



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

## SERVIÇO DE URGÊNCIA

Relatório da base de dados

### **Grupo 504**

Diogo Serra Duque [up201406274@fe.up.pt](mailto:up201406274@fe.up.pt)

José Aleixo Cruz [up201403526@fe.up.pt](mailto:up201403526@fe.up.pt)

Renato Sampaio de Abreu [up201403377@fe.up.pt](mailto:up201403377@fe.up.pt)

## Índice

Descrição do contexto .....	2
Principais conceitos .....	2
Diagrama de classes UML.....	4
Modelo relacional.....	5
Instruções de Linguagem de Definição de Dados SQL .....	6
Instruções de Linguagem de Manipulação de Dados SQL.....	9

## Descrição do contexto

A base de dados implementada terá como objetivo a gestão de um serviço de urgência de um hospital. Para isso, decidimos basearmo-nos no funcionamento (de forma muito simplificada) do serviço de urgência do Hospital Santo António.

Assim, no contexto da nossa base de dados, o serviço de urgência é composto por uma equipa de urgência que inclui médicos, enfermeiros e pessoal administrativo. Esta equipa tratará de todos os aspetos referentes ao utente.

Primeiramente é realizada a admissão nas urgências, que está a cargo do pessoal administrativo, e respetiva identificação da urgência médica do utente. Após isto, e de forma a representar o real funcionamento deste serviço, é feita a triagem de Manchester, pelos enfermeiros, atribuindo assim uma prioridade (emergente, muito urgente, urgente, pouco urgente e não urgente) e uma área de ação (médica, cirúrgica, ortopedia, clínica geral), conforme a condição atual do paciente. Com isto o utente estará sempre associado a uma urgência médica desde a admissão inicial, a qual será atendida, mais ou menos rapidamente, de acordo com os princípios da triagem.

Desta forma, e de acordo com a informação das urgências médicas existentes, os utentes associados a estas receberão o diagnóstico conforme a sua prioridade e por um médico que possa exercer funções na área de ação. Após o diagnóstico, e dependendo deste, é realizado o tratamento (cirurgia em casos mais graves, prescrições nos restantes), o qual está a cargo do médico. Uma vez que todos os tratamentos tenham sido realizados, o utente é autorizado, pelo médico, a ter alta.

## Principais conceitos

Para gestão dos recursos humanos, há uma superclasse *Pessoa* caracterizada pela idade, morada, nome, sexo e nif. Esta terá as seguintes classes derivadas, que herdam os atributos anteriormente referidos:

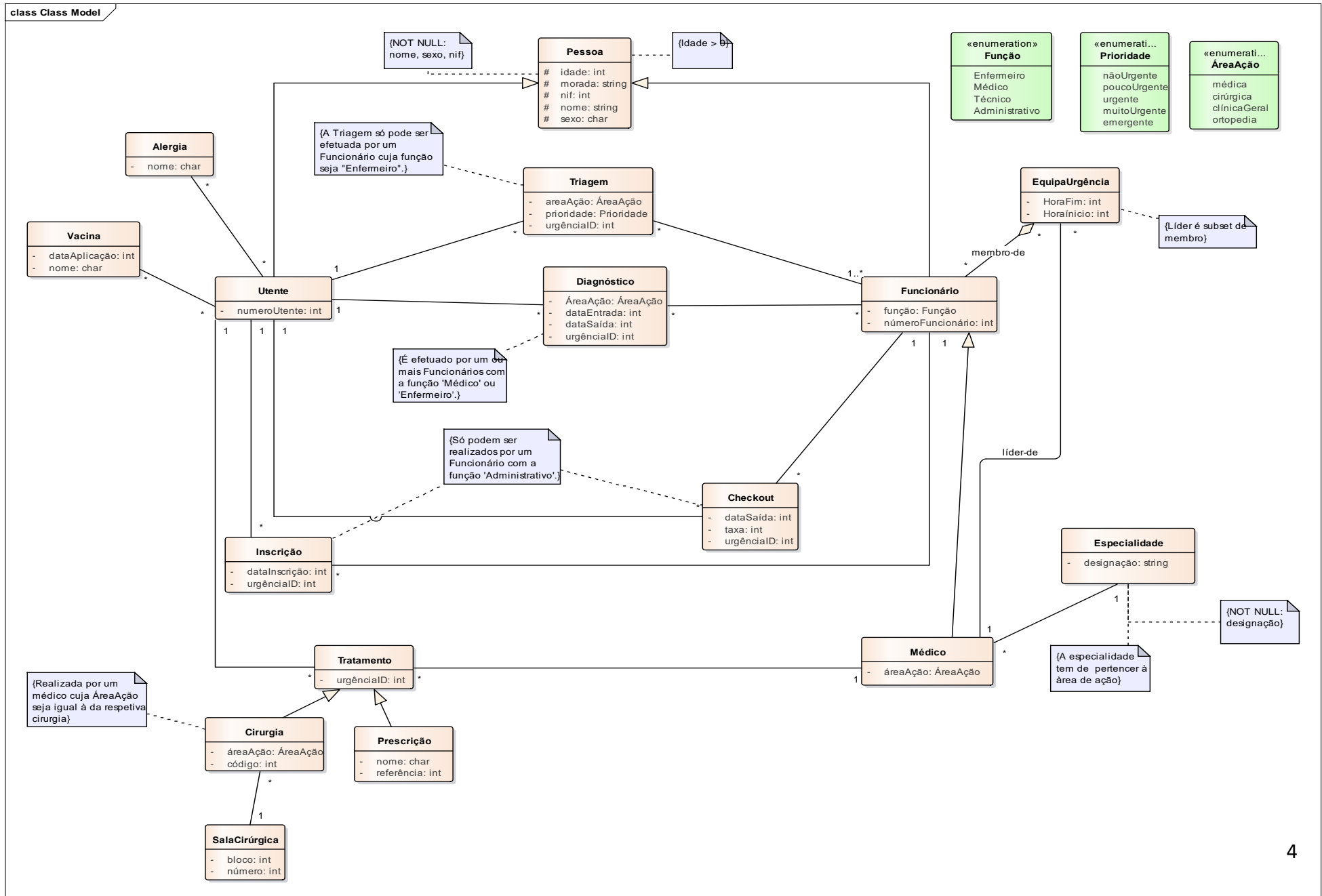
- *Utente*, definido pelo seu registo médico (alergias e vacinação) e pelo seu número de utente. Esta classe estará sempre associada aos vários procedimentos existentes numa urgência médica real (inscrição, diagnóstico, tratamento, etc), os quais estarão a cargo da equipa de urgência.
- *Funcionário* definido por um ID (identificação única dentro do hospital) e pela sua função (médico, enfermeiro, técnico, administrativo):
  - O funcionário com função Administrativo é responsável por proceder à Inscrição do Utente, isto é, o registo da hora de entrada e a identificação única da urgência médica em questão bem como Checkout do utente (pagamento de taxas e restantes burocracias).
  - O enfermeiro, tem como funções aplicar a *Triagem* a cada urgência médica. Isto implica definir a área de ação da urgência consoante a condição do utente. Para além disso, também é definida a *Prioridade* e o tempo-alvo para ser atendido.

- O médico que é caracterizado pela sua área de ação e consequentemente pela especialidade. Este faz parte da *EquipaUrgência*, podendo ser o líder da equipa e/ou fazer parte, e a sua função é, após a inscrição e triagem, realizar o *Diagnóstico*, o qual pode ser auxiliado por enfermeiros. O diagnóstico permite então saber o *Tratamento* necessário para cada urgência médica, o qual também é efetuado pelo médico.

Em relação à gestão dos recursos logísticos existem algumas classes que foram mencionadas acima mas não devidamente explicadas:

- *EquipaUrgência*, corresponde à agregação das várias instâncias de *Funcionário*, constituindo assim a equipa de urgência existente num determinado momento no hospital, liderada por um médico.
- *Diagnóstico*, que tem como objetivo determinar o Tratamento que se deve aplicar ao caso médico em questão sendo este realizado por um médico com o auxílio, se necessário de enfermeiros. Por seu lado, o *Tratamento* pode ser uma *Cirurgia* ou uma *Prescrição* de fármacos. A *Cirurgia* é sempre realizada por um médico cuja área de ação seja idêntica à área de ação da urgência médica e que tenha a especialidade necessária para realizar essa cirurgia.

