

# **Sistema de Gestão de um Hospital**

## **Projeto Base de Dados**

Engenharia de Computadores e Telemática



Ana Rosa, 98678

Vasco Santos, 98391

P2G3

Ano letivo 2021/2022

## Introdução

No âmbito da cadeira de Base de Dados, do Mestrado Integrado em Engenharia de Computadores e Telemática, o projeto que vamos desenvolver é um Sistema de Gestão de um Hospital.

A primeira parte deste relatório foca-se na realização da análise de requisitos, no diagrama Entidade-Relacionamento e o seu mapeamento para o Esquema Relacional.

## Análise de Requisitos

A organização de um hospital é bastante importante para que os departamentos individuais, e mesmo o hospital no geral, funcione de forma correta e consiga tratar os seus utentes o melhor possível. Desta forma, o nosso objetivo será o foco na gestão da informação em geral, isto é, não nos vamos apenas concentrar nos dados de um ou vários departamentos.

A base de dados que irá ser desenvolvida permitirá guardar todos os dados necessários de um utente que esteja a ser tratado no hospital. Poderá possibilitar que qualquer funcionário, se tiver autorização, possa consultar ou alterar os dados do paciente com o objetivo de se manter informado e ir mantendo a informação atualizada.

Os utilizadores deste sistema (funcionários do hospital) devem conseguir:

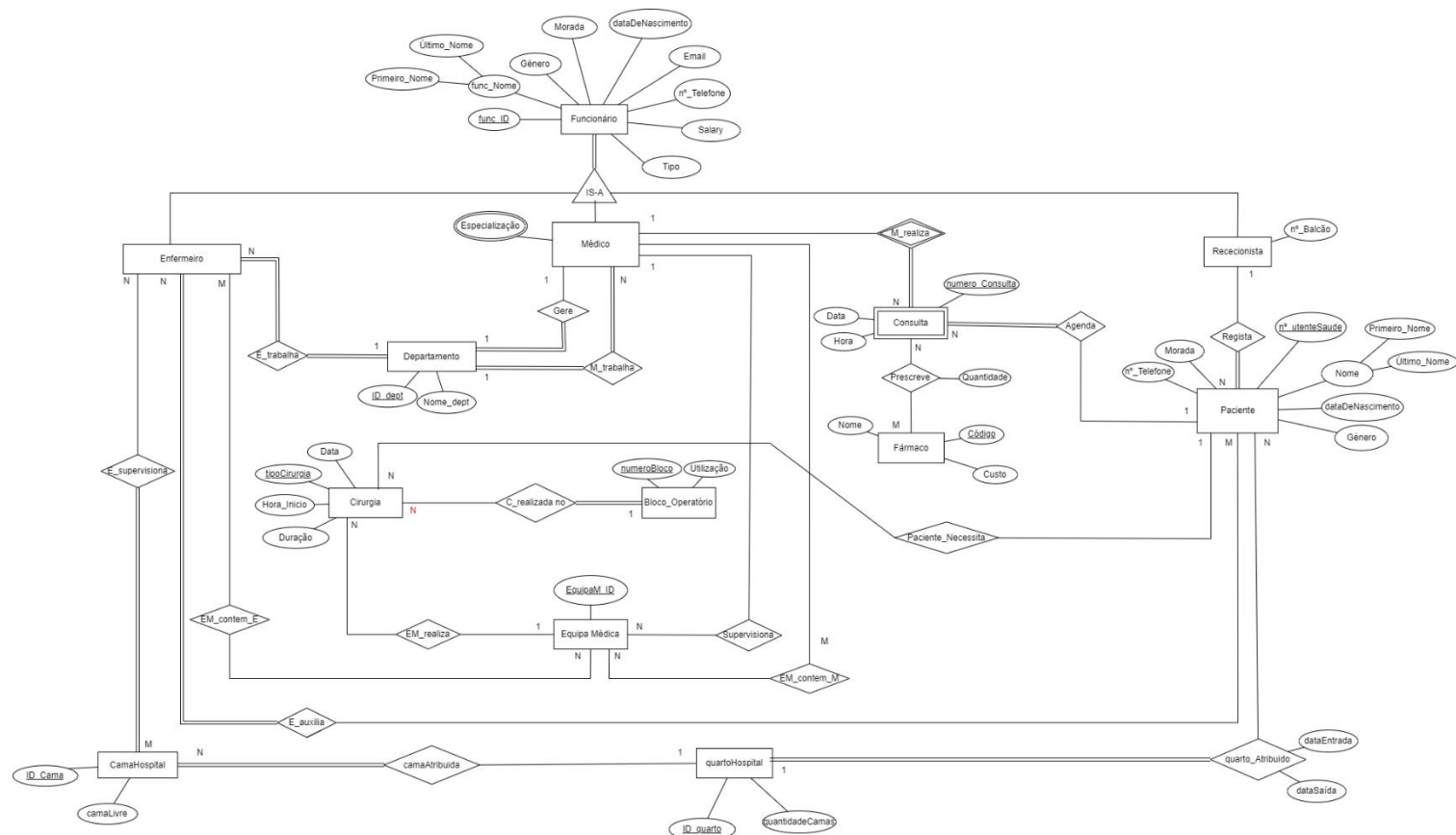
- Registrar um utente não internado na base de dados do hospital.
