

# Rede de Centros de Vacinação

## *Entrega II*

*Bases de Dados 2020/2021*

*Grupo 709*

Adriano Soares [up201904873@fe.up.pt](mailto:up201904873@fe.up.pt)  
Francisco Cerqueira [up201905337@fe.up.pt](mailto:up201905337@fe.up.pt)  
Guilherme Calassi [up201800157@fe.up.pt](mailto:up201800157@fe.up.pt)

# *Índice*

<b>Contexto</b>	<b>3</b>
<b>Diagrama UML (Revisto)</b>	<b>4</b>
<b>Esquema Relacional</b>	<b>5</b>
<b>Dependências Funcionais e Formas Normais</b>	<b>6</b>
<b>Restrições</b>	<b>6</b>
<b>Dificuldades Encontradas</b>	<b>11</b>

## Contexto

Uma rede de centros de vacinação pretende armazenar informação relativamente aos mesmos, onde os enfermeiros irão vacinar um conjunto de utentes.

Ambos são **peessoas**, descritos pelo seu nome, idade, morada, código postal, telefone, email, data de nascimento, gênero, data em que foi registrado no sistema e ainda um identificador global que servirá de identificação no sistema.

Cada **centro** é caracterizado por um identificador único, uma morada, telefone, email e possivelmente um website.

Para fins estatísticos, uma **morada** deve conter informação sobre o distrito, concelho e código postal e cada pessoa deve estar associada a exatamente uma morada.

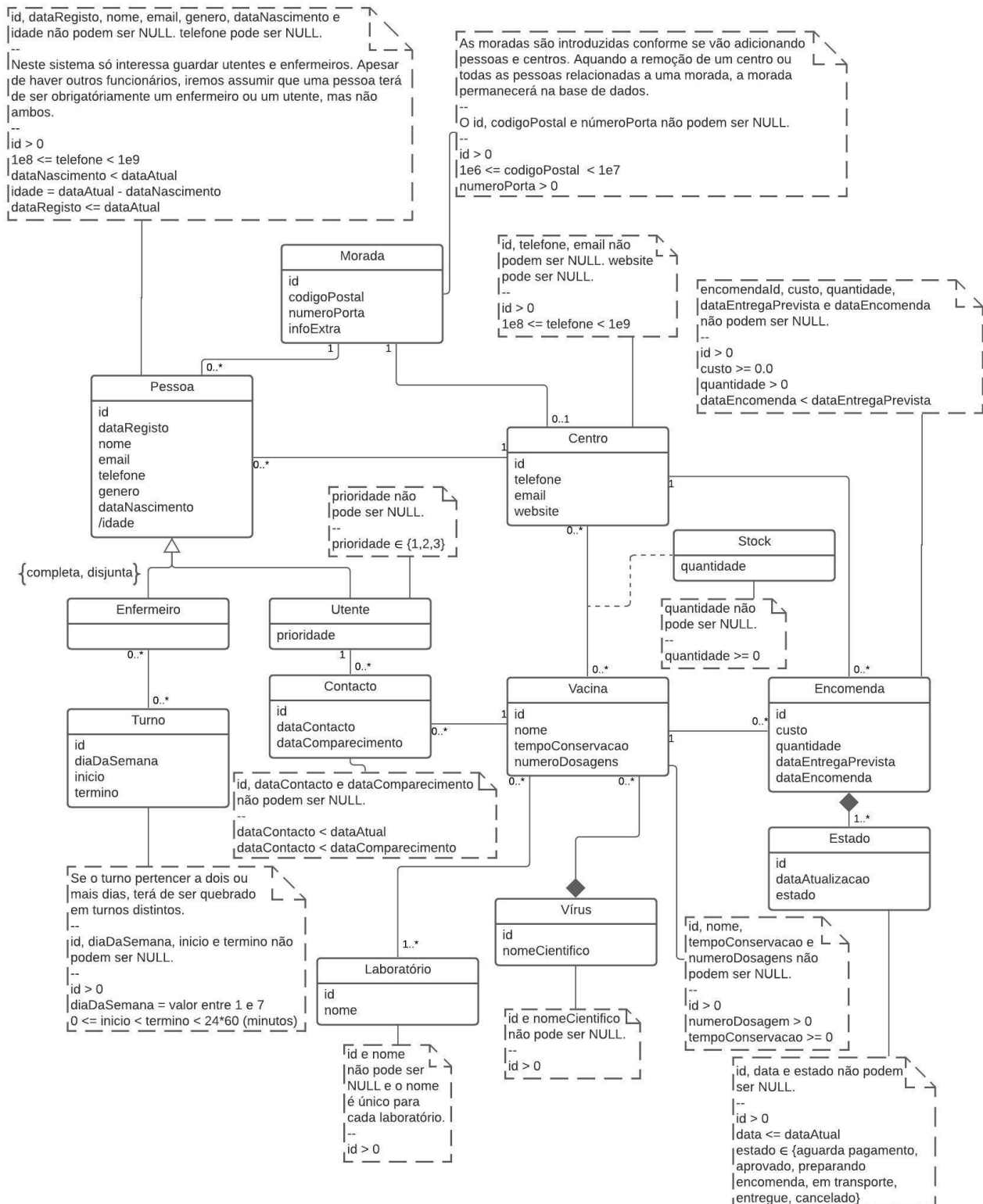
Os **utentes** poderão ser assinalados como prioritários, consoante a sua situação, e poderão ser **contactados** pelo centro, pretendendo-se guardar a respetiva data em que foi contactado e a data de comparecimento.