# Diagrama de classes UML



## Modelo relacional

No modelo relacional da nossa base de dados, além de traduzirmos as associações do diagrama de classes em relações, decidimos interligar objetos relacionados à mesma urgência, usando “*urgênciaID*” como atributo de certas relações a de que uma urgência está dependente. Desta forma, escusamos a criação de uma nova classe e podemos aceder a todos os objetos relacionados com uma determinada urgência, realizando uma pesquisa restrita a um ID.

O texto abaixo representa o nosso modelo relacional, sendo que os atributos a negrito são as chaves primárias de cada relação.

- *Pessoa* (**idPessoa**, nome, idade, sexo, morada, nif)
- *Utente* (**númeroUtente**, idPessoa → Pessoa)
- *EquipaUrgência* (**idEquipa**, dataInício, dataFim, idLíder → Médico)
- *Funcionário* (**númeroFuncionário**, idPessoa → Pessoa, função, idEquipa → Equipa)
- *Médico* (**númeroFuncionário** → **Funcionário**, idEspecialidade → Especialidade, áreaAção)
- *Alergia* (**idAlergia**, nome)
- *Vacina* (**idVacina**, nome, dataAplicação)
- *Especialidade* (**idEspecialidade**, designação)
- *Inscrição* (**urgênciaID**, númeroUtente → Utente, númeroFuncionário → Funcionário, horaEntrada)
- *Checkout* (**urgênciaID**, númeroUtente → Utente, númeroFuncionário → Funcionário, taxa, horaSaída)
- *Triagem* (**urgênciaID**, númeroUtente → Utente, númeroFuncionário → Funcionário, áreaAção, prioridade)
- *Diagnóstico* (**urgênciaID**, **dataEntrada**, **dataSaída**, númeroUtente → Utente, númeroFuncionário → Funcionário, áreaAção)
- *Tratamento* (**urgênciaID**, númeroUtente → Utente, númeroFuncionário → Funcionário)
- *Cirurgia* (**código**, **urgênciaID**, ÁreaAção, idSalaCirúrgica → SalaCirúrgica)
- *Prescrição* (**referência**, **urgênciaID**, nome)
- *SalaCirúrgica* (**idSalaCirúrgica**, bloco, número)
- *UtenteAlergia* (**idAlergia** → **Alergia**, **númeroUtente** → **Utente**)
- *UtenteVacina* (**idVacina** → **Vacina**, **númeroUtente** → **Utente**)

