

Farmácia Hospitalar



Grupo 404:

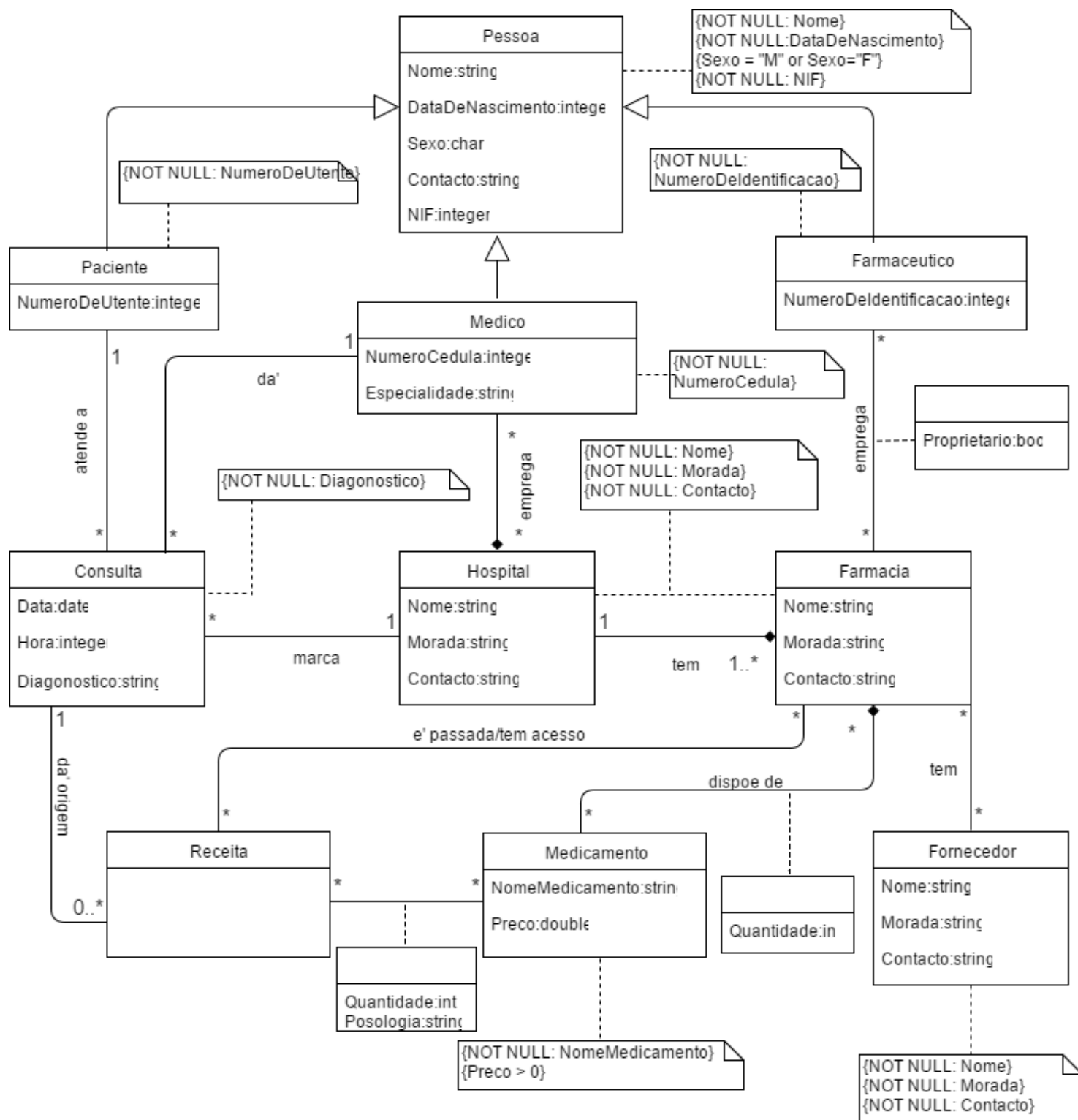
Pedro Santos – 201200764

Pedro Ferreira – 201103084

Vinicius Ferretti - 201600721

A. Definição do Modelo Conceptual

Um Hospital (Nome,Morada,Contacto) tem ao seu serviço um conjunto de médicos e serve um conjunto de pacientes. Ambos estão associados a uma classe Pessoa, descrita pelos parâmetros Nome, DataDeNascimento, sexo, contacto, NIF. Tanto o médico (NumCedulaProfissional,Especialidade) como o Paciente (NumDeUtente) fazem parte da consulta, que é caracterizada por uma data e um diagnóstico de consulta. Pretende-se também guardar as receitas de medicamentos feitas pelos médicos aos pacientes durante as consultas. Dependendo então do diagnóstico de consulta, o médico, pode ou não emitir de seguida uma prescrição ao paciente na qual vai estar presente o nome do medicamento a ser adquirido na farmácia do hospital, bem como a quantidade de embalagens prescritas e o laboratório onde estas foram produzidas. Estas receitas de medicamentos podem então no fim da consulta ser levantadas na farmácia do hospital presente. A Farmácia à qual o paciente se vai dirigir é caracterizada por um nome, morada e contacto . À farmácia estão associadas pessoas, nomeadamente farmacêuticos e fornecedores. Do Farmacêutico é necessário guardar as informações acerca do seu número de identificação, bem como saber se este é o