

Base de Dados

Centros de Doação de sangue



Grupo 68 – P10

Trabalho realizado por:

Beatriz Rebelo nº 50511

Mauro Cardoso nº 49845

Sara Simões nº 50512

Descrição e Objetivos

Neste trabalho iremos desenvolver uma base de dados para um sistema que gere vários centros de doação de sangue. Este sistema irá regular todas as doações de sangue feitas nestes centros bem como todas as partes envolvidas neste processo.

Adicionalmente incluirá uma pequena parte relativa às urgências de um hospital (simplificado) para gerir as transfusões de sangue para o paciente.

A base de dados permitirá registar várias doações de sangue num determinado centro de doação e guardar dados sobre as mesmas.

Sobre cada doação de sangue são registados os seguintes dados: data da doação (dia/mês/ano), tipo de sangue doado (Grupo A +, O +, B +, O -, A -, AB+, B - e AB-), quantidade recolhida (em ml) e o NIF do dador. A quantidade recolhida vai ter uma restrição, visto que não se pode colher mais de 450 ml de sangue.

Relativamente ao dador (que terá de fazer uma inscrição da primeira vez que faz uma doação) sabe-se um email, um número de telefone, código postal, nome, NIF (identificador único), data de nascimento (dia/mês/ano), peso na altura da inscrição (em kg), sexo (feminino/masculino), tipo de sangue e um histórico de doações. Quando é feita uma doação a idade e o peso representam restrições porque é necessário ter uma idade entre os 18 e os 65 anos e em simultâneo ter mais de 50 kg. Da mesma maneira o histórico de doações vai constituir uma restrição pois um dador (dependendo do seu sexo) só pode doar sangue de tempo a tempo.

Para além disso, irão ser registados dados sobre funcionários que poderão ser médicos, técnicos, auxiliares de limpeza ou enfermeiros. Os funcionários terão um salário fixo (em euros), bem como um email, um número de telefone, código postal, nome e NIF. Estes têm acesso a vários equipamentos médicos para os quais a base de dados guardará o nome e o número de série (identificador único).

Todos os funcionários têm de trabalhar num estabelecimento, apenas enfermeiros podem trabalhar em centros de doação, mas em urgências de hospitais podem trabalhar todos os tipos de funcionários.

Um funcionário também poderá ser dador. Neste caso específico o sistema também guardará as informações necessárias para este efetuar uma doação, a quando da sua inscrição como tal.

Cada centro de doação de sangue vai ter um identificador e o sistema vai também guardar o seu nome, código postal e um telefone.

A base de dados irá armazenar dados sobre as urgências de hospitais incluindo o código postal, nome e um telefone. Nestas é possível efetuar várias transfusões de sangue a pacientes,

estes serão registados com o seu nome, NIF, tipo de sangue, um email, um número de telefone, código postal, data de nascimento (dia/mês/ano) e um histórico de transfusões.

Para ser efetuada uma transfusão de sangue é necessário verificar se existe em stock os tipos de sangue compatíveis com o grupo sanguíneo do paciente. Assim é preciso guardar no sistema, informações sobre o stock nomeadamente se determinado tipo de sangue está disponível.

Por fim falta apenas guardar no sistema, dados sobre cada grupo sanguíneo, tais como a sua sigla e quais os outros grupos compatíveis consigo.

Esquema Relacional

✓ Pessoas (NIF, Código-Postal, Nome, Número de Telefone, Email, Data de Nascimento)

✓ Dadores (NIF, Peso, Sexo)

- NIF é chave estrangeira referindo NIF em Pessoas.

✓ Doações (Data de Doação, Quantidade Recolhida, NIF, Sigla)

- NIF é chave estrangeira referindo NIF em Dadores.
- Sigla é chave estrangeira referindo Sigla em Tipos de Sangue.

✓ Histórico de Doações (NIF, Número da doação, Data de doação)

- NIF é chave estrangeira referindo NIF em Dadores.

✓ Tipo de Sangue (Sigla)

✓ Pacientes (NIF, Sigla)

- NIF é chave estrangeira referindo NIF em Pessoas.
- Sigla é chave estrangeira referindo Sigla em Tipos de Sangue.

✓ Transfusões (Data de Transfusão, Quantidade Recolhida, NIF, Sigla)

- NIF é chave estrangeira referindo NIF em Pacientes.
- Sigla é chave estrangeira referindo Sigla em Tipos de Sangue.

✓ Histórico de Transfusões (Número da Transfusão, Data de Transfusão, NIF)

- NIF é chave estrangeira referindo NIF em Pacientes.

✓ Estabelecimentos (Nome de Estabelecimento, Número de Telefone, Código-Postal)

✓ Centros (Nome de Estabelecimento)

- Nome de Estabelecimento é chave estrangeira Nome de Estabelecimento em Estabelecimentos

✓ Urgências de Hospitais (Nome de Estabelecimento, nrSerie)

- Nome de Estabelecimento é chave estrangeira referindo Nome de Estabelecimento em Estabelecimentos.
- nrSerie é chave estrangeira referindo nrSerie em Stock.

✓ Funcionários (NIF, Salário, Nome de Estabelecimentos)

- NIF é chave estrangeira referindo NIF em Pessoas.
- Nome do Estabelecimento é chave estrangeira referindo Nome de Estabelecimento em Urgências de Hospital.

✓ Médicos (NIF, Especialidade)

- NIF é chave estrangeira referindo NIF em Funcionários.

✓ Auxiliares de Limpeza (NIF)

- NIF é chave estrangeira referindo NIF em Funcionários.

✓ Técnicos (NIF)

- NIF é chave estrangeira referindo NIF em Funcionários.

✓ Enfermeiros (NIF, Nome do Centro)

- NIF é chave estrangeira referindo NIF em Funcionários.
- Nome do Centro é chave estrangeira referindo Nome de Estabelecimento em Centros.

✓ Equipamentos médicos (Nome de Equipamento, nrSerieEquip)

✓ Stock (nrSerie, Quantidade)

✓ Compatível (Sigla do tipo de sangue, Sigla do tipo de sangue compatível)

- Sigla do tipo de sangue é chave estrangeira referindo Sigla em Tipos de Sangue.
- Sigla do tipo de sangue compatível é chave estrangeira referindo Sigla em Tipos de Sangue.

✓ Tipo em stock (Sigla, nrSerie)

- Sigla é chave estrangeira referindo Sigla em Tipos de Sangue.
- nrSerie é chave estrangeira referindo nrSerie em Stock.

✓ Doam em (NIF, Nome de Estabelecimento)

- NIF é chave estrangeira referindo NIF em Dadores.
- Nome de Estabelecimento é chave estrangeira referindo nome em Centros.