## Instruções de Linguagem de Definição de Dados SQL

De seguida encontram-se explícitas as instruções que usamos para criar a estrutura da nossa base de dados, gerando as tabelas necessárias, com os respetivos atributos.

```
CREATE TABLE Pessoa (
  idPessoa INT NOT NULL PRIMARY KEY,
  nome CHAR(30) NOT NULL,
  idade INT NOT NULL,
  sexo CHAR(20) NOT NULL,
  morada VARCHAR(50),
  NIF INT NOT NULL UNIQUE);

CREATE TABLE EquipaUrgencia (
  idEquipa INT NOT NULL PRIMARY KEY,
  dataInicio DATE NOT NULL,
  dataFim DATE,
  idLider INT NOT NULL,
  FOREIGN KEY (idLider) REFERENCES Medico(idLider));

CREATE TABLE Utente (
  numeroUtente INT NOT NULL PRIMARY KEY,
  idPessoa INT NOT NULL,
  FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa));

CREATE TABLE Funcionario (
  numeroFuncionario INT NOT NULL PRIMARY KEY,
  idPessoa INT NOT NULL,
  funcao VARCHAR(50) NOT NULL,
  idEquipa INT,
  FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa),
  FOREIGN KEY (idEquipa) REFERENCES EquipaUrgencia(idEquipa));

CREATE TABLE Medico (
  numeroFuncionario INT NOT NULL PRIMARY KEY,
  idEspecialidade INT NOT NULL,
  areaAcao VARCHAR(50) NOT NULL,
  FOREIGN KEY (numeroFuncionario) REFERENCES Funcionario(numeroFuncionario),
  FOREIGN KEY (idEspecialidade) REFERENCES Especialidade(idEspecialidade));

CREATE TABLE Alergia (
  idAlergia INT NOT NULL PRIMARY KEY,
  nome VARCHAR(50) NOT NULL);

CREATE TABLE Vacina (
  idVacina INT NOT NULL PRIMARY KEY,
  nome VARCHAR(50) NOT NULL);

CREATE TABLE Especialidade (
  idEspecialidade INT NOT NULL PRIMARY KEY,
  designacao VARCHAR(50));

CREATE TABLE SalaCirurgica (
  idSalaCirurgica INT NOT NULL PRIMARY KEY,
  bloco CHAR(20) NOT NULL,
  numero INT NOT NULL);
```

```
CREATE TABLE UtenteAlergia(  
  idAlergia INT NOT NULL REFERENCES Alergia(idAlergia),  
  numeroUtente INT NOT NULL REFERENCES Utente(numeroUtente),  
  PRIMARY KEY(idAlergia, numeroUtente));  
  
CREATE TABLE UtenteVacina(  
  idVacina INT NOT NULL REFERENCES Vacina(idVacina),  
  numeroUtente INT NOT NULL REFERENCES Utente(numeroUtente),  
  dataAplicacao DATE,  
  PRIMARY KEY(idVacina, numeroUtente, dataAplicacao));  
  
CREATE TABLE EquipaUrgencia(  
  idEquipa INT NOT NULL PRIMARY KEY,  
  dataInicio DATE NOT NULL,  
  dataFim DATE,  
  idLider INT NOT NULL,  
  FOREIGN KEY (idLider) REFERENCES Medico(idLider));  
  
CREATE TABLE Inscricao  
  (urgenciaID INT PRIMARY KEY NOT NULL,  
  numeroUtente INT REFERENCES Utente(numeroUtente) NOT NULL,  
  numeroFuncionario INT REFERENCES Funcionario(numeroFuncionario) NOT  
  NULL,  
  horaEntrada INT NOT NULL);  
  
CREATE TABLE Checkout  
  (urgenciaID INT PRIMARY KEY NOT NULL,  
  numeroUtente INT REFERENCES Utente(numeroUtente) NOT NULL,  
  numeroFuncionario INT REFERENCES Funcionario(numeroFuncionario) NOT  
  NULL,  
  taxa INT NOT NULL,  
  dataSaida INT NOT NULL);  
  
CREATE TABLE Triagem  
  (urgenciaID INT PRIMARY KEY NOT NULL,  
  numeroUtente INT REFERENCES Utente(numeroUtente) NOT NULL,  
  numeroFuncionario INT REFERENCES Funcionario(numeroFuncionario) NOT  
  NULL,  
  areaAcao TEXT NOT NULL,  
  prioridade TEXT NOT NULL);  
  
CREATE TABLE Diagnostico  
  (urgenciaID INT NOT NULL,  
  numeroUtente INT REFERENCES Utente(numeroUtente) NOT NULL,  
  numeroFuncionario INT REFERENCES Funcionario(numeroFuncionario) NOT  
  NULL,  
  dataEntrada INT NOT NULL,  
  dataSaida INT,  
  PRIMARY KEY(urgenciaID, dataEntrada, dataSaida),  
  CHECK (dataSaida > dataEntrada));  
  
CREATE TABLE Tratamento  
  (urgenciaID INT PRIMARY KEY NOT NULL,  
  numeroUtente INT REFERENCES Utente(numeroUtente) NOT NULL,  
  numeroFuncionario INT REFERENCES Funcionario(numeroFuncionario) NOT  
  NULL);
```



```
CREATE TABLE Cirurgia(  
urgenciaID INT NOT NULL REFERENCES Tratamento(urgenciaID),  
codigo INT NOT NULL UNIQUE,  
areaAcao TEXT NOT NULL,  
idSalaCirurgica INT NOT NULL UNIQUE REFERENCES  
SalaCirurgica(idSalaCirurgica),  
PRIMARY KEY(urgenciaID, codigo));
```

```
CREATE TABLE Prescricao(  
urgenciaID INT NOT NULL REFERENCES Tratamento(urgenciaID),  
referencia INT NOT NULL,  
nome CHAR(20) NOT NULL,  
PRIMARY KEY (urgenciaID, referencia));
```