proprietário ou não da farmácia. Do Fornecedor pretende-se saber o seu nome, morada e contacto. A esta farmácia está também associado um Stock de fármacos da qual é importante armazenar o nome do medicamento, a quantidade presente na farmácia bem como o seu preço.



B. Definição do Esquema Relacional

Paciente(pacienteID, Nome, DataDeNascimento, Sexo, Contato, NIF, NumeroDeUtente)

Medico(medicoID, Nome, DataDeNascimento, Sexo, Contato, NIF, NumeroCedula, Especialidade)

Farmaceutico(farmaceuticoID, Nome, DataDeNascimento, Sexo, Contato, NIF, NumeroDeIdentificacao)

Consulta(consultaID, dataconsulta, Diagnostico, pacienteID → Paciente, medicoID → medico, hospitalID → hospital)

Hospital(hospitalID, Nome, Morada, Contacto)

Farmacia(farmaciaID, Nome, Morada, Contacto, hospitalID → Hospital)

Receita(receitaID, Posologia, consultaID → consulta)

Medicamento(medicamentoID, NomeMedicamento, Preco)

Fornecedor(fornecedorID, Nome, Morada, Contacto)

ReceitaMedicamento(receitaID → receita, medicamentoID → Medicamento, Quantidade)

ReceitaFarmacia (receitaID → Receita, farmaciaID → Farmacia)

MedicoHospital (medicoID → Medico , hospitalID → Hospital)

FarmaceuticoFarmacia (farmaceuticoID → Farmaceutico, farmaciaID → Farmacia, Proprietario)

MedicamentoFarmacia (medicamentoID → Medicamento, farmaciaID → Farmacia, Quantidade)

FornecedorFarmacia(fornecedorID → Fornecedor, famaciaID → Farmacia)

