

Hospital Database (Parte 1)

Grupo 406

up201806538@fe.up.pt	Henrique Manuel Ruivo Pereira
up201801011@fe.up.pt	Iohan Xavier Sardinha Dutra Soares
up201806554@fe.up.pt	Telmo Alexandre Espirito Santo Baptista

30 de Março de 2020

Projeto BDAD - 2019/20 - MIEIC

Professora das Aulas Laboratorias: Carla Alexandra Teixeira Lopes



Índice

1	Contexto	3
2	UML - Modelo Conceptual	4
3	Modelo Relacional	5
3.1	Classes	5
3.2	Associações	6
3.3	Generalizações	6
4	Análise de FDs e FNs	7
4.1	Classes	7
4.2	Associações	7
5	Restrições	8

1 Contexto

É pretendido modelar uma base de dados para um hospital com diversos tipos de serviços disponíveis.

Sobre o próprio hospital, interessa guardar informação genérica, como o seu nome, localização completa (morada, código postal, localidade, etc), contato telefónico, bem como se o hospital é público ou privado.

O hospital é constituído por vários departamentos. Cada um destes tem nome, identificador, especializações e entidade responsável pelo departamento.

Staff, pacientes e médicos de família são pessoas, acerca das quais interessa saber o nome, o seu número de identificação único (cartão de cidadão ou equivalente), a sua morada completa (morada propriamente dita, código postal, localidade, etc), a(s) sua(s) nacionalidade(s), o seu contato telefónico, o seu número de beneficiário, o seu sexo e a sua data de nascimento.

A entidade responsável por um departamento é um membro da staff do hospital, podendo este tratar-se de um médico, enfermeiro ou técnico no hospital. Cada membro do staff tem o seu código identificador no hospital.

Especificamente sobre os médicos, é necessário guardar o número do seu consultório no hospital (caso tenha) e a sua especialização (caso tenha). Sobre os enfermeiros, apenas é necessário guardar a sua especialização, caso a tenha. Sobre os técnicos, apenas interessa guardar que serviço oferecem.

Sobre os médicos de família, interessa guardar o centro de saúde ao qual estão associados.