✓ Recebem em (NIF, Nome de Estabelecimento)

- NIF é chave estrangeira referindo NIF em Pacientes.
- Nome de Estabelecimento é chave estrangeira referindo Nome de Estabelecimento em Urgências de Hospitais.

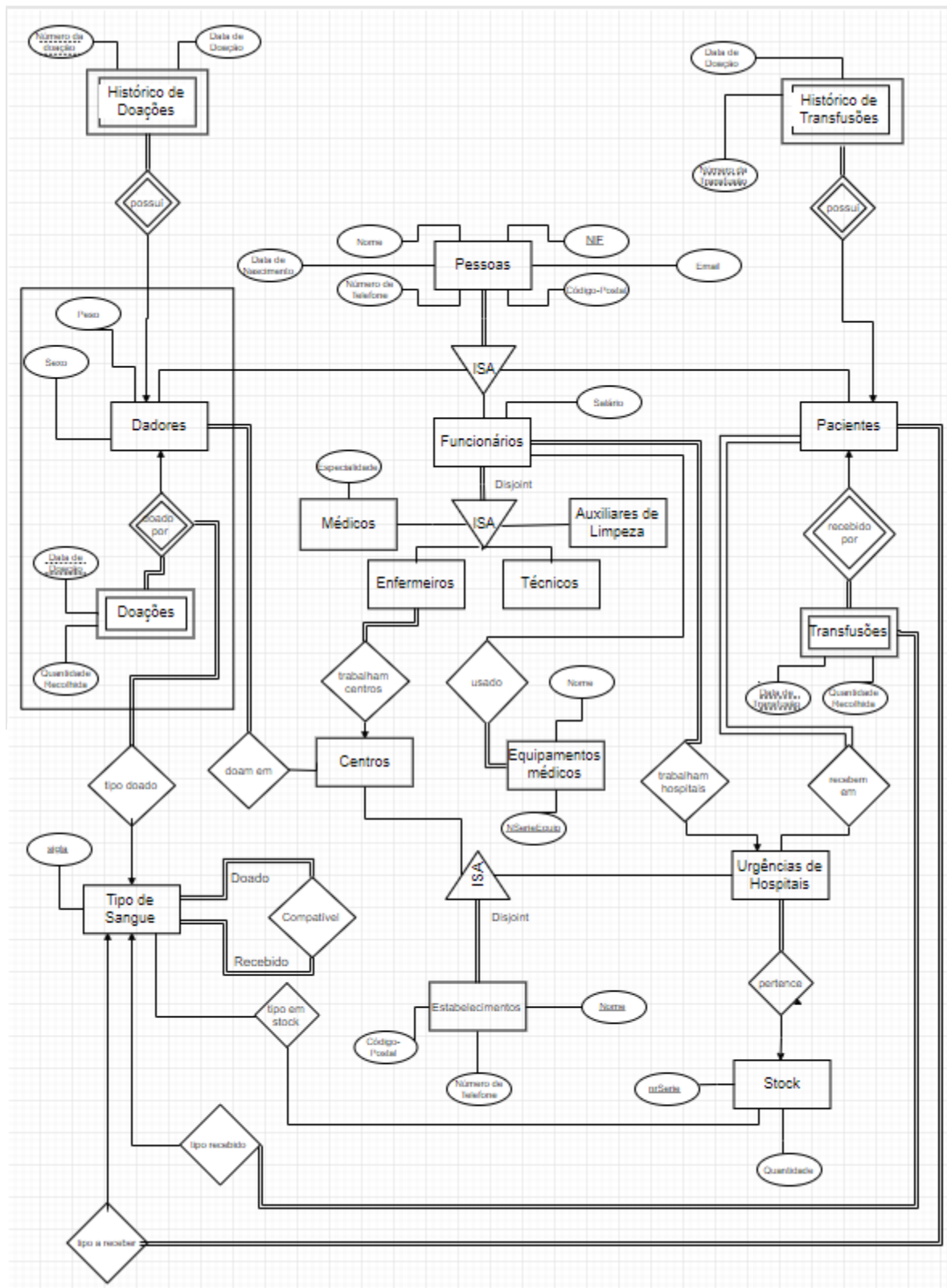
✓ Usado (NIF, Número de Serie)

- NIF é chave estrangeira referindo NIF em Funcionários.
- Número de Série é chave estrangeira referindo número de série em Equipamentos médicos.

Notas:

- As chaves primárias de todas as tabelas estão identificadas com um traço a sublinhado.
- Todas as doações feitas bem como todas as transfusões recebidas serão registadas com um número de identificação único no histórico de doações/transfusões de cada dador ou paciente respetivamente.