## Instruções de Linguagem de Manipulação de Dados SQL

Os próximos comandos de SQL são responsáveis por inserir objetos de exemplo nas tabelas anteriormente geradas.

```
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, morada, NIF) VALUES
(1, 'Renato Abreu', 19, 'Masculino', 'Esposende', 123456789);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, morada, NIF) VALUES
(2, 'Diogo Duque', 19, 'Masculino', 'Trofa', 234567891);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, morada, NIF) VALUES
(3, 'José Aleixo', 19, 'Masculino', 'Mortosa', 345678912);

INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (4, 'Nuno
Vieira', 27, 'Masculino', 456789123);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (5, 'Pedro
Filipe', 57, 'Masculino', 567891234);

INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (6,
'Fernando', 27, 'Masculino', 789123456);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, morada, NIF) VALUES
(7, 'Elias', 43, 'Masculino', 'Lisboa', 891234567);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (8,
'Hugo', 57, 'Masculino', 912345678);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, morada, NIF) VALUES
(9, 'Ines', 28, 'Feminino', 'Porto', 111111111);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (10,
'Joao', 42, 'Masculino', 222222222);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (11,
'Liliana', 27, 'Feminino', 333333333);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (12,
'Manuel', 43, 'Masculino', 444444444);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (13,
'Nuno', 57, 'Masculino', 555555555);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, morada, NIF) VALUES
(14, 'Otavio', 28, 'Masculino', 'Porto', 666666666);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (15,
'Paulo', 42, 'Masculino', 777777777);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, NIF) VALUES (16,
'Tania', 28, 'Feminino', 888888888);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, morada, NIF) VALUES
(17, 'Daniel', 28, 'Masculino', 'Esposende', 999999999);
INSERT INTO Pessoa(idPessoa, nome, idade, sexo, morada, NIF) VALUES
(18, 'Daniela Silva', 28, 'Feminino', 'Esposende', 111222333);

INSERT INTO Utente(numeroUtente, idPessoa) VALUES (1000, 1);
INSERT INTO Utente(numeroUtente, idPessoa) VALUES (1001, 2);
INSERT INTO Utente(numeroUtente, idPessoa) VALUES (1002, 3);
INSERT INTO Utente(numeroUtente, idPessoa) VALUES (1003, 4);
INSERT INTO Utente(numeroUtente, idPessoa) VALUES (1004, 5);

INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (500, 6, 'Enfermeiro', 1);
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (501, 7, 'Enfermeiro', 1);
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (502, 8, 'Administrativo', 1);
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (503, 9, 'Administrativo', 1);
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (504, 10, 'Tecnico', 1);
```

```
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao) VALUES
(505, 11, 'Administrativo');
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao) VALUES
(506, 12, 'Enfermeiro');
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (507, 13, 'Administrativo', 2);
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (508, 14, 'Enfermeiro', 2);
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (509, 15, 'Tecnico', 2);

INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (510, 16, 'Medico', 1);