Os **enfermeiros** deverão estar associados a **turnos**, que são caracterizados por uma data de início, término e dia da semana, podendo excepcionalmente estar de baixa ou de férias, não tendo algum turno atribuído.

Acerca das **vacinas**, é necessário saber a data de validade, nome, modo de administração, **laboratório(s)**, número de dosagens e o **vírus** ao qual está associada, interessando apenas saber o nome científico do mesmo. Também será necessário guardar o **stock** de cada uma das vacinas em cada centro, para que este seja renovado quando necessário.

Finalmente, pretende-se guardar informações acerca das **encomendas** realizadas por cada centro, sendo necessário saber o respetivo o id, que dependerá do centro associado, custo, quantidade de cada tipo de vacina, o **estado** da encomenda, interessando guardar a data de cada estado pelo qual a encomenda passa, a data de entrega prevista e a data da encomenda.

# Diagrama UML (Revisto)



# Esquema Relacional

1. **Morada** (id, codigoPostal, numeroPorta, infoExtra)
  - i. {id} -> {codigoPostal, numeroPorta, infoExtra}
2. **Centro** (id, telefone, email, website, moradaID -> Morada)
  - i. {id} -> {telefone, email, website, moradaID}
3. **Enfermeiro** (id, dataRegisto, nome, email, telefone, genero, dataNascimento, idade, moradaID -> Morada, centroID -> Centro)
  - i. {id} -> {dataRegisto, nome, email, telefone, genero, dataNascimento, idade, moradaID, centroID}
4. **Utente** (id, dataRegisto, nome, email, telefone, genero, dataNascimento, idade, moradaID -> Morada, prioridade, centroID -> Centro)
  - i. {id} -> {dataRegisto, nome, email, telefone, genero, dataNascimento, idade, moradaID, centroID, prioridade}
5. **Turno** (id, diaDaSemana, inicio, termino)
  - i. {id} -> {diaDaSemana, inicio, termino}
6. **Horario** (enfermeiroID -> Enfermeiro, turnoID -> Turno)
7. **Contacto** (id, dataContacto, dataComparecimento, utenteID -> Utente, vacinaID -> Vacina)
  - i. {id} -> {dataContacto, dataComparecimento, utenteID, vacinaID}
8. **Virus** (id, nomeCientifico)
  - i. {id} -> {nomeCientifico}
9. **Laboratorio** (id, nome)
  - i. {id} -> {nome}
10. **Vacina** (id, nome, dataValidade, numeroDosagens, virusID -> Virus)
  - i. {id} -> {nome, dataValidade, numeroDosagens, virusID}
11. **Fabricante** (vacinaID -> Vacina, laboratorioID -> Laboratorio)
12. **Stock** (vacinaID -> Vacina, centroID -> Centro, quantidade)
  - i. {vacinaID, centroID} -> {quantidade}
13. **Encomenda** (id, centroID -> Centro, vacinaID -> Vacina, custo, quantidade, dataEntregaPrevista, dataEncomenda)
  - i. {id} -> {centroID, vacinaID, custo, quantidade, dataEntregaPrevista, dataEncomenda}
14. **Estado** (id, encomendaID -> Encomenda, estado, dataAtualização)
  - i. {id} -> {encomendaID, estado, dataAtualização}

**Nota:** A estratégia escolhida para converter a generalização foi *object-oriented*, visto que é ideal para generalizações disjuntas.