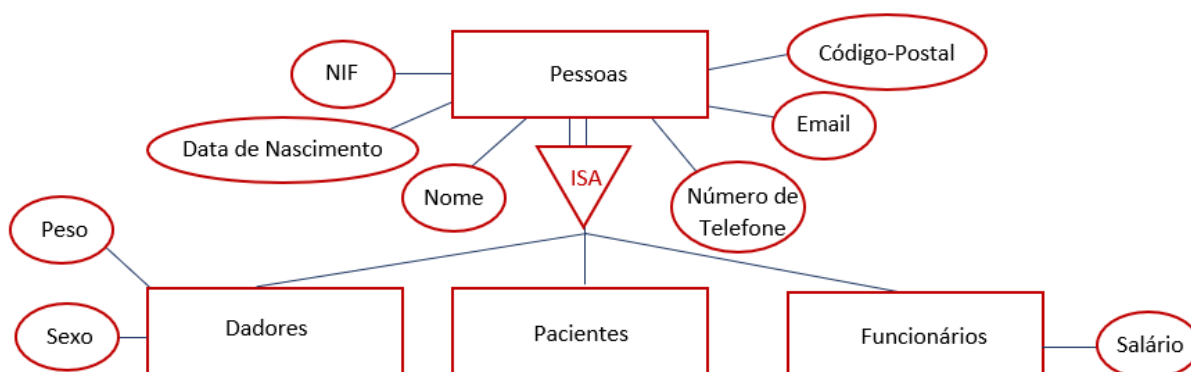
Diagrama Relacional



Modelação de Dados

Pessoas, Dadores, Funcionários e Pacientes

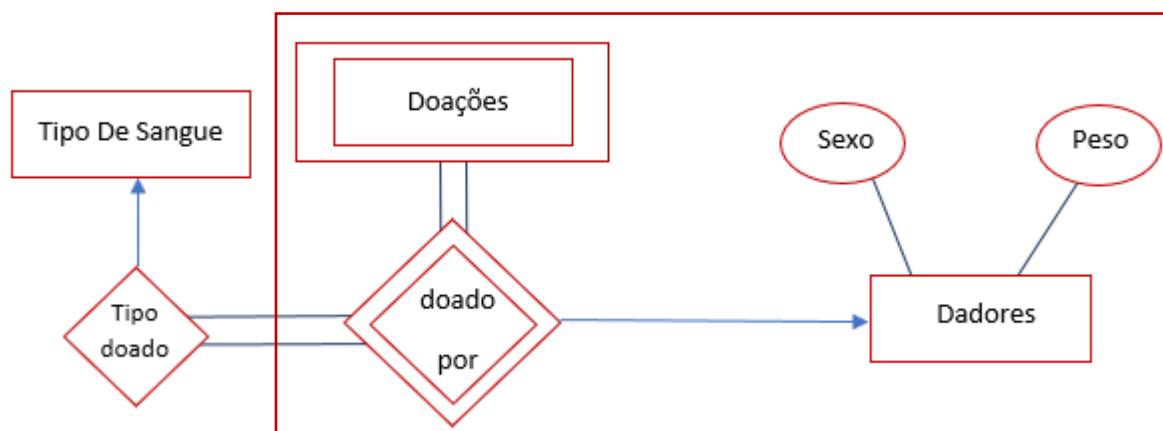
Na base de dados Centros de Doação de Sangue, existem Funcionários, Pacientes e Dadores, como tal, decidimos que a melhor implementação a tomar seria generalizar (ISA). A generalização é total tendo em conta que os únicos aderentes ao hospital que queremos manter informação são os três apresentados acima.



Dadores, Doações e Tipo de Sangue

No Centro de Doação de Sangue, cada dador poderá efetuar várias doações. Sendo esta uma relação de N:1. Quanto ao Tipo doado, as doações podem ter qualquer um dos tipos (O+, O-, AB+, AB-, A-, A+, B-, B+) associado a si, mas apenas um dos tipos por doação e dador.

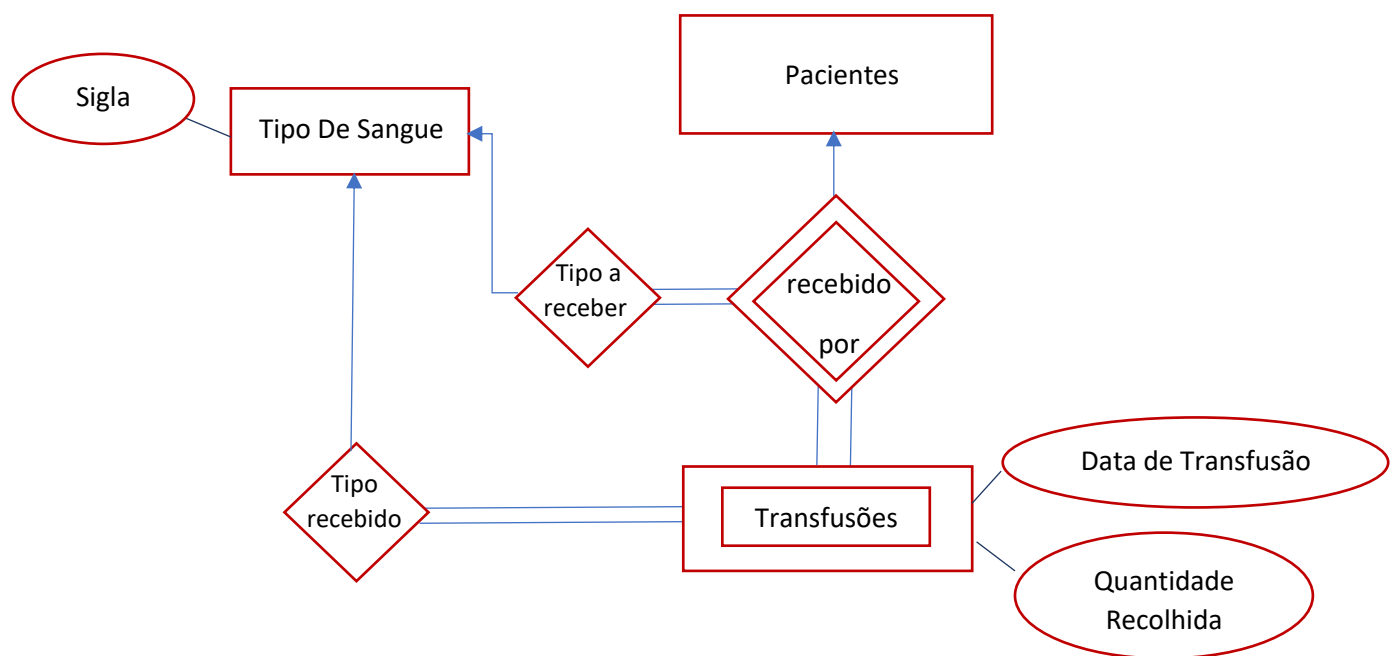
Visto que todas as doações são obrigatoriamente do tipo de sangue do dador que lhes corresponde, sem necessidade de uma confirmação de compatibilidade, foi utilizada uma agregação.



Pacientes, Transfusões e Tipos de Sangue

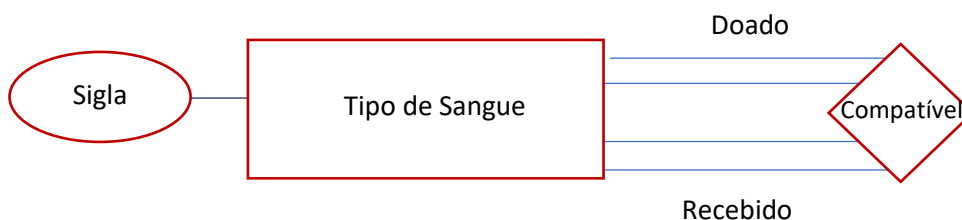
Nas urgências de um hospital, cada paciente poderá receber várias transfusões. Sendo esta uma relação de N:1. Quanto ao Tipo doado, as doações podem ter qualquer um dos tipos (O+, O-, AB+, AB-, A-, A+, B-, B+) associado a si, mas apenas um dos tipos por transfusão e paciente.

Ao fazer uma transfusão (visto que vem de outra pessoa) tem de haver uma confirmação de compatibilidade para verificar se o sangue que está a ser transferido para o paciente é compatível com o seu próprio sangue. Assim não existe uma agregação pacientes/transfusões e ambas as entendidas tem uma relação com tipos de sangue.



