



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

# *Relatório Base de Dados*

## *Projeto Base de Dados PL8 Hospital Management System*

---

Alexandre José Martins Rodrigues nº2022249408

Duarte Dinis Pereira nº2022236193

Pedro Filipe Gonçalves Pires nº2022247126

# Diagrama ER (ANTIGO)

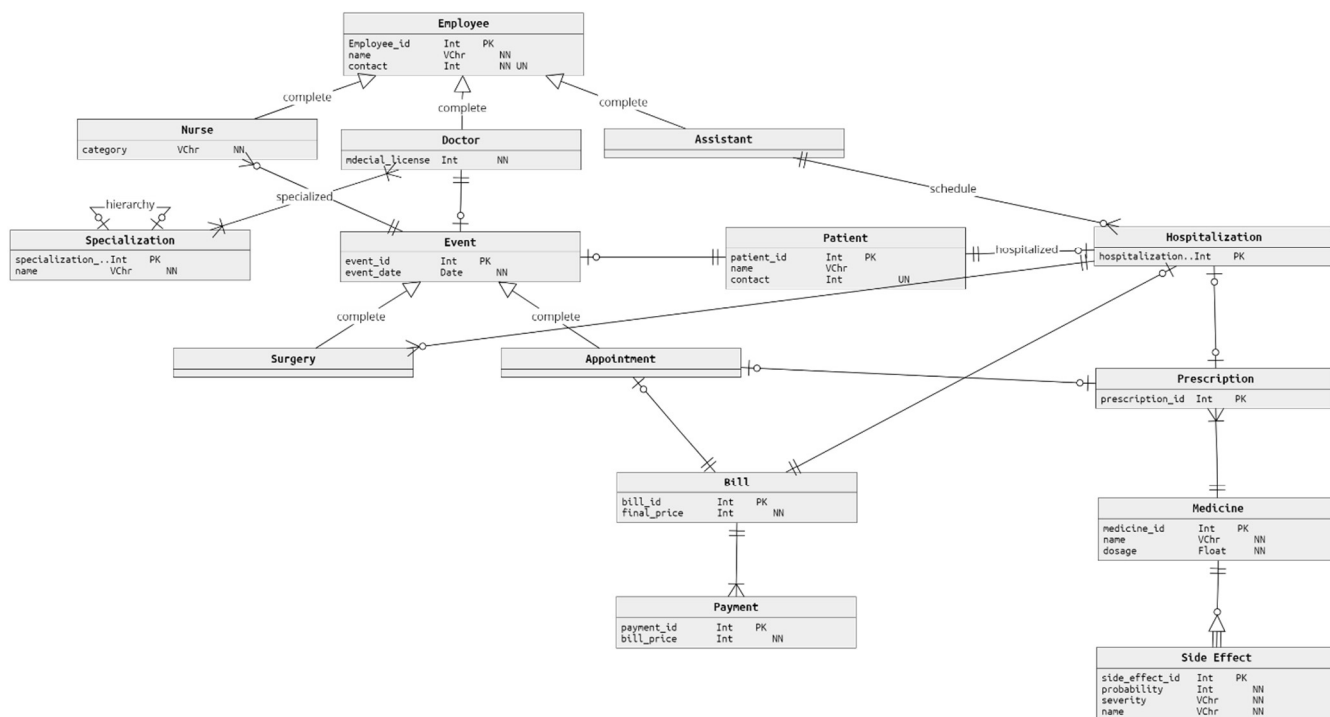


Figure 1 Primeira meta

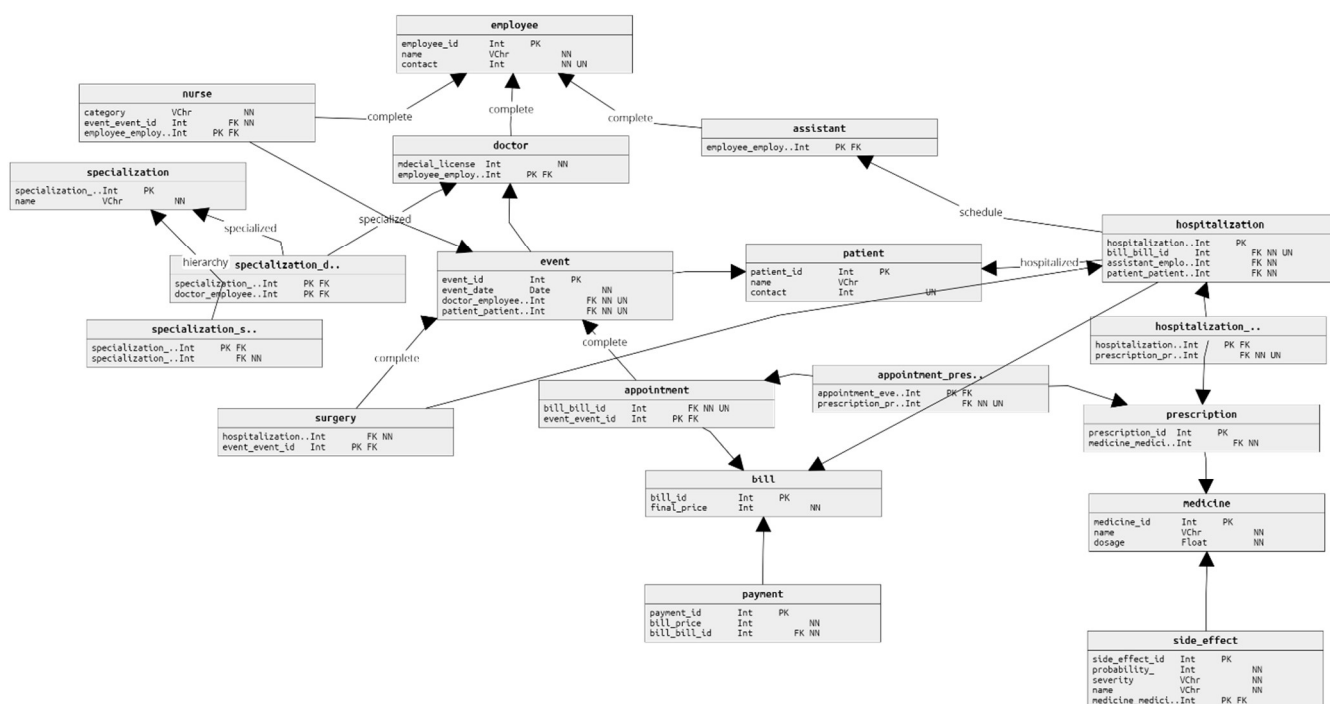


Figure 2 Durante a realização do projeto

# Diagrama ER final

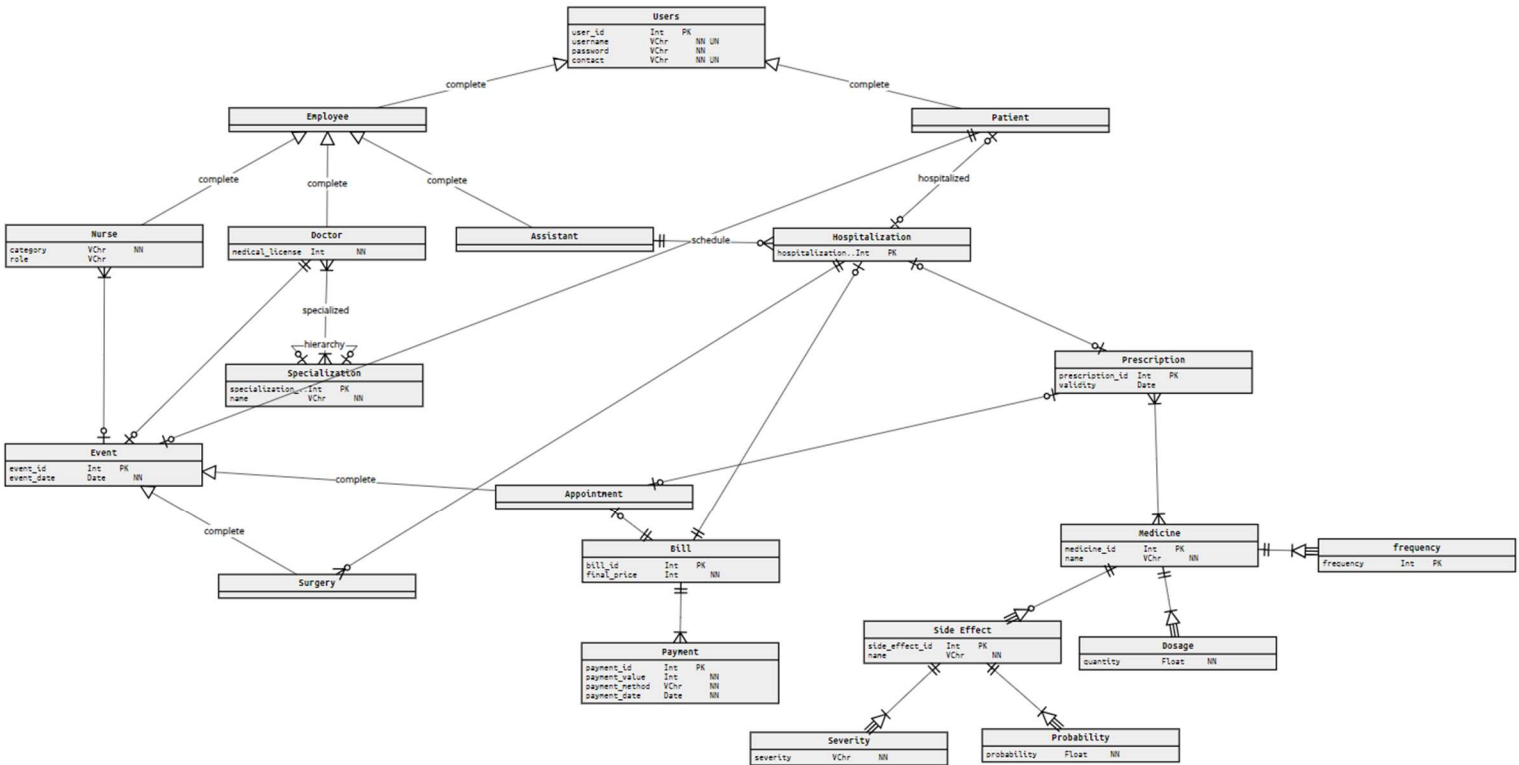


Figura 2 Conceptual

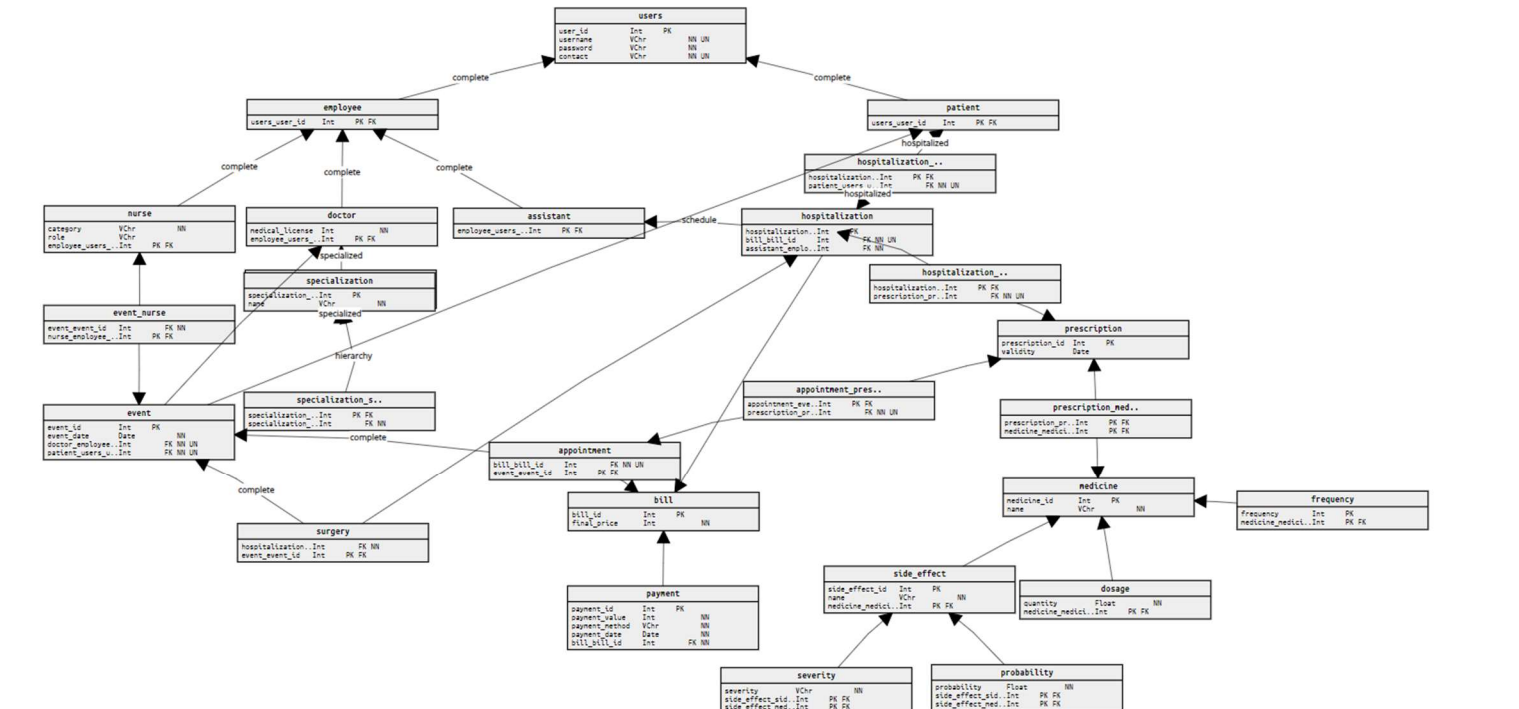


Figura 1 Physical