C. Análise de Dependências Funcionais e Formas Normais

#	NAME	CHAVES CANDIDATAS	DEPENDENCIAS FUNCIONAIS
R01	Paciente	pacienteID	pacienteID → nome, data de nascimento, sexo, contacto, nif, NumeroDeUtente
R02	Medico	medicoID	medicoID → nome, data de nascimento, sexo, contacto, nif, numero de cedula, especialidade
R03	Farmaceutico	farmaceuticoID	farmaceuticoID → nome, DataDeNascimento, sexo, contacto, nif, numero de identificacao
R04	Consulta	consultaID	ConsultaID → data de consulta, diagnostico, pacienteID, medicoID, hospitalID
R05	Hospital	hospitalID	hospitalID → nome, morada, contacto
R06	Farmacia	farmaciaID	farmaciaID → nome, morada, contacto, hospitalID
R07	Receita	receitaID	receitaID → posologia, consultaID
R08	Medicamento	medicamentoID	medicamentoID → nome do medicamento, preco
R09	Fornecedor	fornecedorID	fornecedorID → nome, morada, contacto
R10	ReceitaMedicamento	receita medicamentoID	receita medicamentoID → medicamentoID, quantidade
R11	ReceitaFarmacia	receita farmaciaID	receita farmaciaID → farmaciaID
R12	MedicoHospital	medico hospitalID	medico hospitalID → hospitalID
R13	FarmaceuticoFarmacia	farmaceutico farmaciaID	farmaceutico farmaciaID → farmaciaID, proprietário
R14	MedicamentoFarmacia	medicamento farmaciaID	medicamento farmaciaID → farmaciaID, quantidade
R15	FornecedorFarmacia	fornecedor farmaciaID	fornecedor farmaciaID → farmaciaID

D. Criação da base de dados em SQLite

Ficheiro 'criar.sql' entregue em anexo.

E. Adição de restrições à base de dados

(Restrições já adicionadas ao ficheiro 'criar.sql' entregue em anexo.)

paciente:

Restrição chave PRIMARY KEY: *pacienteid* (não pode haver dois pacientes com o mesmo id);

Restrição NOT NULL: *nome* (o paciente tem que ter um nome);

Restrição NOT NULL: *contacto* (ao paciente está associado um contacto);

Restrição chave UNIQUE: *nif* (Cada pessoa possui o seu próprio número de identificação fiscal);

Restrição chave UNIQUE: *numutente* (Cada paciente tem um número de utente que o identifica);

Restrição CHECK: *sex_char* (Verifica o sexo da pessoa: se recebe um 'M' para masculino ou 'F' para feminino);

medico:

Restrição chave PRIMARY KEY: *medicoid* (não pode haver dois médicos com o mesmo id);

Restrição NOT NULL: *nome* (o médico tem que ter um nome);

Restrição NOT NULL: *contacto* (Ao médico está associado um contacto);

Restrição chave UNIQUE: *nif* (Cada pessoa possui o seu próprio número de identificação fiscal);

Restrição chave UNIQUE: *numcedula* (Cada médico tem um número de cédula que o identifica);

Restrição CHECK: *sex_char* (Verifica o sexo da pessoa: se recebe um 'M' para masculino ou 'F' para feminino);

farmaceutico:

Restrição chave PRIMARY KEY: *farmaceuticoid* (não pode haver dois farmacêuticos com o mesmo id);

Restrição NOT NULL: *nome* (O Farmacêutico tem que ter um nome);

Restrição NOT NULL: *contacto* (Ao Farmacêutico está associado um contacto);

Restrição chave UNIQUE: *nif* (Cada pessoa possui o seu próprio número de identificação fiscal);

Restrição chave UNIQUE: *numidentificacao* (Cada farmacêutico possui o seu próprio número de identificação);

Restrição CHECK: *sex_char* (Verifica o sexo da pessoa: se recebe um 'M' para masculino ou 'F' para feminino);

hospital:

Restrição chave PRIMARY KEY: *hospitalid* (não pode haver dois hospitais com o mesmo id);

Restrição NOT NULL: *nome* (O Hospital tem um nome que o caracteriza);

Restrição NOT NULL: *contacto* (Ao Hospital está associado um contacto);

medicamento:

Restrição chave PRIMARY KEY: *medicamentoid* (não pode haver dois medicamentos com o mesmo id);

Restrição NOT NULL: *nomemedicamento* (O medicamento tem nome);

Restrição NOT NULL: *preco* (o preço não pode ser nulo);

fornecedor:

Restrição chave PRIMARY KEY: *fornecedor* (não pode haver dois fornecedores com o mesmo id);

Restrição NOT NULL: *nome* (O Fornecedor tem que ter um nome);