Tipo de Sangue

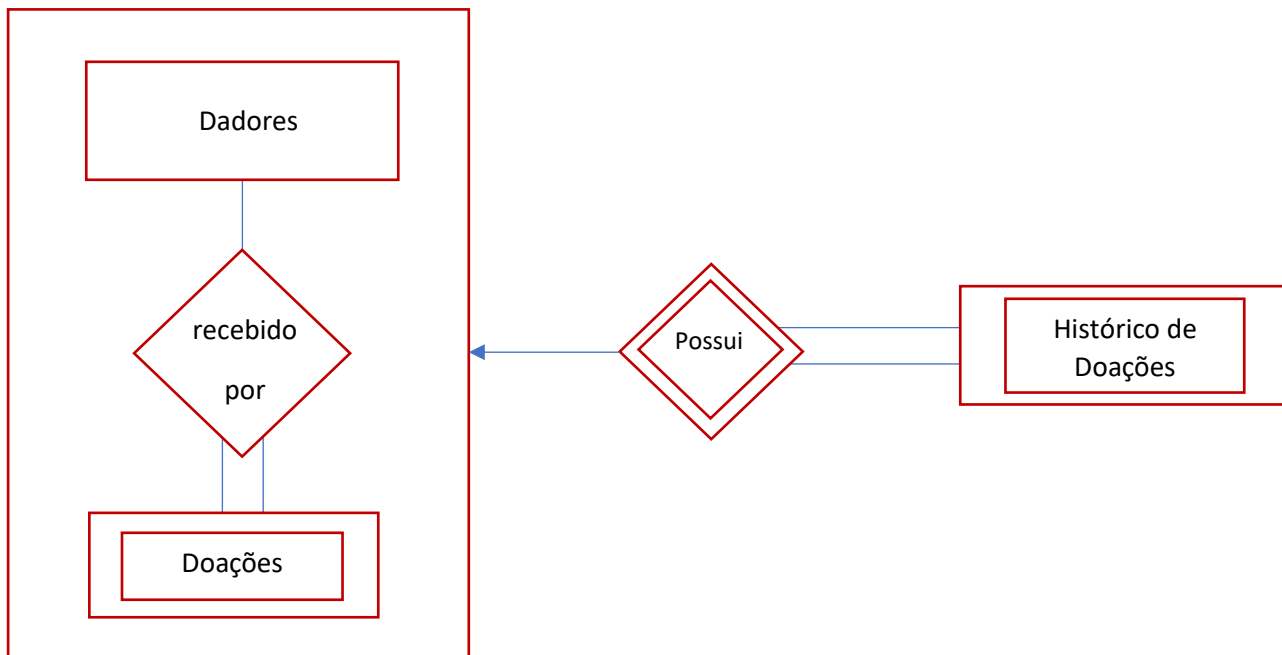
Dado um tipo de sangue, este irá ter uma relação consigo mesmo para verificar a compatibilidade com outro tipo (ou o mesmo).



Doações, Dadores e Histórico de Doações

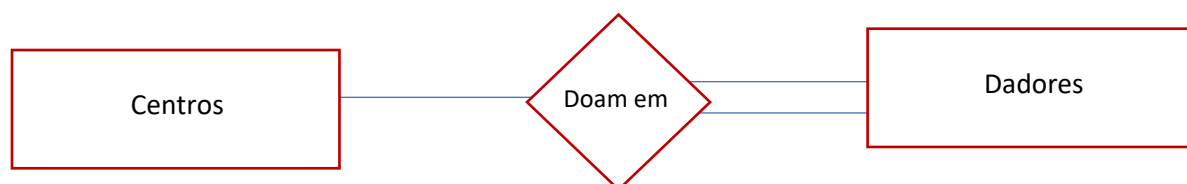
Cada Dador poderá realizar várias Doações, sendo que para cada doação realizada será inserida a informação no Histórico de Doações do Dador.

Considerámos que um dador é alguém inscrito na base de dados e que não tem obrigatoriamente de ter feito doações. Sendo assim nem todos os dadores têm um histórico de doações.



Centros e Dadores

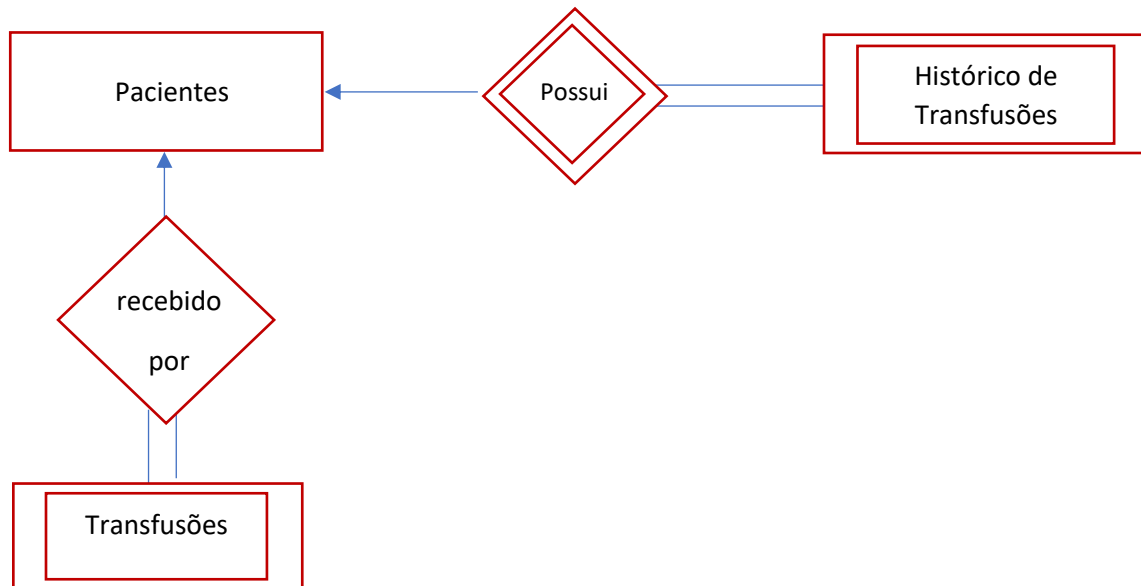
Os Dadores apenas podem realizar doações em Centros. Os Dadores não têm nenhum associado permanentemente, podendo assim doar em qualquer um deles.



Pacientes, Transfusões e Histórico de Transfusões

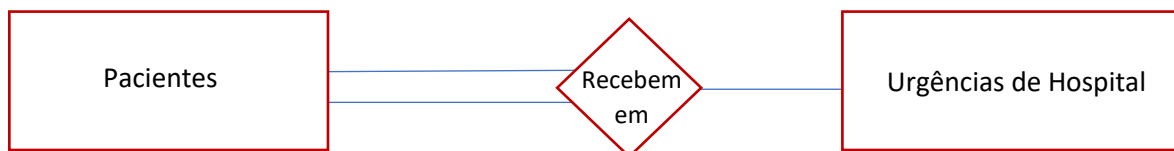
Cada Paciente poderá receber várias transfusões, sendo que para cada transfusão realizada será inserida a informação no Histórico de Transfusões do Paciente.

Para ser considerado um paciente não é preciso ter recebido uma transfusão de sangue, assim nem todos os pacientes terão um histórico de transfusões.