# *Dependências Funcionais e Formas Normais*

Todas as relações cumprem com os requisitos da Forma Normal de Boyce-Codd, na medida em que o lado esquerdo de cada dependência enumerada na página anterior é uma super-key da respetiva relação - condição suficiente, encontrando-se consequentemente também na 3ª Forma Normal.

## *Restrições*

**PK:** Primary Key

**UK:** Unique Key

**CK:** Check

**NN:** Not Null

### Morada

Não podem haver duas moradas com o mesmo id. Id é positivo, código postal tem que ter obrigatoriamente 7 dígitos e o número de porta tem que ser positivo. Todas as moradas devem ter um código postal, e número de porta associado, sendo a informação extra opcional.

*id: PK, CHECK (id > 0)*

*codigoPostal: NN, CHECK (1e6 <= codigoPostal && codigoPostal < 1e7)*

*numeroPorta: NN, CHECK (numeroPorta > 0)*

*infoExtra: --*

### Centro

Não podem haver dois centros com o mesmo id nem com a mesma morada. Id é positivo e o número de telefone, se existente, terá 9 dígitos. Todos os centros devem ter um email e morada associados, sendo o website e o número de telefone opcionais.

*id: PK, CHECK (id > 0)*

*telefone: NN, CHECK (1e8 <= telefone && telefone < 1e9)*

*email: NN*

*moradaID: NN, UK, REFERENCES Morada*

*website: --*

## Enfermeiro

Não pode haver dois enfermeiros com o mesmo id. Id é positivo e o número de telefone, se existente, terá 9 dígitos. Todos os enfermeiros devem ter uma data de registo, nome, email, género, data de nascimento, morada e centro associados.

*id: **PK, CHECK (id > 0)***  
*dataRegisto: **NN, CHECK (dataRegisto < NOW())***  
*nome: **NN***  
*email: **NN***  
*telefone: **CHECK (1e8 <= telefone && telefone < 1e9)***  
*genero: **NN***  
*dataNascimento: **NN, CHECK (dataNascimento < NOW())***  
*moradaID: **NN, REFERENCES Morada***  
*centroID: **NN, REFERENCES Centro***

## Utente

Não pode haver dois utentes com o mesmo id. Id é positivo, o número de telefone, se existente, terá 9 dígitos e a prioridade é um número entre 1 e 3. Todos os utentes devem ter uma data de registo, nome, email, género, data de nascimento, morada e centro associados.

*id: **PK, CHECK (id > 0)***  
*dataRegisto: **NN, CHECK (dataRegisto < NOW())***  
*nome: **NN***  
*email: **NN***  
*telefone: **CHECK (1e8 <= telefone && telefone < 1e9)***  
*genero: **NN***  
*dataNascimento: **NN, CHECK (dataNascimento < NOW())***  
*prioridade: **NN, CHECK (prioridade > 0 && prioridade < 4)***  
*moradaID: **NN, REFERENCES Morada***  
*centroID: **NN, REFERENCES Centro***

## Turno

Não podem haver dois turnos com o mesmo id. Id é positivo, o dia da semana está compreendido entre 1 e 7, a hora de início é superior a 0 e inferior à hora de término, sendo esta inferior a 24\*60 (minutos). Todos os turnos devem ter um dia da semana, hora de início e término associado. Não podem haver dois turnos com as mesmas horas de início e término iguais.

*id: PK, CHECK (id > 0)*  
*diaDaSemana: NN, CHECK (diaDaSemana > 0 && diaDaSemana <= 7)*  
*inicio: NN, CHECK (inicio >= 0)*  
*termino: NN, CHECK (termino < 24\*60)*  
*CHECK (inicio < termino)*  
*UNIQUE (diaDaSemana, inicio, termino)*

## Horário

Não podem haver dois horários com o mesmo enfermeiro e turno associado. Todos os horários devem ter um enfermeiro e turno associados.