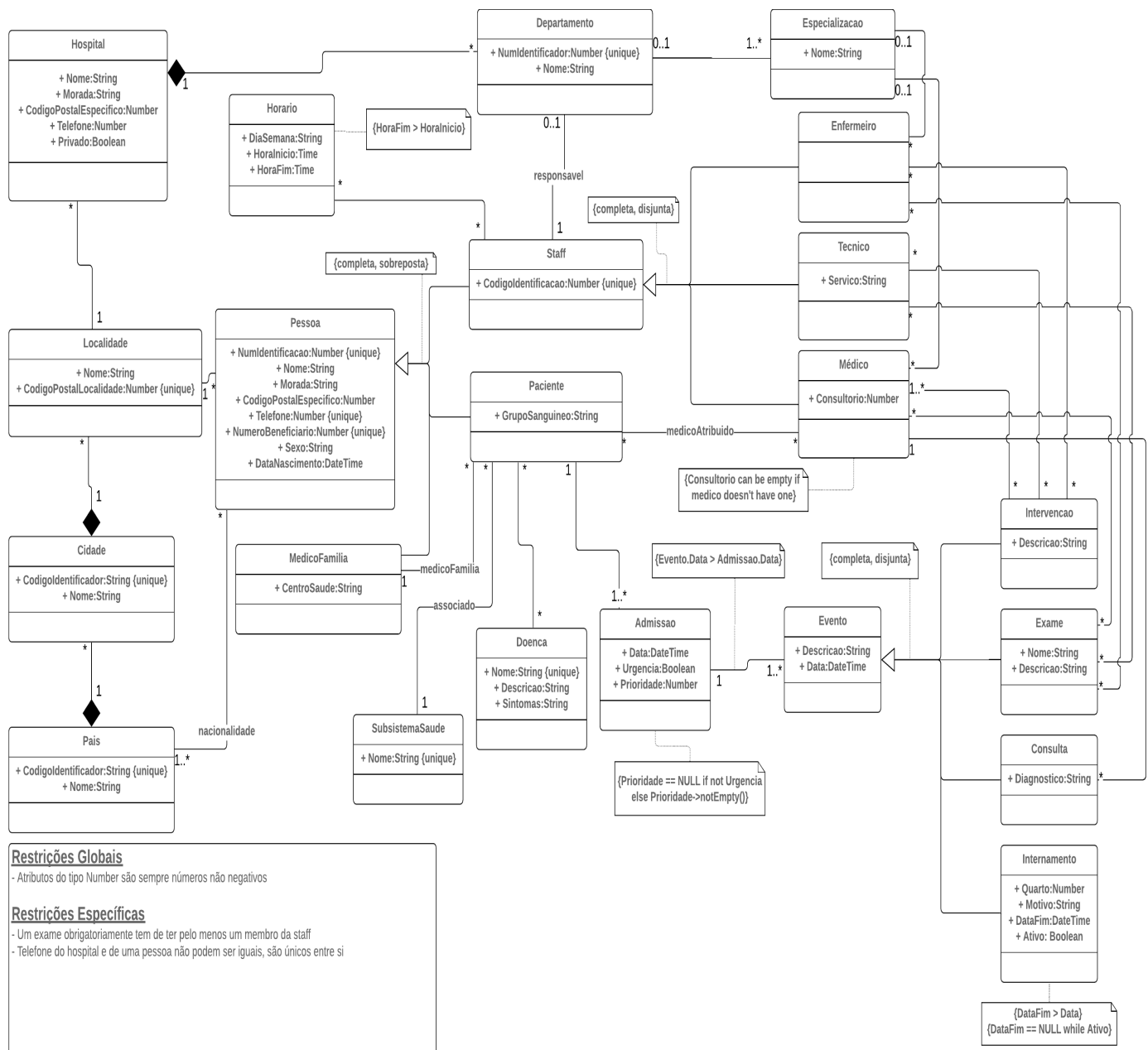
O hospital guarda informação sobre os seus pacientes como o seu grupo sanguíneo, o subsistema de saúde ao qual o paciente está associado, as doenças que o paciente tem, o seu médico de família, os médicos atribuídos naquele hospital, e as suas admissões no hospital.

Sobre cada doença interessa guardar o seu nome, uma descrição da doença e os seus sintomas predominantes.

Uma admissão no hospital tem uma data, se se trata de uma admissão de urgência e, caso seja uma urgência, a prioridade desta. Uma admissão pode desencadear vários tipos de eventos. Cada evento tem uma descrição sobre do que se trata, uma data, e outras informações dependendo do tipo de evento, que pode corresponder a: uma consulta, um exame, uma intervenção ou um internamento.

Numa consulta, interessa guardar o médico que a realizou e o diagnóstico da consulta. Num exame, guarda-se o nome do exame feito e uma descrição deste. Numa intervenção, guarda-se uma descrição da intervenção realizada. Num internamento é necessário guardar o quarto onde o paciente se encontra, o motivo do internamento e a data na qual o paciente recebeu alta (saiu do internamento) caso tal já tenha acontecido.

2 UML - Modelo Conceptual



3 Modelo Relacional

3.1 Classes

Classe Pessoa

- Pessoa(NumIdentificacao, Nome, Morada, Localidade, CodigoPostal, Cidade, Pais, Telefone, NumeroBeneficiario, Sexo, DataNascimento)

Classe Staff

- Staff(PessoaID→Pessoa, CodigoIdentificacao, Especializacao→Especializacao)
 - Enfermeiro(StaffID→Staff)
 - Tecnico(StaffID→Staff)
 - Medico(StaffID→Staff, Consultorio→Quarto)
- Especializacao(Nome, Departamento→Departamento)
- Horario(HorarioID, DiaSemana, HoraInicio, HoraFim)
- Departamento(NumIdentificador, Nome, Responsavel→Staff)

Classe Paciente

- Paciente(PessoaID→Pessoa, GrupoSanguineo, SubsistemaSaude→SubsistemaSaude)
- SubsistemaSaude(Nome)