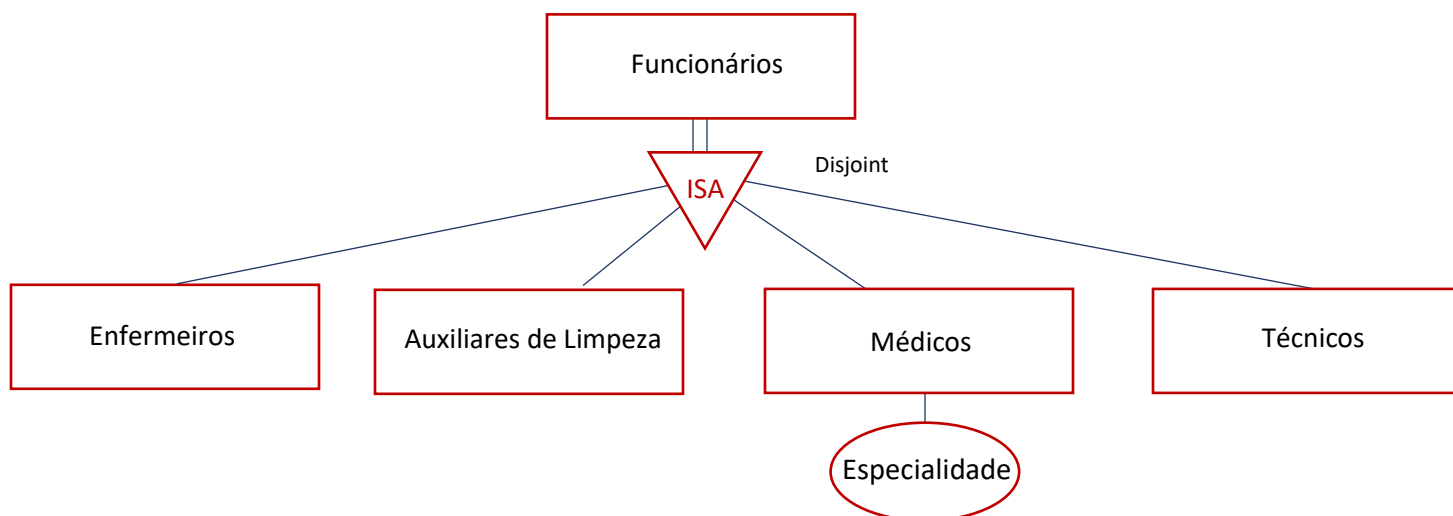
Pacientes, Urgências de Hospital

Os Pacientes apenas podem receber transfusões em Urgências de Hospital. Os Pacientes não têm nenhuma urgência associada permanentemente, podendo assim receber em qualquer uma delas.



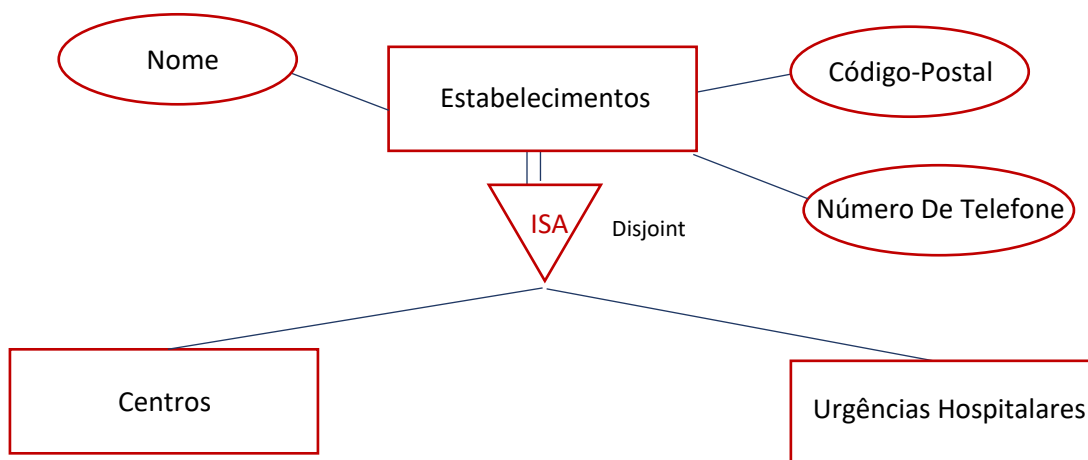
Funcionários, Enfermeiros, Técnicos e Auxiliares de Limpeza

Na base de dados, os Funcionários podem ocupar uma das quatro funções, sendo assim decidimos que a melhor implementação seria especializar (ISA) os Funcionários em Enfermeiros, Auxiliares de Limpeza, Médicos e Técnicos. Esta implementação não permite que um funcionário tenha mais de um cargo. A generalização é total tendo em conta que todos os elementos dos Funcionários têm de ser de uma das funções obrigatoriamente.



Estabelecimentos, Centros e Urgências Hospitalares

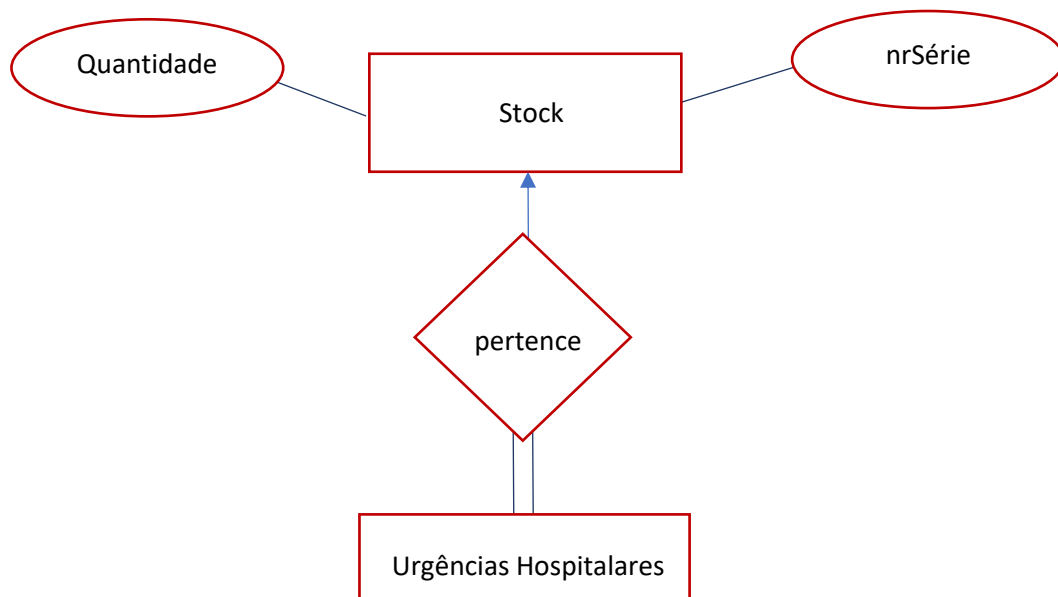
No Centro de Doação de Sangue, os Estabelecimentos podem ser ou Centros ou Urgências, sendo assim decidimos que a melhor implementação a tomar seria especializar (ISA). Esta implementação não permite que um Estabelecimento seja, em simultâneo, Urgência e Centro. A generalização é total tendo em conta que todos os elementos dos Estabelecimentos têm de ser ou um outro obrigatoriamente.