Restrição NOT NULL: *contacto* (O Fornecedor tem um contacto associado);

Restrição NOT NULL: *morada* (O Fornecedor tem uma morada associada);

consulta:

Restrição chave PRIMARY KEY: *consulta*id (não pode haver duas consultas com o mesmo id);

Restrição NOT NULL: *dataconsulta* (A data da consulta não pode ser inexistente);

Restrição chave FOREIGN KEY: "*FK_Consulta_Paciente*"

Restrição chave FOREIGN KEY: "*FK_Consulta_Medico*"

Restrição chave FOREIGN KEY: "*FK_Consulta_Hospital*"

receita:

Restrição chave PRIMARY KEY: *receita*id (não pode haver duas receitas com o mesmo id);

Restrição chave FOREIGN KEY: "*FK_Receita_Consulta*"

farmacia:

Restrição chave PRIMARY KEY: *farmacia*id (não pode haver duas farmácias com o mesmo id);

Restrição NOT NULL: *nome* (A Farmácia tem um nome que a identifica);

Restrição NOT NULL: *contacto* (A Farmácia tem um contacto associado)

Restrição chave FOREIGN KEY: "*FK_Farmacia_Hospital*"

receitamedicamento:

Restrição NOT NULL: *quantidade*

Restrição chave FOREIGN KEY: "*Receita*"

Restrição chave FOREIGN KEY: "*medicamento*"

receitafarmacia:

Restrição chave PRIMARY KEY: (*receita*id,*farmacia*id)

Restrição chave FOREIGN KEY: "*Receita*"

Restrição chave FOREIGN KEY: "*Farmacia*"

medicohospital:

Restrição chave PRIMARY KEY: (*medicoid*,*hospital*id)

Restrição chave FOREIGN KEY: "*Medico*"

Restrição chave FOREIGN KEY: "*Hospital*"

farmaceuticofarmacia:

Restrição chave PRIMARY KEY: (*farmaceuticoid,farmacicaid*)

Restrição chave FOREIGN KEY: "*Farmaceutico*"

Restrição chave FOREIGN KEY: "*Farmacia*"

medicamentofarmacia:

Restrição chave PRIMARY KEY: (*medicamentoid,farmacicaid*)

Restrição chave FOREIGN KEY: "*Medicamento*"

Restrição chave FOREIGN KEY: "*Farmacia*"

Restrição NOT NULL: *quantidade*

fornecedorfarmacia:

Restrição chave PRIMARY KEY: (*fornecedorid,farmacicaid*)

Restrição chave FOREIGN KEY: "*Fornecedor*"

Restrição chave FOREIGN KEY: "*Farmacia*"

F. Carregamento de dados

Ficheiro 'povoar.sql' entregue em anexo.

G. Interrogação da Base de dados

1.

Objectivo: Saber quais os pacientes atendidos por um determinado médico.

Sql:

```
SELECT paciente.nome FROM paciente WHERE pacienteid IN (  
SELECT pacienteid FROM consulta WHERE medicoid = 2 ); -- escolher médico
```

2.

Objectivo: Quais os 5 médicos que têm o maior número de consultas ?

Sql:

```
SELECT medico.medicoid, medico.nome, count(*) AS total FROM consulta  
INNER JOIN medico ON consulta.medicoid = medico.medicoid  
GROUP BY medico.medicoid  
ORDER BY total DESC  
LIMIT 5;
```

3.

Objectivo: Em qual farmácia se encontra determinado medicamento e qual a sua quantidade e preço

Sql:

```
SELECT medicamentofarmacia.farmacaid, farmacia.nome,  
medicamentofarmacia.quantidade, medicamento.preco  
FROM medicamentofarmacia  
INNER JOIN medicamento ON medicamentofarmacia.medicamentoid =  
medicamento.medicamentoid  
INNER JOIN farmacia ON medicamentofarmacia.farmacaid = farmacia.farmacaid  
WHERE medicamentofarmacia.medicamentoid = 11; -- escolher qual a farmacia
```

4.