- Registrar um utente internado na base de dados do hospital, sendo que esta funcionalidade implica atribuir um quarto ao utente verificando se esse quarto está completo ou não.
- Eliminar um Utente não internado da base de dados do sistema.
- Eliminar um Utente internado e, consequentemente libertar uma cama do quarto onde esse paciente estava atribuído.
- Atualizar informação de um Paciente. Os dados que podem ser atualizados podem ser: o primeiro e último nome, data de nascimento, número de telefone, género, ID rececionista, ID quarto, data de entrada e saída. No caso do ID quarto, se ele for preenchido então poderá ser para internar um paciente e se for vazio poderá ser para o paciente deixar de estar internado.
- Pesquisar um paciente em função do seu número de utente de saúde.
- Aceder à lista de Pacientes internados.
- Na lista de Pacientes internados é possível adicionar um paciente e consequentemente atribuir-lhe um quarto com cama livre.
- Na lista de Pacientes internados é possível eliminar um paciente e consequentemente libertar a cama ocupada.
- Na lista de Pacientes internados é possível atualizar os campos de data de entrada e saída.
- Aceder à lista de camas de um determinado quarto, o que dá para verificar, por exemplo, se a cama que está a ser atribuída encontra-se disponível.
- Aceder à lista de enfermeiros supervisores de um determinado quarto.
- Aceder à lista de departamentos e gestores do mesmo.
- Verificar consultas ou cirurgias agendadas.

