

# 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

1 de Abril de 2020

**Projeto BDAD - 2019/20 - MIEIC**

**Professora das Aulas Laboratorias:** Carla Alexandra Teixeira Lopes



# Índice

<b>1</b>	<b>Contexto</b>	<b>3</b>
<b>2</b>	<b>UML - Modelo Conceptual</b>	<b>4</b>
<b>3</b>	<b>Modelo Relacional</b>	<b>5</b>
3.1	Classes . . . . .	5
3.2	Associações . . . . .	6
3.3	Generalizações . . . . .	6
<b>4</b>	<b>Análise de DFs e FNs</b>	<b>7</b>
<b>5</b>	<b>Restrições</b>	<b>11</b>

## 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.

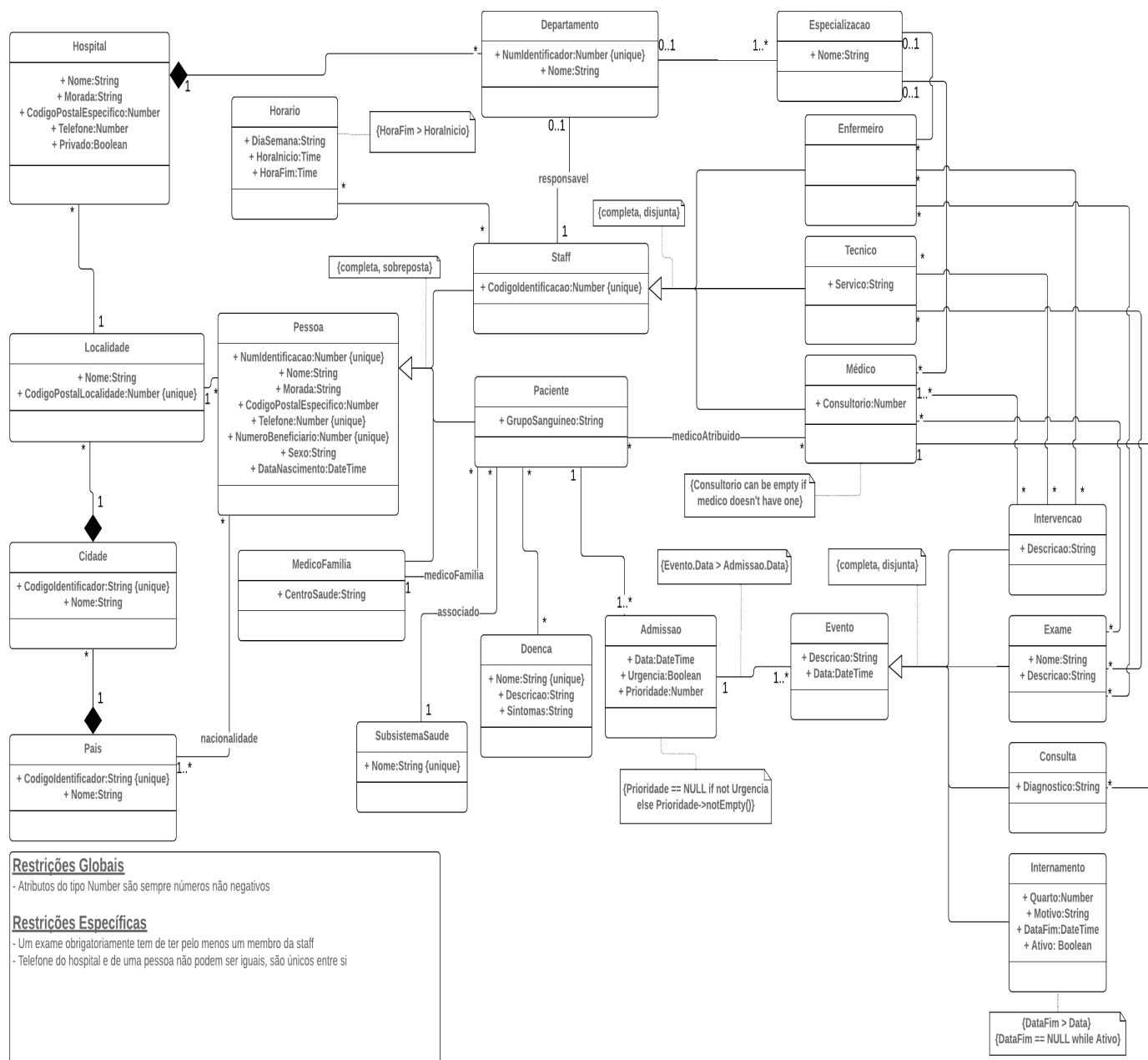
O hospital guarda informação sobre os seus pacientes como o seu grupo sanguíneo, o subsistema de saúde ao qual o paciente está associado, as doenças que o paciente tem, o seu médico de família, os médicos atribuídos naquele hospital, e as suas admissões no hospital.