Objectivo: Quais as consultas de um determinado médico .

Sql:

```
SELECT medico.medicoid, medico.nome, consulta.dataconsulta  
FROM consulta  
INNER JOIN medico ON consulta.medicoid = medico.medicoid  
WHERE medico.medicoid = 5 ; --ID do medico  
GROUP BY consulta.dataconsulta;
```

5.

Objectivo: Quais as consultas de um determinado médico nos próximos 7 dias e quais os pacientes de cada consulta.

Sql:

```
SELECT medico.medicoid, medico.nome AS medico, consulta.dataconsulta AS data,
paciente.nome AS paciente
FROM consulta
INNER JOIN medico ON consulta.medicoid = medico.medicoid
INNER JOIN paciente ON consulta.pacienteid = paciente.pacienteid
WHERE consulta.dataconsulta >= (SELECT datetime('now'))
AND consulta.dataconsulta <= (SELECT datetime('now','+7 days'))
ORDER BY consulta.dataconsulta ASC;
```

6.

Objectivo: Qual é o medicamento mais caro ?.

Sql:

```
SELECT medicamento.nomemedicamento AS medicamento,
MAX(medicamento.preco) AS preco
FROM medicamento;
```

7.

Objectivo: Saber o custo total uma receita x.

Sql:

```
SELECT SUM(preco*quantidade) FROM (SELECT
receitamedicamento.medicamentoid, medicamento.preco AS preco
FROM receitamedicamento
INNER JOIN medicamento
ON receitamedicamento.medicamentoid = medicamento.medicamentoid
WHERE receitamedicamento.receitaid = 2); -- numero da receita
```

8.

Objectivo: Lista dos medicamentos de uma receita.

Sql:

```
SELECT receitamedicamento.medicamentoid, medicamento.nomemedicamento AS
nome_medicamento
FROM receitamedicamento
INNER JOIN medicamento
ON receitamedicamento.medicamentoid = medicamento.medicamentoid
WHERE receitamedicamento.receitaid = 3; -- numero da receita
```

9.

Objectivo: Lista dos pacientes com mais de 18 anos

Sql:

```
SELECT nome 'nome', datanasc 'data de nascimento'  
FROM paciente  
WHERE datanasc >= '01/01/1999'  
ORDER BY datanasc ASC;
```

10.

Objectivo: Listagem dos fornecedores

Sql:

```
SELECT nome FROM fornecedor;
```

H. Adição de gatilhos à base de dados

1.

Objectivo: Sempre que forem inseridos dados na tabela receita medicamento, isto é, sempre que um medicamento for prescrito numa determinada receita, atualiza a quantidade do mesmo medicamento na farmácia onde vai ser recolhido vai ser .

Sql:

```
CREATE TRIGGER adiciona_medicamento
AFTER INSERT ON receitamedicamento
FOR EACH ROW
BEGIN
UPDATE medicamentofarmacia
SET quantidade = COALESCE(quantidade, 0) - NEW.quantidade
WHERE medicamentoid = NEW.medicamentoid;
END
```

2.

Objectivo: Se um paciente for apagado da base de dados, as suas consultas são removidas

Sql:

```
CREATE TRIGGER apagaConsulta
AFTER DELETE ON paciente
FOR EACH ROW
BEGIN
DELETE FROM consulta WHERE pacienteid = OLD.pacienteid;
END;
```

3. Objectivo: Verifica se a data da consulta marcada se encontra dentro dos limites possíveis, isto é, se a data não é anterior ao dia em que está a ser marcada.

Sql:

```
CREATE TRIGGER IF NOT EXISTS verificar_data
BEFORE INSERT ON Consulta
WHEN ((SELECT date('now')) > NEW.dataconsulta)
BEGIN
SELECT RAISE(ABORT, 'Nao podes adicionar uma consulta numa data anterior a atual');
END;
```