- Aceder à lista de funcionários do hospital.
- Eliminar um funcionário do hospital, Médico, Enfermeiro ou Rececionista.
- Atualizar informações dos funcionários.
- Consultar a especialização de um Médico. Sendo que o Médico pode ser especializado em mais do que uma área.
- Consultar que médicos e enfermeiros compõem a equipa médica.

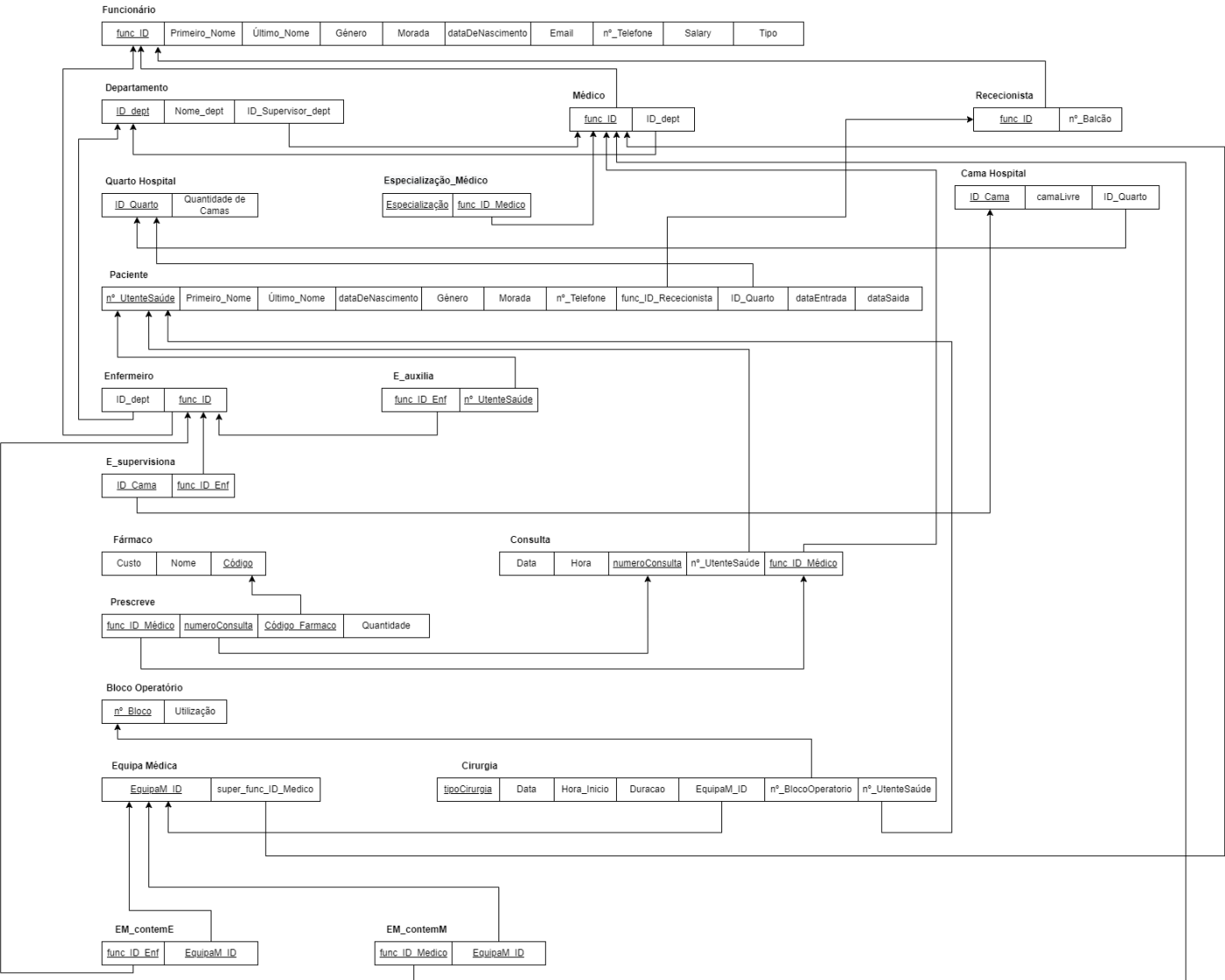
## Diagrama Entidade-Relacionamento

- O **Funcionário** é uma pessoa que trabalha no hospital e é caracterizado por um ID (func\_ID), nome (primeiro e último), género, morada, data de nascimento, email, número de Telefone e tipo (Médico, Enfermeiro, Rececionista). O funcionário pode ser um **Enfermeiro**, um **Médico** ou um **Rececionista**.
- O **Enfermeiro** é caracterizado pelo seu número de identificação (ID) que herda da entidade Funcionário e pelo ID do departamento onde trabalha.
- O **Médico**, além do número de funcionário, é caracterizado pela sua/suas especialização/especializações. E ainda contém o ID do departamento onde trabalha.
- O **Rececionista** é caracterizado pelo seu número de identificação (ID) que herda da entidade Funcionário e do número de Balção onde está a registar o Paciente.
- O **Departamento** é caracterizado pelo seu nome, ID e o Médico que o supervisiona (nomeSupervisor). Ainda contém a identificação dos médicos que lá trabalham.
- A **CamaHospital** caracteriza-se pelo seu ID e pela camaOcupada ( verifica se está disponível ou não).
- O **quartoHospital** é caracterizado pelo ID e quantidade de camas que contém, que influência o número de paciente que consegue aceitar.
- O **Paciente** é caracterizado pelo seu número de Saúde, nome (primeiro e último), data de nascimento, género, morada, número de telefone. Caso esteja internado contém informação sobre o ID do quarto, data de entrada e saída do internamento. E ainda contém o ID do rececionista que o registou no hospital.
- A **Consulta** é caracterizada pela hora, data e número de Consulta. A Consulta é dada por um médico, que por sua vez, pode prescrever vários fármacos. Se houver consulta o médico é identificado com o seu ID de funcionário, tal como o utente com o seu número de saúde.
- O **Fármaco** é caracterizado pelo seu código, quantidade receitada, custo e nome.
- A **Cirurgia** é caracterizada pelo tipo de cirurgia, data e hora. A Cirurgia é composta por uma **equipa médica** que opera num **Bloco Operatório**.
- A **Equipa Médica** caracteriza-se pela quantidade de enfermeiros e médicos, mas principalmente pelo nome do Cirurgião Chefe.
- O **bloco Operatório** é caracterizado pelo número do Bloco e a Utilização (em uso ou não).