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(PessoaID, NumIdentificacao, Nome, Morada, Telefone, NumeroBeneficiario, Sexo, DataNascimento)

#### Classe Staff

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

#### Classe Paciente

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

#### Classe Evento

- Admissao(AdmissaoID, Data, Urgencia, Prioridade, Paciente→Paciente)
- Doenca(DoencaID, 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, devido às restrições que necessitam ser implementadas nestas classes.

## 4 Análise de Dependências Funcionais e Formas Normais

### Pessoa

- $\text{PessoaID} \rightarrow \text{NumIdentificacao}, \text{Nome}, \text{Morada}, \text{Telefone}, \text{NumeroBeneficiario}, \text{Sexo}, \text{Data-Nascimento}$
- $\text{NumIdentificacao} \rightarrow \text{PessoaID}, \text{Nome}, \text{Morada}, \text{Telefone}, \text{NumeroBeneficiario}, \text{Sexo}, \text{Data-Nascimento}$
- $\text{Telefone} \rightarrow \text{PessoaID}, \text{NumIdentificacao}, \text{Nome}, \text{Morada}, \text{NumeroBeneficiario}, \text{Sexo}, \text{Data-Nascimento}$
- $\text{NumeroBeneficiario} \rightarrow \text{PessoaID}, \text{NumIdentificacao}, \text{Nome}, \text{Morada}, \text{Telefone}, \text{Sexo}, \text{Data-Nascimento}$

### Staff

- $\text{PessoaID} \rightarrow \text{CodigoIdentificacao}, \text{Especializacao}$
- $\text{CodigoIdentificacao} \rightarrow \text{PessoaID}, \text{Especializacao}$

### Enfermeiro

Não tem dependências funcionais

### Tecnico

Não tem dependências funcionais

### Medico

- $\text{StaffID} \rightarrow \text{Consultorio}$
- $\text{Consultorio} \rightarrow \text{StaffID}$

### Especializacao

- $\text{EspecializacaoID} \rightarrow \text{Nome}, \text{Departamento}$

- Nome→EspecializacaoID, Departamento

## Horario

- HorarioID→DiaSemana, HoraInicio, HoraFim

## Departamento

- NumIdentificador→Nome, Responsavel
- Responsavel→NumIdentificador, Nome

## Paciente

- PessoaID→GrupoSanguineo, SistemaSaude

## SubsistemaSaude

- SistemaSaudeID→Nome
- Nome→SistemaSaudeID

## Admissao

- AdmissaoID→Data, Urgencia, Prioridade, Paciente
- Data, Paciente→AdmissaoID, Urgencia, Prioridade

## Doenca

- DoencaID→Nome, Descricao, Sintomas
- Nome→DoencaID, Descricao, Sintomas

## Ocorrencia

- OcorrenciaID→DataInicio, DataFim, Paciente, Doenca



- DataInicio, Paciente, Doenca  $\rightarrow$  OcorrenciaID, DataFim

## Quarto

Não tem dependências funcionais.

## Evento

- EventoID  $\rightarrow$  Descricao, Data, Admissao, Quarto

## Intervencao

- EventoID  $\rightarrow$  Descricao

## Exame

- EventoID  $\rightarrow$  Nome, Descricao

## Consulta

- EventoID  $\rightarrow$  Diagnostico, Medico

## Internamento

- EventoID  $\rightarrow$  Motivo, DataFim, Ativo

## HorarioTrabalho

Não tem dependências funcionais.