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (511, 17, 'Medico', 2);
INSERT INTO Funcionario(numeroFuncionario, idPessoa, funcao, idEquipa)
VALUES (512, 18, 'Medico', 2);

INSERT INTO Medico(numeroFuncionario, idEspecialidade, areaAcao)
VALUES (510, 1, 'Ortopedia');
INSERT INTO Medico(numeroFuncionario, idEspecialidade, areaAcao)
VALUES (511, 2, 'Cirurgica');
INSERT INTO Medico(numeroFuncionario, idEspecialidade, areaAcao)
VALUES (512, 2, 'ClinicaGeral');

INSERT INTO Alergia VALUES(1, 'Polen');
INSERT INTO Alergia VALUES(2, 'Acaros');
INSERT INTO Vacina VALUES(1, 'Tetano');
INSERT INTO Vacina VALUES(2, 'Hepatite');

INSERT INTO Especialidade VALUES(1, 'Ortopedica');
INSERT INTO Especialidade VALUES(2, 'Neurocirurgiao');

INSERT INTO SalaCirurgica VALUES(2120,'B', 120);
INSERT INTO SalaCirurgica VALUES(3045,'C', 45);

INSERT INTO UtenteAlergia VALUES(1, 1001);
INSERT INTO UtenteAlergia VALUES(2, 1002);
INSERT INTO UtenteAlergia VALUES(2, 1001);
INSERT INTO UtenteVacina VALUES(1, 1004, '01-07-1996');
INSERT INTO UtenteVacina VALUES(1, 1003, '28-02-2002');
INSERT INTO UtenteVacina VALUES(2, 1000, '21-03-2001');

INSERT INTO EquipaUrgencia VALUES(1, '22-04-2016', '25-04-2016', 512);
INSERT INTO EquipaUrgencia VALUES(2, '26-04-2016', '28-04-2016', 510);

INSERT INTO Inscricao(urgenciaID, numeroUtente, numeroFuncionario,
horaEntrada) VALUES(1, 1000, 502, '21-04-2016');
INSERT INTO Inscricao(urgenciaID, numeroUtente, numeroFuncionario,
horaEntrada) VALUES(2, 1001, 502, '22-04-2016');
INSERT INTO Inscricao(urgenciaID, numeroUtente, numeroFuncionario,
horaEntrada) VALUES(3, 1003, 503, '24-04-2016');
INSERT INTO Inscricao(urgenciaID, numeroUtente, numeroFuncionario,
horaEntrada) VALUES(4, 1004, 507, '27-04-2016');
INSERT INTO Inscricao(urgenciaID, numeroUtente, numeroFuncionario,
horaEntrada) VALUES(5, 1001, 507, '28-04-2016');

INSERT INTO Checkout(urgenciaID, numeroUtente, numeroFuncionario,
dataSaida, taxa) VALUES(1, 1000, 502, '22-04-2016', 100);
```

```
INSERT INTO Checkout(urgenciaID, numeroUtente, numeroFuncionario,
dataSaida, taxa) VALUES(2, 1001, 503, '23-04-2016', 50);
INSERT INTO Checkout(urgenciaID, numeroUtente, numeroFuncionario,
dataSaida, taxa) VALUES(3, 1003, 502, '25-04-2016', 200);
INSERT INTO Checkout(urgenciaID, numeroUtente, numeroFuncionario,
dataSaida, taxa) VALUES(4, 1004, 507, '28-04-2016', 25);
INSERT INTO Checkout(urgenciaID, numeroUtente, numeroFuncionario,
dataSaida, taxa) VALUES(5, 1001, 507, '28-04-2016', 15);

INSERT INTO Triagem(urgenciaID, numeroUtente, numeroFuncionario,
areaAcao, prioridade) VALUES(1, 1000, 501, 'Medica', 'poucoUrgente');
INSERT INTO Triagem(urgenciaID, numeroUtente, numeroFuncionario,
areaAcao, prioridade) VALUES(2, 1001, 500, 'Cirurgica', 'emergente');
INSERT INTO Triagem(urgenciaID, numeroUtente, numeroFuncionario,
areaAcao, prioridade) VALUES(3, 1003, 501, 'ClinicaGeral',
'poucoUrgente');
INSERT INTO Triagem(urgenciaID, numeroUtente, numeroFuncionario,
areaAcao, prioridade) VALUES(4, 1004, 508, 'Medica', 'muitoUrgente');
INSERT INTO Triagem(urgenciaID, numeroUtente, numeroFuncionario,
areaAcao, prioridade) VALUES(5, 1001, 508, 'Ortopedia', 'urgente');

INSERT INTO Cirurgia(urgenciaID, codigo, areaAcao, idSalaCirurgica)
VALUES(2,3, 'Cirurgica', 2120);

INSERT INTO Prescricao(referencia, urgenciaID, nome)
VALUES(12349,1, 'Ben-u-ron');
INSERT INTO Prescricao(referencia, urgenciaID, nome) VALUES(13938, 3,
'Brufen');
```