**A versão final do diagrama entidade/Relacionamento é o seguinte:**



# Esquema Relacional - versão final



## **Diretório SQL scripts**

Todo o código desenvolvido está inserido na pasta SQL\_scripts. Esta pasta contém os seguintes ficheiros: dropTables.sql, Tables.sql, insertValues.sql, stored\_procedures.sql, UDFs.sql e triggers.sql.

### **Drop**

O ficheiro dropTables.sql contém código sql que permite a eliminação das tabelas construídas na nossa base de dados.

### **Tables**

O ficheiro Tables.sql contém código sql que permite a criação das tabelas construídas na nossa base de dados.

### **Insert**

O ficheiro insertValues.sql contém código sql que permite a inserção de valores nas tabelas criadas na nossa base de dados.

### **Stored Procedures**

O ficheiro stored\_procedures.sql contém código sql que permite a criação de todos os procedures usados no nosso projeto.

Stored Procedure	Descrição
sp_registoPaciente	Registrar um Paciente (Internado ou não internado). Recebe como parâmetros todos os dados necessários do paciente. Se o parâmetro ID quarto for NULL, então significa que o paciente não será internado. Caso o parâmetro ID quarto estiver preenchido, então o paciente será internado. Quando o paciente é internado o stored procedure tem em conta se o quarto inserido está cheio, se não estiver então o paciente é atribuído a um quarto e existe uma atualização na lista de camas do hospital tal que uma cama passa a estar ocupada.
sp_eliminarPaciente	Eliminar um Paciente do sistema. Recebe como parâmetro o número de utente de saúde do respetiva paciente a eliminar. Se o Paciente estiver internado, o stored procedure atualiza uma cama do quarto do paciente para o estado livre.
sp_updateInfoPaciente	Atualiza os dados de um Paciente do sistema. Recebe como parâmetro todos os atributos do paciente do DER. A atualização é realizada em função do número de utente de saúde. Nesta situação é implementado com Transaction, pois vários UPDATES são feitos ao longo do stored procedure.
UpdatePaciente_NaoInternado	Atualiza a lista de pacientes internados. O paciente dessa lista deixa de estar internado e é eliminado da lista. Recebe como parâmetros o