## MedicoAtribuido

Não tem dependências funcionais.

## EnfermeiroInterv

Não tem dependências funcionais.

## **EnfermeiroExame**

Não tem dependências funcionais.

## **TecnicoInterv**

Não tem dependências funcionais.

## **TecnicoExame**

Não tem dependências funcionais.

## **MedicoInterv**

Não tem dependências funcionais.

## **MedicoExame**

Não tem dependências funcionais.

## **OcorrenciaEvento**

Não tem dependências funcionais.

## 5 Restrições

### Pessoa

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

### Staff

- cada membro da staff tem um ID único (**PRIMARY KEY**)
- além disso, 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 serão feitas em toda a staff que mapeia essa especialização (**ON DELETE SET NULL** e **ON UPDATE CASCADE**)

### Enfermeiro

- cada enfermeiro tem o seu ID (**StaffID**) que é chave estrangeira para Staff (**REFERENCES**)
- 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 ID deve ser único (**PRIMARY KEY**)

### Tecnico

- cada técnico tem o seu ID (**StaffID**) que é chave estrangeira para Staff (**REFERENCES**)
- 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 ID deve ser único (**PRIMARY KEY**)

## Medico

- cada médico tem o seu ID (**StaffID**) que é chave estrangeira para Staff (**REFERENCES**)
- 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 ID deve ser único (**PRIMARY KEY**)
- Consultorio é chave estrangeira para Quarto (**REFERENCES**)
- o consultório deve ser único (**UNIQUE**)
- caso se elimine um quarto que é mapeado como consultório, o valor do consultório é colocado a nulo, em caso de alteração, o consultório também é alterado (**ON DELETE SET NULL** e **ON UPDATE CASCADE**)

## Especializacao

- cada especialização deve ter um ID único (**PRIMARY KEY**)
- nome da especialização deve ser único e não nulo (**NOT NULL** e **UNIQUE**)
- Departamento é chave estrangeira para Departamento
- caso se elimine um departamento, será colocado a nulo o valor do departamento em todas as especializações que o mapeiem, em caso de alteração, será alterado também (**ON UPDATE CASCADE** e **ON DELETE SET NULL**)

## Horario

- cada horário tem um ID único (**PRIMARY KEY**)
- cada horário deve ter um dia da semana em qual está incidido, uma hora de início e uma hora de fim (**NOT NULL**)
- A hora de início deve ser antes da hora de fim (**CHECK**)

## Departamento

- cada departamento tem o seu número identificador único (**PRIMARY KEY**)
- cada departamento tem um nome e um responsável (**NOT NULL**)

- Responsavel é chave estrangeira para staff (**REFERENCES**)
- não é possível remover um membro da staff caso esse esteja mapeado como um Responsavel de um departamento, mas é possível alterá-lo, tais alterações serão aplicadas no departamento que mapeia o membro da staff (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)
- não pode haver o mesmo membro da staff responsável por dois departamentos (**UNIQUE**)

## Paciente

- cada paciente tem um 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 um paciente mapeada para essa pessoa, mas pode-se alterar a pessoa que alterará no paciente que a mapeia (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)
- cada paciente tem um grupo sanguineo (**NOT NULL**)
- SubistemaSaude é chave estrangeira para SubistemaSaude (**REFERENCES**)
- ao eliminar um subsistema de saúde colocará em todos os pacientes que o mapeavam o valor **NULL**, em caso de alteração, alterará também no paciente que o mapeia (**ON DELETE SET NULL** e **ON UPDATE CASCADE**)

## SubistemaSaude

- cada subsistema de saúde tem um ID único (**PRIMARY KEY**)
- nome do subsistema de saúde deve ser único e não nulo (**NOT NULL** e **UNIQUE**)

## Admissao

- cada admissão tem um ID único (**PRIMARY KEY**)
- cada admissão tem uma data e um paciente (**NOT NULL**)
- uma admissão pode ser urgência ou não (isto é, tem o valor 1 ou 0, respetivamente), por omissão trata-se de uma admissão não urgente (**NOT NULL** e **CHECK** e **DEFAULT**)
- caso seja uma admissão de urgência a prioridade deve ter um valor maior que 0, em caso não urgente, a prioridade por omissão é 0 (**NOT NULL**, **CHECK** e **DEFAULT**)
- Paciente é chave estrangeira para Paciente (**REFERENCES**)