Stock e Urgência. Hospitalar

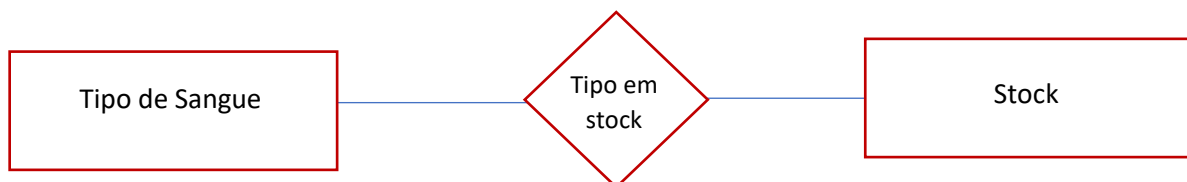
No Centro De Doação de Sangue, cada Urgência possui um Stock que irá guardar o sangue recebido nas doações e que serão utilizados nas transfusões. Cada stock terá uma quantidade máxima.

Cada stock tem disponível vários tipos de sangue, partindo do pressuposto que todos os tipos de sangue estão sempre disponíveis. Todos os stock têm um número de série geral, o que permite controlar mais facilmente todos os stocks dos vários hospitais da base de dados.



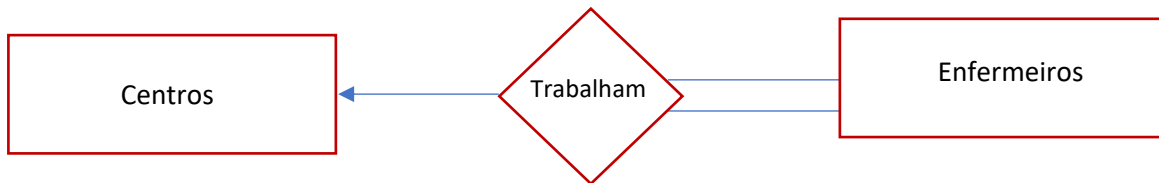
Tipo em Stock

Cada tipo de sangue pode estar em vários stocks em diferentes urgências bem como cada stock pode ter vários tipos de sangue diferentes.



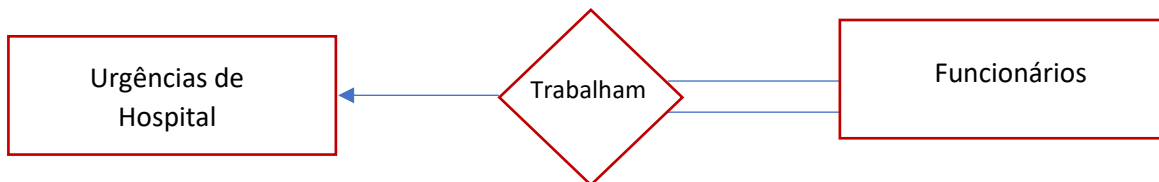
Centros e Enfermeiros

Os Enfermeiros têm associado um Centro, apenas podendo trabalhar nesse mesmo.



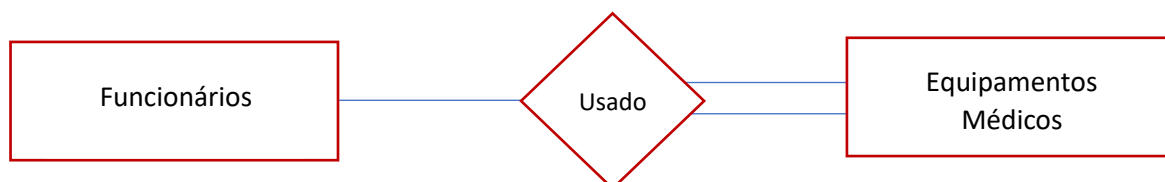
Urgências de Hospital e Funcionários

Os Funcionários têm associado um Hospital, apenas podendo trabalhar nesse mesmo. Médicos, técnicos e auxiliares de limpeza apenas podem trabalhar em urgências, enquanto que os enfermeiros podem trabalhar tanto nas urgências como nos centros.



Funcionários e Equipamentos Médicos

Os Funcionários têm à sua disposição diversos equipamentos que podem usar durante o seu trabalho. Todos os equipamentos têm de ser usados por um funcionário.



Análise das Dependências Funcionais e da Forma Normal

As dependências funcionais que o esquema relacional apresenta são as seguintes:

- ✓ NIF -> Código-Postal, Nome, Número de Telefone, Email, Peso, Data de Nascimento, Sexo, Sigla
- ✓ Data de Doação, NIF -> Quantidade Recolhida, Sigla
- ✓ NIF, Número da doação -> Data de doação
- ✓ Data de Transfusão, NIF -> Quantidade Recolhida, Sigla
- ✓ Número da Transfusão, NIF -> Data de Transfusão
- ✓ Nome Estabelecimento -> Número de Telefone, Código-postal, nrSérie
- ✓ NIF -> Salário, Nome Estabelecimento
- ✓ NIF -> Especialidade
- ✓ nrSerieEquip -> Nome do Equipamento
- ✓ nrSérie-> Quantidade em stock
- ✓ NIF -> Nome Estabelecimento

Assim podemos verificar que o esquema relacional apresentado acima se encontra na Forma Normal de Boyce-Codd (BCNF), relativamente ao conjunto de dependências anteriores, pois todas estas são da forma $\alpha \rightarrow \beta$ onde α é superchave de um dos vários esquemas que constituem o esquema relacional, deste modo α identifica univocamente todos os atributos de β .

Concluimos que esta seria a melhor forma de apresentar o esquema relacional pois não só evita redundâncias, mas também permite uma decomposição sem perdas de informação. Para além disso o esquema preserva as dependências funcionais.