	número de utente de saúde, para identificar o utente, e o ID do quarto que serve para atualizar os dados do quarto (uma cama fica livre). Nesta situação é implementado com Transaction, pois são efetuados UPDATES ao longo do stored procedure.
UpdatePaciente_Internado	Atualiza a lista de pacientes internados. Um paciente é internado no hospital e dessa forma adicionado a essa lista. Recebe como parâmetros o número de utente de saúde, para a identificação do utente, a data de entrada e saída do internamento e o ID do quarto que serve para atualizar os dados do quarto (uma cama fica ocupada). Nesta situação é implementado com Transaction, pois são efetuados UPDATES ao longo do stored procedure.
UpdateDadosPaciente_Internado	Atualiza os dados de um paciente da lista de pacientes internados. Recebe como parâmetros o número de utente de saúde do paciente, para a identificação do utente, e data de entrada e saída para serem atualizadas. Nesta situação é implementado com Transaction, pois são efetuados UPDATES ao longo do stored procedure.
sp_addEnfermeiros	Adiciona um enfermeiro à lista de enfermeiros e supervisionam uma determinada cama ocupada do hospital.
sp_removeEnfermeiros	Remove um enfermeiro da lista de enfermeiros supervisores de uma cama do hospital
sp_registoFuncionario	Registrar um Funcionário (Rececionista, Enfermeiro ou Médico). Recebe como parâmetros todos os dados necessários do funcionário. Se estiver preenchido o parâmetro deptID será um Médico ou Enfermeiro e o parâmetro balcão terá de estar a NULL, no caso de o funcionário ser um rececionista temos o inverso. O Store Procedure tem ainda o cuidado de não deixar adicionar um funcionário com um ID já existente.
sp_eliminarFuncionario	Eliminar um funcionário do sistema. Recebe como parâmetro o ID do funcionário a eliminar. Consoante o tipo de funcionário que seja, que podemos descobrir a partir do tuplo tipo, teremos que, no caso do Médico, eliminar as suas consultas e cirurgias, no caso do Enfermeiro elimina as suas cirurgias, assim como as camas que supervisiona e pacientes que auxilia, no caso do Rececionista não existe nenhum cuidado especial a ter.
sp_updateInfoFuncionario	Atualiza os dados de um funcionário do sistema. Recebe como parâmetro todos os atributos do funcionário do DER. A atualização é realizada a partir de quaisquer campos que sejam alterados, excepto o tipo. Nesta situação é implementado com Transaction, pois vários UPDATES são feitos ao longo do stored procedure.
schedule_surgery	Insere uma cirurgia. Recebe como parâmetro os valores de uma cirurgia. Verifica a sua data, a equipa médica atribuída, assim como o bloco operatório e paciente.
delete_apointment	Elimina uma consulta. Recebe como parâmetro o número de Utente de Saúde e número da consulta. Verifica o paciente e apaga os registos de uma possível prescrição, para poder apagar o registo da consulta.
delete_surgery	Elimina uma cirurgia. Recebe como parâmetro os valores de uma cirurgia. Faz uma verificação da data.

## UDFs

O ficheiro UDFs.sql contém código sql que permite a criação de todas as UDFs usadas no nosso projeto.

UDFs	Descrição
getPacienteBy_NoUtenteSaude	Devolve Pacientes em função do seu número de Utente de Saúde.
getPacienteBy_IDQuarto	Devolve Pacientes Internados em função do número de identificação do quarto onde se encontram.
getCamaHospitalBy_IDQuarto	Devolve a lista de camas e o seu estado em função da identificação do quarto.
getEnfSupervisorBy_IDQuarto	Devolve a lista de enfermeiros supervisores e a cama que estão a supervisionar em função da identificação do quarto.
getDepartBy_Name	Devolve um departamento em função do seu nome.
getEnfermeiros_naoS	Devolve a lista de enfermeiros não supervisores de uma determinada cama de hospital.
getFuncByID	Devolve um funcionário em função do seu ID.
getConsulta	Devolve uma Consulta em função do número de Utente de Saúde.
getCirurgia	Devolve uma Cirurgia em função do número de Utente de Saúde.

## Triggers

O ficheiro triggers.sql contém três triggers um que é acionado quando se faz uma inserção nas cirurgias. Este não permite que um paciente tenha mais que uma cirurgia do mesmo tipo, no mesmo dia. Com um cursor que é criado, pelo número de Utente de Saúde, data da Cirurgia e tipo de Cirurgia, fazemos a comparação do que está a ser inserido, se for, quer dizer está a tentar adicionar uma cirurgia no mesmo paciente, no mesmo dia e à mesma parte corporal (não aconselhável), e portanto ocorre um erro e realiza o rollback para que não permita realizar os updates, caso contrário, se o loop while acabar podemos fazer a marcação da cirurgia (inserir na base de dados) do que está na tabela inserted.

