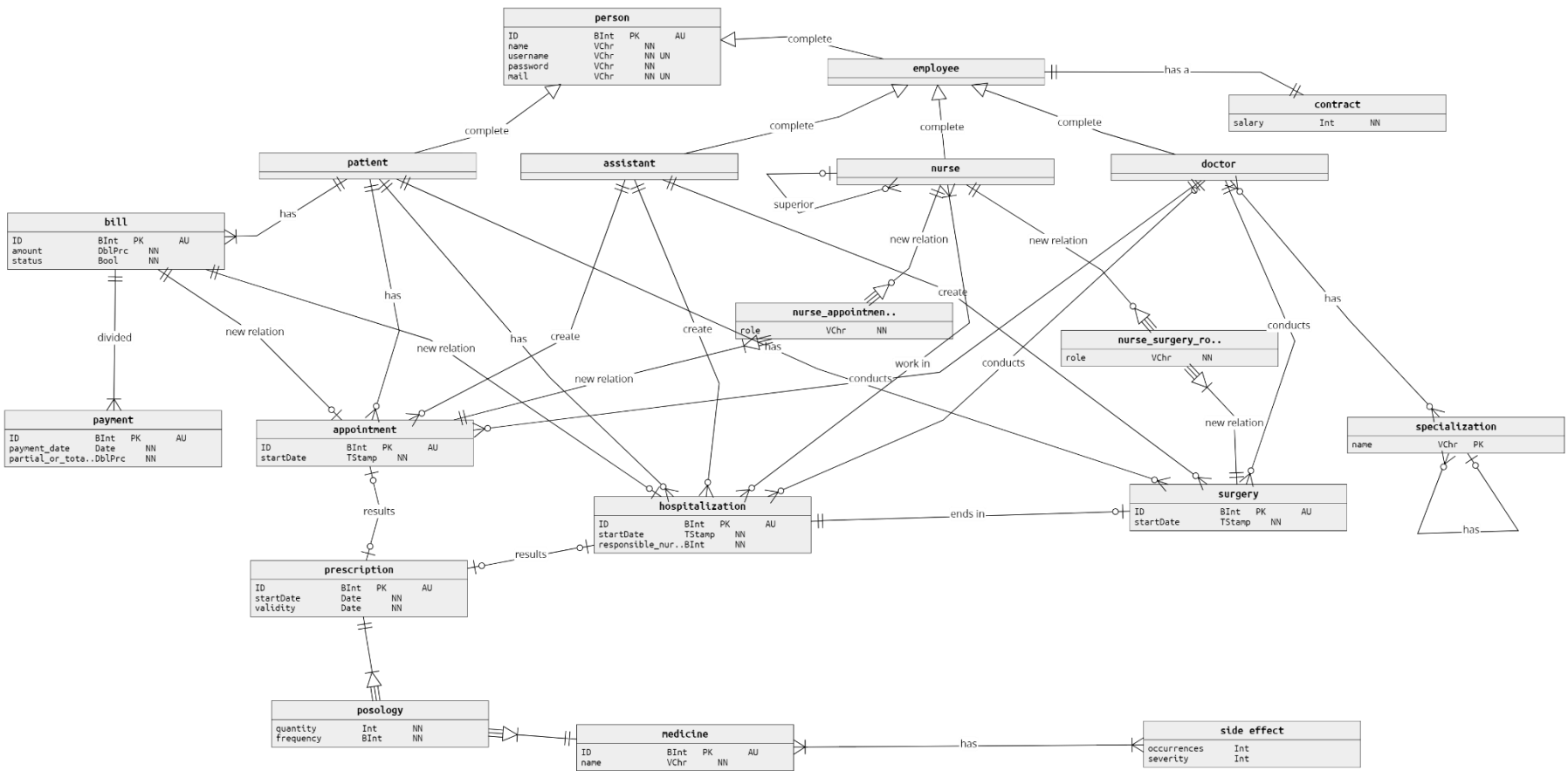
Com este pedido os assistentes conseguem informações sobre o médico que fez mais cirurgias em cada mês do último ano. Este pedido devolve o nome do médico, o mês e quantas cirurgias efetuou nesse período.

# FINAL ER AND RELATIONAL DATA MODELS

## • PHYSICAL DIAGRAM



## • CONCEPTUAL DIAGRAM



# DEVELOPMENT PLAN

Para podermos concluir este projeto necessitámos de implementar as seguintes tarefas:

- Modelação da Base de Dados
- Desenvolvimento da interface do usuário, denominadamente o Login dos utilizadores para conseguirem aceder à aplicação
- Implementação das Queries em SQL/PSQL
- Criação dos endpoints em Python
- Criação de Triggers e mecanismos de Bloqueio na Base de Dados, para lidar com Conflitos de Simultaneidade
- Testes e Validações do Sistema
- Medidas de Segurança

Algumas das tarefas acima descritas foram implementadas em equipa, trabalhando assim em conjunto e sendo deste modo impossível de atribuir a um dos elementos acima.

Contudo, inicialmente tentámos dividir as Queries que eram necessário implementar pelos elementos do grupo de modo uniforme, apesar de aquando da sua conclusão, todos termos discutido e verificado o que cada um tinha feito.

Queries feitas pelo Alexandre:

- POST – Schedule Surgeries
- POST – Schedule Appointment
- GET – See Appointments
- POST – Add Prescriptions

Queries feitas pelo Diogo:

- POST – Schedule Surgeries
- GET – Get Prescriptions
- GET – Daily Summary
- GET – Generate a Monthly Report

Queries feitas pelo Tomás:

- POST – Add Users (Patient, Doctor, Nurse, Assistant)
- PUT – User Authentication
- POST – Execute Payments
- GET – List Top 3 Patients

Para conseguirmos cumprir todos os requisitos e completar o projeto, cada membro do grupo necessitou de despende cerca de 70 horas, incluindo todo o tempo usado desde a idealização do design da Base de Dados até à conclusão do projeto.