Código SQL

Criação de tabelas

- Cria a tabela Pessoas

```
drop table pessoas cascade constraints;

create table pessoas (
NIF number(9) primary key,
codigoPostal varchar(8) not null,
nomePessoa varchar(100) not null,
nTelefone number(9) not null,
email varchar(100) not null,
dataNasc date not null
);
```

- Cria a tabela de Dadores

```
drop table dadores cascade constraints;

create table dadores (
NIF number(9) primary key,
peso number not null check (peso >= 50),
sexo char(1) not null check (upper(sexo) in ('F', 'M')),
foreign key (NIF) references pessoas
);
```

- Cria a tabela de Tipos de Sangue

```
drop table tiposDeSangue cascade constraints;

create table tiposDeSangue(
sigla varchar(3) primary key
    check (upper(sigla) in ('A+', 'O+', 'B+', 'O-', 'A-', 'AB+', 'B-', 'AB-'))
);
```

- Cria a tabela Compatível

```
drop table compativel cascade constraints;

create table compativel(
sigla varchar(3),
siglaCompativel varchar(3),
primary key (sigla, siglaCompativel),
foreign key (sigla) references tiposDeSangue,
foreign key (siglaCompativel) references tiposDeSangue
);
```

- Cria a tabela Pacientes

```
drop table pacientes cascade constraints;

create table pacientes(
NIF number(9) primary key,
sigla varchar(3) not null,
foreign key (NIF) references pessoas,
foreign key (sigla) references tiposDeSangue
);
```

- Cria a tabela Doações

```
drop table doacoes cascade constraints;

create table doacoes (
dataDoacao date not null,
quantidadeRecolhida number not null check (quantidadeRecolhida > 0 and quantidadeRecolhida <= 450),
NIF number(9) not null,
sigla varchar(3) not null,
primary key(dataDoacao,NIF),
foreign key (NIF) references dadores,
foreign key (sigla) references tiposDeSangue
);
```

- Cria a tabela Histórico de Doações

```
drop table historicoDeDoacoes cascade constraints;

create table historicoDeDoacoes(
NIF number(9),
numeroDaDoacao number check (numeroDaDoacao>=0),
dataDoacao date not null,
primary key(numeroDaDoacao,NIF),
foreign key(NIF) references dadores
);
```

- Cria a tabela de Transfusões

```
drop table transfusoes cascade constraints;

create table transfusoes(
dataTransfusao date,
quantidadeRecolhida number not null check (quantidadeRecolhida>0 and quantidadeRecolhida<=450),
NIF number(9),
sigla varchar(3) not null,
primary key(dataTransfusao,NIF),
foreign key (NIF) references pacientes,
foreign key (sigla) references tiposDeSangue
);
```


- Cria a tabela de Histórico de Transfusões

```
drop table historicoDeTranfusoos cascade constraints;  
  
create table historicoDeTranfusoos(  
  NIF number(9),  
  numeroDeTransfusao number check (numeroDeTransfusao >= 0),  
  dataTransfusao date not null,  
  primary key (numeroDeTransfusao, NIF),  
  foreign key (NIF) references pacientes  
);
```

- Cria a tabela de Estabelecimentos

```
drop table estabelecimentos cascade constraints;  
  
create table estabelecimentos(  
  nomeEstabelecimento varchar(100) primary key,  
  nTelefone number(9) not null,  
  codigoPostal varchar(8) not null  
);
```

- Cria a tabela de Urgências de Hospital

```
drop table urgenciasDeHospital cascade constraints;  
  
create table urgenciasDeHospital(  
  nomeEstabelecimento varchar(100) primary key,  
  nrSerie number not null,  
  foreign key (nrSerie) references stock,  
  foreign key (nomeEstabelecimento) references estabelecimentos  
);
```

- Cria a tabela de Centros

```
drop table centros cascade constraints;  
  
create table centros(  
  nomeEstabelecimento varchar(100) primary key,  
  foreign key (nomeEstabelecimento) references estabelecimentos  
);
```

- Cria a tabela de Funcionários

```
drop table funcionarios cascade constraints;
create table funcionarios (
NIF number(9) primary key,
salario number(7,2) not null check(salario>0),
nomeEstabelecimento varchar(100),
foreign key (nomeEstabelecimento) references urgenciasDeHospital,
foreign key (NIF) references pessoas
);
```

- Cria a tabela de Médicos

```
drop table medicos cascade constraints;
create table medicos(
NIF number(9) primary key,
especialidade varchar(100) not null,
foreign key (NIF) references funcionarios
);
```

- Cria a tabela de Auxiliares de Limpeza

```
drop table auxiliaresDeLimpeza cascade constraints;
create table auxiliaresDeLimpeza(
NIF number(9) primary key,
foreign key (NIF) references funcionarios
);
```

- Cria a tabela de Técnicos

```
drop table tecnicos cascade constraints;
create table tecnicos(
NIF number(9) primary key,
foreign key (NIF) references funcionarios
);
```

- Cria a tabela de Enfermeiros

```
drop table enfermeiros cascade constraints;
create table enfermeiros(
NIF number(9) primary key,
nomeEstabelecimento varchar(100),
foreign key (nomeEstabelecimento) references centros,
foreign key (NIF) references funcionarios
);
```

- Cria a tabela de Equipamentos Médicos

```
drop table equipamentosMedicos cascade constraints;
create table equipamentosMedicos (
  nomeMaquina varchar(100) not null,
  nrSerieEquip number primary key
);
```

- Cria a tabela de Stock

```
drop table stock cascade constraints;
create table stock (
  nrSerie number primary key,
  quantidade number not null check (quantidade >= 0)
);
```

- Cria a tabela de Tipo em Stock

```
drop table tipoEmStock cascade constraints;
create table tipoEmStock (
  sigla varchar(3),
  nrSerie number,
  primary key (sigla, nrSerie),
  foreign key (sigla) references tiposDeSangue,
  foreign key (nrSerie) references stock
);
```

- Cria a tabela de Doem Em

```
drop table doamEm cascade constraints;
create table doamEm (
  NIF number(9) primary key,
  nomeEstabelecimento varchar(100) not null,
  foreign key (NIF) references dadores,
  foreign key (nomeEstabelecimento) references centros
);
```

- Cria a tabela de Recebem em

```
drop table recebemEm cascade constraints;
create table recebemEm (
  NIF number(9) primary key,
  nomeEstabelecimento varchar(100) not null,
  foreign key (NIF) references pacientes,
  foreign key (nomeEstabelecimento) references urgenciasDeHospital
);
```

- Cria a tabela de Usado

```
drop table usado cascade constraints;
create table usado (
  NIF number(9) not null,
  numeroSerie number primary key,
  foreign key (NIF) references funcionarios,
  foreign key (numeroSerie) references equipamentosMedicos
);
```

Sequências

- Sequência criada para gerar o número da próxima doação

```
drop sequence seq_num_doacao;
create sequence seq_num_doacao increment by 1 start with 1;
```

- Sequência criada para gerar o número da próxima transfusão

```
drop sequence seq_num_transfusao;
create sequence seq_num_transfusao increment by 1 start with 1;
```