O segundo trigger é acionado quando se insere valores na Equipa\_Medica, este, não permite que um médico seja supervisor em mais do que uma equipa, utilizamos um cursor para percorrer os valores inseridos na base de dados e, se já for supervisor de uma equipa fazemos os rollback da transaction.

O último trigger tem a intenção do primeiro, mas dirige-se desta vez às consultas, marcando uma consulta se o paciente já não tiver outra marcada na mesma data.



## Interface

Para a interface do projeto utilizamos todas as tabelas que foram criadas. A nossa interface é composta pelos seguintes formulários: Registo Paciente (registoPaciente.vb), informação Pacientes (infoPacientes.vb), informação quartos (infoQuartos.vb), informação departamentos (Departamentos.vb) e informação funcionários (infoFuncionario.vb).

### Registo Paciente

Funcionalidades:

- Registrar um utente não internado na base de dados do hospital.
- Registrar um utente internado na base de dados do hospital, sendo que esta funcionalidade implica atribuir um quarto ao utente verificando se esse quarto está completo ou não.



O formulário, intitulado "Registo Paciente", contém os seguintes campos de entrada:

Número Utente de Saúde	Primeiro Nome	Último Nome
<input type="text"/>	<input type="text"/>	<input type="text"/>
Data de Nascimento <input type="text" value="yyyy-mm-dd"/>	Número de telefone <input type="text"/>	Género <input type="text" value="F/M"/>
ID Quarto <input type="text"/>	Data de Entrada <input type="text"/>	Data de Saída <input type="text"/>
ID Rececionista <input type="text"/>		

Um botão "Registrar Paciente" está localizado na base do formulário.

### Informação Pacientes

Funcionalidades:

- Eliminar um Utente não internado da base de dados do sistema.
- Eliminar um Utente internado e, conseqüentemente libertar uma cama do quarto onde esse paciente estava atribuído.
- Atualizar informação de um Paciente. Os dados que podem ser atualizados podem ser: o primeiro e último nome, data de nascimento, número de telefone, género, ID rececionista, ID quarto, data de entrada e saída. No caso do ID quarto, se ele for preenchido então poderá ser para internar um paciente e se for vazio poderá ser para o paciente deixar de estar internado.
- Pesquisar um paciente em função do seu número de utente de saúde.

Hospital Management System

MENU

Registo Paciente

Informação Pacientes

Informação Quartos

Departamentos

Informação Funcionários

Consultas

Cirurgias

Informação Pacientes

Insira o número de Utente de Saúde do Paciente

44445533

Joao Sousa

Mario Silva

Vitor Martins

Maria Fernandes

Pedro Gomes

Carlos Silva

Mariana Amaral

Catátina Costa

Andre Rosa

Afonso Pereira

Número Utente de Saúde

44445533

Primeiro Nome

Mariana

Último Nome

Amaral

Data de Nascimento

02/05/1977

Número de telefone

924444442

Género

F

ID Quarto

5

ID Rececionista

1

Data de Entrada

17/06/2022

Data de Saída

Atualizar

Eliminar

## Informação dos Quartos

Funcionalidades:

- Aceder à lista de Pacientes internados.
- Na lista de Pacientes internados é possível adicionar um paciente e consequentemente atribuir-lhe um quarto com cama livre.
- Na lista de Pacientes internados é possível eliminar um paciente e consequentemente libertar a cama ocupada.
- Na lista de Pacientes internados é possível atualizar os campos de data de entrada e saída.
- Aceder à lista de camas de um determinado quarto, o que dá para verificar, por exemplo, se a cama que está a ser atribuída encontra-se disponível.
- Aceder à lista de enfermeiros supervisores de um determinado quarto.