# Introdução

Este projeto tem como objetivo desenvolver uma aplicação para gerir dados de um hospital usando uma base de dados e aplicando os conhecimentos obtidos na cadeira de Base de Dados.

## Plano de Desenvolvimento

Após a conclusão da primeira versão do diagrama ER, começamos a planear o desenvolvimento da aplicação. Utilizamos Python como linguagem de programação e Postgres como base de dados.

A divisão dos endpoints entre os dois membros do grupo ocorreu da seguinte maneira: cada um ficou responsável por 6 endpoints. O Tomás trabalhou nos endpoints relacionados ao serviço premium, artistas, cartões e playlists, enquanto o Gabriel ficou responsável pelos endpoints de músicas, reports, álbuns e usuários.

No entanto, sempre colaboramos e trabalhamos em conjunto quando necessário. O tempo dedicado por cada membro do grupo foi equilibrado, aproximadamente 20 horas para cada um.

## Detalhes e decisões

Durante o desenvolvimento do projeto, realizamos algumas alterações significativas que consideramos importantes. Isso incluiu ajustes nas relações do diagrama ER e modificações nos tipos de algumas variáveis. Por exemplo, criamos uma tabela específica para armazenar o top10, o que nos permitiu melhorar a estrutura da base de dados.

Todos os requisitos estabelecidos para o trabalho foram cumpridos. Implementamos um sistema de geração automática de IDs sempre que dados são inseridos na base de dados, o que garante a integridade e unicidade dos registros. Além disso, foram adicionadas autorizações utilizando tokens para identificar o nome de usuário e uma função para determinar o tipo de usuário.

Também tomamos medidas para manter a integridade da base de dados, como a inicialização de transações em todos os scripts relevantes.

Essas mudanças e implementações adicionais foram realizadas para aprimorar a funcionalidade e segurança da plataforma de streaming, tornando-a mais robusta e eficiente.

## Instalação

Para este trabalho foi utilizado como linguagem de programação Python, e para comunicar com a base de dados foi usado a aplicação Postman.