- não é possível eliminar um paciente caso este esteja mapeado em Admissao, e alterar um paciente altera também em Admissao (**ON UPDATE CASCADE** e **ON DELETE RESTRICT**)

## Doenca

- cada doença tem um ID único (**PRIMARY KEY**)
- cada doença tem um nome, uma descrição e os seus sintomas (**NOT NULL**)
- o nome da doença deve ser único (**UNIQUE**)

## Ocorrancia

- cada ocorrência tem um ID único (**PRIMARY KEY**)
- cada ocorrência tem uma data início, um paciente e uma doença associada à ocorrência (**NOT NULL**)
- Paciente é chave estrangeira para Paciente, Doenca é chave estrangeira para Doenca (**REFERENCES**)
- não é possível eliminar uma doença ou um paciente se houver ocorrências a mapeá-los, mas ao alterar os mesmos, a informação também será alterada nas ocorrências que os mapeiam (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)
- se a data de fim for não nula, então deve ser depois da data de início (**CHECK**)

## Quarto

- cada quarto tem um número único (**PRIMARY KEY**)

## Evento

- cada evento tem um ID único (**PRIMARY KEY**)
- cada admissão tem uma descrição e uma data, a admissão ao qual está relacionada e o quarto onde foi procedido esse evento (**NOT NULL**)
- Admissao é chave estrangeira para Admissao, Quarto é chave estrangeira para Quarto (**REFERENCES**)

- não é possível eliminar uma admissão nem um quarto enquanto houver ocorrências a mapeá-los, em caso de alteração, estas também são aplicadas nas ocorrências que os mapeiam (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)

## Inervenciaao

- cada intervenção tem um ID (do evento) único (**PRIMARY KEY**)
- cada intervenção tem uma descrição (**NOT NULL**)
- o ID da intervenção (**EventoID**) é chave estrangeira para Evento (**REFERENCES**)
- não é possível eliminar um evento se houver uma intervenção que o mapeia, em caso de alterações, estas também são aplicadas nas intervenções que os mapeiam (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)

## Exame

- cada exame deve ter um ID (do evento) único (**PRIMARY KEY**)
- cada exame tem um nome e uma descrição (**NOT NULL**)
- o ID do exame (**EventoID**) é chave estrangeira para Evento (**REFERENCES**)
- não é possível eliminar um evento se houver um exame que o mapeia, em caso de alterações, estas também são aplicadas nos exames que os mapeiam (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)

## Consulta

- cada consulta deve ter um ID (do evento) único (**PRIMARY KEY**)
- cada consulta tem um médico associado e um diagnóstico (**NOT NULL**)
- o ID da consulta (**EventoID**) é chave estrangeira para Evento, Medico é chave estrangeira para Medico (**REFERENCES**)
- não é possível eliminar um evento se houver uma consulta que o mapeia, em caso de alterações, estas também são aplicadas nas consulta que os mapeiam. O mesmo se aplica para a chave estrangeira Medico (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)

## Internamento

- cada internamento deve ter um ID (do evento) único (**PRIMARY KEY**)
- cada internamento tem um motivo e se o internamento está ativo ou não (**NOT NULL**)
- o ID do internamento (**EventoID**) é chave estrangeira para Evento (**REFERENCES**)
- não é possível eliminar um evento se houver um internamento que o mapeia, em caso de alterações, estas também são aplicadas nos internamento que os mapeiam (**ON DELETE RESTRICT** e **ON UPDATE CASCADE**)
- Ativo é um valor booleano, tomando o valor de 0 ou 1 (**CHECK**)
- se Ativo estiver a 1 então DataFim deve ser não nula, se tiver a 0 então DataFim deve ser nula (**CHECK**)
- DataFim deve ser maior que a Data do evento mapeado na chave estrangeira EventoID (**TRIGGER**)