Classe Evento

- Admissao(AdmissaoID, Data, Urgencia, Prioridade, Paciente→Paciente)
- Doenca(Nome, Descricao, Sintomas)
- Ocorrencia(OcorrenciaID, DataInicio, DataFim, Paciente→Paciente, Doenca→Doenca)
- Quarto(Numero)
- Evento(EventoID, Descricao, Data, Admissao→Admissao, Quarto→Quarto)
 - Intervencao(EventoID→Evento, Descricao)
 - Exame(EventoID→Evento, Nome, Descricao)
 - Consulta(EventoID→Evento, Diagnostico, Medico→Medico)
 - Internamento(EventoID→Evento, Motivo, DataFim, Ativo)

3.2 Associações

Associações muitos-para-muitos

Associações relativas à classe Staff:

- HorarioTrabalho(StaffID→Staff, HorarioID→Horario)

Associações relativas à classe Paciente:

- MedicoAtribuido(PacienteID→Paciente, MedicoID→Medico)

Associações relativas à classe Evento:

- EnfermeiroInterv(EnfermeiroID→Enfermeiro, IntervID→Intervencao)
- EnfermeiroExame(EnfermeiroID→Enfermeiro, ExameID→Exame)
- TecnicoInterv(TecnicoID→Tecnico, IntervID→Intervencao)
- TecnicoExame(TecnicoID→Tecnico, ExameID→Exame)
- MedicoInterv(MedicoID→Medico, IntervID→Intervencao)
- MedicoExame(MedicoID→Medico, ExameID→Exame)
- OcorrenciaEvento(OcorrenciaID→Ocorrencia, EventoID→Evento)

Associações muitos-para-um

Para este tipo de associações foi adotado o método de adicionar uma chave estrangeira para a relação "um" na relação "muitos".

Associações um-para-um

Para este tipo de associações foi adicionado uma chave estrangeira à relação que possuirá o menor número de tuplos, exceto nos casos em que essa relação poder ter esse elemento como nulo, nos casos em que ambos podem ser nulos,

3.3 Generalizações

Quanto as generalizações, optou-se por usar o método E/R, criando uma relação para cada classe, e adicionando a chave da super-classe às subclasses.

4 Análise de Dependências Funcionais e Formas Normais

4.1 Classes

Pessoa

4.2 Associações

5 Restrições

Pessoa

- cada pessoa tem um número de identificação único (**PRIMARY KEY**)
- cada pessoa deve ter um nome, morada, localidade, código postal, cidade, país, número beneficiário, sexo e data nascimento (**NOT NULL**)
- o números de telefone devem ser únicos (**UNIQUE**)

Staff

- cada membro da staff tem um código identificação único, além do número de identificação único (**PRIMARY KEY**)
- além disso, número de identificação (**PessoaID**) é chave estrangeira para Pessoa (**REFERENCES**)
- não é possível eliminar uma pessoa enquanto houver staff mapeada para essa pessoa, mas pode-se alterar a pessoa que alterará na staff que a mapeia (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)
- Especializacao é chave estrangeira para Especializacao, e tal será posta a **NULL** caso a Especializacao seja eliminada, em caso de alterações, as alterações serem feitas em toda a staff que mapeia essa especialização (**ON DELETE SET NULL** e **ON UPDATE CASCADE**)

Enfermeiro

- cada enfermeiro tem o seu código de identificação (**StaffID**) que é chave estrangeira para Staff
- não é possível eliminar um membro da staff enquanto houver um enfermeiro mapeado para esse membro da staff, mas pode-se alterar a staff que alterará também no enfermeiro que a mapeia (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)
- além disso, este código de identificação deve ser único (**PRIMARY KEY**)

Tecnico

- cada técnico tem o seu código de identificação (**StaffID**) que é chave estrangeira para Staff
- não é possível eliminar um membro da staff enquanto houver um técnico mapeado para esse membro da staff, mas pode-se alterar a staff que alterará também no técnico que a mapeia (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)
- além disso, este código de identificação deve ser único (**PRIMARY KEY**)

Medico

- cada médico tem o seu código de identificação (**StaffID**) que é chave estrangeira para Staff
- não é possível eliminar um membro da staff enquanto houver um médico mapeado para esse membro da staff, mas pode-se alterar a staff que alterará também no médico que a mapeia (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)
- além disso, este código de identificação deve ser único (**PRIMARY KEY**)