**Passo 1:** Deverá instalar a linguagem usada (Python) e um IDE à sua escolha;

**Passo 2:** No IDE instalar as bibliotecas: flask (pip install flask), psycpg2 (pip install psycpg2), jwt (o mesmo já vem com o anaconda se o tiver previamente instalado);

**Passo 3:** Instalar o Postgres para criar e aceder à base de dados;

**Passo 4:** Descarregar e abrir no IDE a pasta onde se encontra o programa em Python;

**Passo 5:** Criar uma base de dados no Postgres;

**Passo 6:** Correr o código com o projeto;

**Passo 7:** Por fim, usar o Postman.

# Manual de Utilização

Add Patient, Doctor, Nurse, and Assistant:

POST

▼

http://localhost:8080/dbproj/register/doctor

ParamsAuthorizationHeaders (9)Body ●ScriptsSettings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▼

1

{

2

"username": "doctor3",

3

"password": "securepassword",

4

"contact": "doctor1@example.com",

5

"medical\_license": 12345

6

}

7

Figure 3 Exemplo: Médico

Schedule Appointment:

POST

▼

http://localhost:8080/dbproj/appointment

ParamsAuthorization ●Headers (11)Body ●ScriptsSettings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▼

1

▼

{

2

"doctor\_id": 1,

3

"date": "2024-06-01T10:00:00"

4

}

5

### See Appointments:

GET

▼

http://localhost:8080/dbproj/appointments/7

ParamsAuthorization ●Headers (10)Body ●ScriptsSettings

☐ none☐ form-data☐ x-www-form-urlencoded☒ raw☐ binary☐ GraphQLJSON ▼

1

### Schedule Surgery:

POST

▼

http://localhost:8080/dbproj/surgery

ParamsAuthorization ●Headers (10)Body ●ScriptsSettings

☐ none☐ form-data☐ x-www-form-urlencoded☒ raw☐ binary☐ GraphQLJSON ▼

1 {  
2     "patient\_id": 7,  
3     "doctor": 3,  
4     "nurses": [  
5         [6, "nurse1"]  
6     ],  
7     "date": "2024-05-31"  
8 }  
9

### Get Prescriptions:

GET

▼

http://localhost:8080/dbproj/prescriptions/1

ParamsAuthorization ●Headers (8)Body ScriptsSettings

☐ none☐ form-data☐ x-www-form-urlencoded☒ raw☐ binary☐ GraphQLJSON ▼

1

## Add Prescriptions: (verificar)

**POST** ▼ http://localhost:8080/dbproj/prescription

Params Authorization ● Headers (10) Body ● Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1  {
2    "type": "appointment",
3    "event_id": 1,
4    "validity": "2024-05-31",
5    "medicines": [
6      {
7        "medicine": "Paracetamol",
8        "posology_dose": 500,
9        "posology_frequency": 3
10     },
11     {
12       "medicine": "Ibuprofen",
13       "posology_dose": 200,
14       "posology_frequency": 2
15     }
16   ]
17 }
18
```

## Execute Payment: (verificar)

**POST** ▼ http://localhost:8080/dbproj/bills/1

Params Authorization Headers (9) Body ● Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1  {
2    "amount": 1,
3    "payment_method": "card"
4  }
5
```



List Top 3 patients:

GET

⌵

http://localhost:8080/dbproj/top3

ParamsAuthorizationHeaders (7)BodyScriptsSettings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

⌵

1

Daily Summary:

GET

⌵

http://localhost:8080/dbproj/daily/1-1-2000

ParamsAuthorizationHeaders (7)BodyScriptsSettings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

⌵

1

Generate a monthly report:

GET

⌵

http://localhost:8080/dbproj/report

ParamsAuthorizationHeaders (7)BodyScriptsSettings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

⌵

1

## TOKENS:

### User Authentication:

PUT

▼

http://localhost:8080/dbproj/user

ParamsAuthorizationHeaders (9)Body●ScriptsSettings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▼

1

{

2

"username": "doctor1",

3

"password": "securepassword"

4

}