## HorarioTrabalho

- cada intermante deve ter um ID de staff e um ID de horário único (**PRIMARY KEY**)
- StaffID é chave estrangeira para Staff e HorarioID chave estrangeira para Horario (**REFERENCES**)
- ao eliminar uma staff que esteja mapeada num HorarioTrabalho, eliminará também o HorarioTrabalho, em caso de alteração, também alterará em HorarioTrabalho. O mesmo aplica-se à chave estrangeira Horario (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)

## MedicoAtribuido

- cada intermante deve ter um ID de um paciente e um ID de um médico único (**PRIMARY KEY**)
- PacienteID é chave estrangeira para Paciente e MedicoID chave estrangeira para Medico (**REFERENCES**)
- ao eliminar um paciente que esteja mapeada num MedicoAtribuido, eliminará também o MedicoAtribuido, em caso de alteração, também alterará em MedicoAtribuido. O mesmo aplica-se à chave estrangeira MedicoID (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)



## EnfermeiroInterv

- deve ter uma intervenção ao qual foi atribuído um enfermeiro (**PRIMARY KEY**)
- EnfermeiroID é uma chave estrangeira para Enfermeiro e IntervID é chave estrangeira para Intervencao
- ao eliminar um enfermeiro que esteja mapeado num EnfermeiroInterv, eliminará também o EnfermeiroInterv, em caso de alteração, também alterará em EnfermeiroInterv. O mesmo aplica-se à chave estrangeira IntervID (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)

## EnfermeiroExame

- deve ter um exame ao qual foi atribuído um enfermeiro (**PRIMARY KEY**)
- EnfermeiroID é uma chave estrangeira para Enfermeiro e ExameID é chave estrangeira para Exame
- ao eliminar um enfermeiro que esteja mapeado num EnfermeiroExame, eliminará também o EnfermeiroExame, em caso de alteração, também alterará em EnfermeiroExame. O mesmo aplica-se à chave estrangeira ExameID (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)

## TecnicoInterv

- deve ter uma intervenção ao qual foi atribuído um técnico (**PRIMARY KEY**)
- TecnicoID é uma chave estrangeira para Tecnico e IntervID é chave estrangeira para Intervencao
- ao eliminar um técnico que esteja mapeado num TecnicoInterv, eliminará também o TecnicoInterv, em caso de alteração, também alterará em TecnicoInterv. O mesmo aplica-se à chave estrangeira IntervID (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)

## TecnicoExame

- deve ter um exame ao qual foi atribuído um técnico (**PRIMARY KEY**)
- TecnicoID é uma chave estrangeira para Tecnico e ExameID é chave estrangeira para Exame

- ao eliminar um técnico que esteja mapeado num TecnicoExame, eliminará também o TecnicoExame, em caso de alteração, também alterará em TecnicoExame. O mesmo aplica-se à chave estrangeira ExameID (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)

## MedicoInterv

- deve ter uma intervenção ao qual foi atribuído um médico (**PRIMARY KEY**)
- MedicoID é uma chave estrangeira para Medico e IntervID é chave estrangeira para Intervencao
- ao eliminar um médico que esteja mapeado num MedicoInterv, eliminará também o MedicoInterv, em caso de alteração, também alterará em MedicoInterv. O mesmo aplica-se à chave estrangeira IntervID (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)

## MedicoExame

- deve ter um exame ao qual foi atribuído um médico (**PRIMARY KEY**)
- MedicoID é uma chave estrangeira para Medico e ExameID é chave estrangeira para Exame
- ao eliminar um médico que esteja mapeado num MedicoExame, eliminará também o MedicoExame, em caso de alteração, também alterará em MedicoExame. O mesmo aplica-se à chave estrangeira ExameID (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)

## OcorrenciaEvento

- deve ter um evento que esteja ligada a uma ocorrência (**PRIMARY KEY**)
- OcorrenciaID é uma chave estrangeira para Ocorrencia e EventoID é chave estrangeira para Evento
- ao eliminar uma ocorrência que esteja mapeado num OcorrenciaEvento, eliminará também o OcorrenciaEvento, em caso de alteração, também alterará em OcorrenciaEvento. O mesmo aplica-se à chave estrangeira EventoID (**ON DELETE CASCADE** e **ON UPDATE CASCADE**)