*enfermeiroID: REFERENCES Enfermeiro*  
*turnoID: REFERENCES Turno*  
*PK (enfermeiroID, turnoID)*

## Contacto

Não podem haver dois contactos com o mesmo id. Id é positivo, a data de contacto é inferior à data de comparecimento. Todos os contactos devem ter uma data de contacto, data de comparecimento, utente e vacina associados.

*id: PK, CHECK (id > 0)*  
*dataContacto: NN, CHECK (dataContacto <= NOW())*  
*dataComparecimento: NN*  
*utenteID: NN, REFERENCES Utente*  
*vacinaID: NN, REFERENCES Vacina*  
*CHECK (dataContacto < dataComparecimento)*



## Virus

Não podem haver dois vírus com o mesmo id, positivo, nem com o mesmo nome científico.

*id: PK, CHECK (id > 0)*

*nomeCientifico: NN, UK*

## Laboratorio

Não podem haver laboratórios com o mesmo id nem com o mesmo nome.

*id: PK, CHECK (id > 0)*

*nome: NN, UK*

## Vacina

Não podem haver duas vacinas com o mesmo id. Id e o número de dosagens têm de ser positivos. Todas as vacinas devem ter um vírus, nome, data de validade e número de dosagens associado.

*id: PK, CHECK (id > 0)*

*virusID: NN, REFERENCES Vacina*

*nome: NN*

*tempoConservacao: NN*

*numeroDosagens: NN, CHECK (numeroDosagens > 0)*

## Fabricante

Não podem haver dois fabricantes com a mesma vacina e laboratório associado. Todos os fabricantes devem ter uma vacina e laboratório associados.

*vacinaID: REFERENCES Vacina*

*laboratorioID: REFERENCES Laboratorio*

*PK (vacinaID, laboratorioID)*

## Stock

Não podem haver dois registos com o mesmo par de vacinaID e centroID. A quantidade é um atributo obrigatório, não podendo ser negativa. VacinaID referencia Vacina, bem como centroID referencia Centro.

*vacinaID: REFERENCES Vacina*  
*centroID: REFERENCES Centro*  
*quantidade: NN, CHECK (quantidade >= 0)*  
*PK (vacinaID, centroID)*

## Encomenda

Não podem haver duas encomendas com o mesmo id. O id é positivo, o custo é não negativo, a quantidade é superior ou igual a 1, e data de encomenda é inferior à data de entrega. CentroID referencia a Centro, assim como VacinaID referencia a Vacina. Nenhum atributo pode ser null.

*id: PK, CHECK (id > 0)*  
*centroID: NN, REFERENCES Centro*  
*vacinaID: NN, REFERENCES Vacina*  
*custo: NN, CHECK (custo >= 0.0)*  
*quantidade: NN, CHECK (quantidade > 0)*  
*dataEntregaPrevista: NN*  
*dataEncomenda: NN*  
*CHECK (dataEncomenda < dataEntregaPrevista)*

## Estado

Não podem existir dois estados com o mesmo id. EncomendaID, data e estado são atributos obrigatórios.

*id: PK, CHECK (id > 0)*  
*encomendaID: NN*  
*dataAtualização: NN, CHECK (data <= NOW())*  
*estado: NN, IN ("aguarda pagamento", "aprovado", "preparando encomenda", "em transporte", "entregue", "cancelado")*

Nota: Foram aplicados a todos os atributos que são chaves estrangeiras as restrições ON DELETE RESTRICT ON UPDATE CASCADE, à exceção das composições que têm ON DELETE CASCADE ON UPDATE CASCADE.

## *Dificuldades Encontradas*

Foram detetados possíveis gatilhos, como a atualização do stock em função das encomendas, ou a idade. No entanto, estas funcionalidades apenas serão implementadas na terceira entrega do trabalho.

Relativamente à verificação dos dados utilizando CHECK(), reparamos que recentemente (2021-03-17) foi implementada uma versão de SQLite que impede a utilização da função *strftime()* sem o 'now' como argumento, pois corrigiu uma falha na deteção da mesma ser não-determinística. Mais informações em <https://sqlite.org/deterministic.html> (Ponto 3.1). Deste modo, optamos por utilizar a variável CURRENT\_TIMESTAMP, de forma a contornar esse obstáculo.