Hospital Management System

MENU

Registo Paciente

Informação Pacientes

Informação Quartos

Departamentos

Informação Funcionários

Consultas

Cirurgias

Informação Quartos

Insira a identificação do quarto

5

Quantidade de Camas por quarto

Enfermeiros Não Supervisores

Lista de Pacientes

Número Utente de Saúde	Nome	Data de Entrada	Data de Saída
44445533	Mariana Amaral	17/06/2022	
66654311	Catátina Costa	17/06/2022	

Lista de Camas

Identificação	Estado
A5	Ocupada
B5	Ocupada

Lista de Enfermeiros Supervisores

Identificação	ID Enfermeiro Supervisor
A5	10
B5	11

Número Utente de Saúde

44445533

Primeiro Nome

Mariana

Último Nome

Amaral

Data de Entrada

17/06/2022

Data de Saída

Adicionar Paciente

Editar dados

Eliminar registo paciente

Insira o ID do Enfermeiro Supervisor

Insira o ID da Cama

Adicionar

Remover

## Informação de Departamentos

Funcionalidades:

- Aceder à lista de departamentos e gestores do mesmo.

The screenshot shows the 'Hospital Management System' interface. On the left is a 'MENU' with buttons for 'Registo Paciente', 'Informação Pacientes', 'Informação Quartos', 'Departamentos' (highlighted with a blue border), 'Informação Funcionários', 'Consultas', and 'Cirurgias'. On the right is the 'Informação Departamentos' form, which contains a table with the following data:

Departamento	ID Departamento	Médico Supervisor
Emergência	1	9
Cardiologia	2	4
Terapia intensiva	3	13
Neurologia	4	5
Oncologia	5	
Obstetria	6	
Patologia	7	8
Radiologia	8	14

Neste formulário apenas se efetuou uma extração de dados a partir do visual studio:

```
CMD.CommandType = CommandType.Text  
CMD.CommandText = "SELECT * FROM Departamento"  
CN.Open()
```

## Informação dos Funcionários

Funcionalidades:

- Aceder à lista de Funcionários existentes no Hospital.
- Na lista de Funcionários é possível adicionar um Funcionário, quer seja Médico, Enfermeiro ou Rececionista.
- Na lista de Funcionários é possível eliminar um Funcionário, consequentemente eliminando as Cirurgias se for Médico ou Enfermeiro e as Consultas do Médico eliminado.
- Na lista de Funcionários é possível atualizar qualquer parâmetro à exceção do tipo de funcionário que é.

Insira um ID de um Funcionário

Miguel	Ferreira
Maria	Sousa
Juliana	Gomes
Joao	Pereira
Ana	Silva
Mariana	Costa
Ricardo	Amaral
Manuel	Santos
Silvia	Pombo
Catarina	Martins
Diana	Rosa
Adriano	Santos
Alice	Silva
Andre	Fernandes
Eduardo	Fernades

ID do Funcionario

Primeiro Nome

Segundo Nome

Data de Nascimento

Genero

Morada

Telefone

Email

Salario

Tipo

**Nota:** O formulário Funcionário não está completamente funcional, no entanto o código desenvolvido para este no visual studio está disponível. Além disso para o formulário Funcionário foram desenvolvidas SPs, UDFs que estão funcionais.

Dois formulários não foram desenvolvidos: o formulário Cirurgia, Consulta e Equipa Médica.

Apesar de não terem sido construídos, para o caso da Cirurgia foi realizado um Trigger também funcional que permite que um paciente não possa fazer mais que uma cirurgia do mesmo tipo no mesmo dia. Para a Consulta foi desenvolvido um trigger que não deixa que o mesmo paciente tenha mais que uma consulta no mesmo dia. Para a Equipa\_Medica foi criado um trigger que só permite que um médico seja supervisor de no máximo uma equipa.

Para efetuar a conexão à base de dados, o código aparece sempre em cima do ficheiro:

```

' Connection to database Hospital Management System
0 referências
Private Sub InfoPatients_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'LAPTOP-4HV6V7EN\SQLEXPRESS
    ' tcp:mednat.ieeta.pt\SQLSERVER,8101
    CN = New SqlConnection("data source=tcp:mednat.ieeta.pt\SQLSERVER,8101;integrated security=true;initial catalog=Hospital_Ma
    CMD = New SqlCommand
    CMD.Connection = CN
End Sub

```