Triggers

- Verifica se o dado a inserir no sistema tem mais de 18 anos se não tiver lança exceção caso contrário é inserido com sucesso.

```
create or replace trigger menos_65
before insert on dados
referencing new as nrow
for each row
declare exist number; age number;
begin
  select count(*) into exist
  from pessoas
  where pessoas.nif = :nrow.nif;

  select SYSDATE-dataNasc into age
  from pessoas
  where pessoas.nif = :nrow.nif;

  if(exist = 0)
  then Raise_Application_Error (-20100, 'Nao existe pessoa no sistema');
  end if;

  if(age/365 > 65)
  then Raise_Application_Error (-20100, 'Demasiado velho');
  end if;
end;
/
```

- Verificar se o dado a inserir no sistema tem menos de 65 anos se tiver lança se exceção caso contrário é inserido com sucesso.

```
create or replace trigger mais_18
before insert on dados
referencing new as nrow
for each row
declare exist number; age number;
begin
    select count(*) into exist
    from pessoas
    where pessoas.nif = :nrow.nif;

    select SYSDATE-dataNasc into age
    from pessoas
    where pessoas.nif = :nrow.nif;

    if(exist = 0)
    then Raise_Application_Error (-20100, 'Nao existe pessoa no sistema');
    end if;

    if(age/365 < 18)
    then Raise_Application_Error (-20100, 'Demasiado novo');
    end if;
end;
/
```

- Verifica se a mulher que quer fazer uma doação não fez uma doação há menos de 120 dias se o tiver feito é lançada uma exceção caso contrário a doação é realizada.

```
create or replace trigger tempoEntreDoacoesMulheres
before insert on doacoes
referencing new as nrow
for each row
declare ultimaDoacao date;
begin
    select MIN(dataDoacao) into ultimaDoacao
    from historicoDeDoacoes, dados
    where historicoDeDoacoes.nif = :nrow.nif and dados.nif = :nrow.nif and dados.sexo = 'F';

    if ((:nrow.dataDoacao - ultimaDoacao) < 120)
    then RAISE_APPLICATION_ERROR (-20100, 'Limite minimo de 120 dias nao cumprido');
    end if;
end;
/
```

- Verifica se o homem que quer fazer uma doação não fez uma doação há menos de 90 dias se o tiver feito é lançada uma exceção caso contrário a doação é realizada.

```
create or replace trigger tempoEntreDoacoesHomens
before insert on doacoes
referencing new as nrow
for each row
declare ultimaDoacao date;
begin
    select MIN(dataDoacao) into ultimaDoacao
    from historicoDeDoacoes, dadores
    where historicoDeDoacoes.nif = :nrow.nif and dadores.nif = :nrow.nif and dadores.sexo = 'M';

    if ((:nrow.dataDoacao - ultimaDoacao) < 90)
    then RAISE_APPLICATION_ERROR (-20100, 'Limite minimo de 90 dias nao cumprido');
    end if;
end;
/
```

- Verifica se o funcionário a inserir no sistema já existe. Se existir lançasse exceção caso contrário é adicionado com sucesso no nosso sistema.

```
create or replace trigger funcionario_existe
before insert on funcionarios
referencing new as nrow
for each row
declare exist number;
begin
    select count(*) into exist
    from funcionarios
    where funcionarios.nif = :nrow.nif;
    if (exist != 0 )
    then Raise_Application_Error (-20100, 'Ja existe um funcionario com o NIF inserido!');
    end if;
end;
/
```

- Verifica se o dador a inserir no sistema já existe. Se existir lançasse exceção caso contrário é adicionado com sucesso no nosso sistema.

```
create or replace trigger dador_existe
before insert on dadores
referencing new as nrow
for each row
declare exist number;
begin
    select count(*) into exist
    from dadores
    where dadores.nif = :nrow.nif;
    if (exist != 0 )
    then Raise_Application_Error (-20100, 'Ja existe um dador com o NIF inserido!');
    end if;
end;
/
```

- Verifica se o paciente a inserir no sistema já existe. Se existir lançasse exceção caso contrário é adicionado com sucesso no nosso sistema.

```
create or replace trigger paciente_existe
before insert on pacientes
referencing new as nrow
for each row
declare exist number;
begin
    select count(*) into exist
    from pacientes
    where pacientes.nif = :nrow.nif;
    if (exist != 0 )
    then Raise_Application_Error (-20100, 'Ja existe um paciente com o NIF inserido!');
    end if;
end;
/
```

- Verifica se o funcionário a inserir no sistema já tem outro emprego. Se tiver lançasse exceção caso contrário é adicionado com sucesso no nosso sistema.

```
create or replace trigger um_emplogo
before insert on funcionarios
referencing new as nrow
for each row
declare exist number;
begin
    select count(*) into exist
    from funcionarios
    where funcionarios.nif = :nrow.nif;

    if (exist != 0)
    then Raise_Application_Error (-20100, 'Ja tem emprego!');
    end if;
end;
/
```

- Verifica se a pessoa que está a fazer uma doação se é um dador. Se for a doação é feita caso contrário é lançada uma exceção.

```
create or replace trigger doacao_by_dador
before insert on doacoes
referencing new as nrow
for each row
declare exist number;
begin
    select count(*) into exist
    from dadores
    where dadores.nif = :nrow.nif;

    if (exist = 0)
    then Raise_Application_Error (-20100, 'Dador nao existente!');
    end if;
end;
/
```

- Verifica se a pessoa que está a fazer uma transfusão é um paciente . Se for a transfusão é feita caso contrário é lançada uma exceção.

```
create or replace trigger transfusao_by_paciente
before insert on transfusoes
referencing new as nrow
for each row
declare exist number;
begin
    select count(*) into exist
    from pacientes
    where pacientes.nif = :nrow.nif;

    if (exist = 0)
    then Raise_Application_Error (-20100, 'Paciente nao existente!');
    end if;
end;
/
```

- Remover um paciente. Remove apenas da tabela de pacientes se o paciente também for um funcionário caso contrário remove da tabela de paciente e da tabela de pessoas.

```
create or replace trigger remover_paciente
before delete on pacientes
referencing new as nprow
for each row
declare existe_paciente number; tmb_funcionario number;
begin
    select count(*) into existe_paciente
    from pacientes
    where pacientes.nif = :nprow.nif;

    select count(*) into tmb_funcionario
    from funcionarios
    where funcionarios.nif = :nprow.nif;

    if (existe_paciente = 0)
    then Raise_Application_Error (-20100, 'Paciente nao existente!');
    end if;

    if (existe_paciente != 0 and tmb_funcionario = 0)
    then delete from pacientes where pacientes.nif = :nprow.nif;
    end if;

    if (existe_paciente != 0 and tmb_funcionario = 0)
    then delete from pessoas where pessoas.nif = :nprow.nif;
    end if;

    if (existe_paciente != 0 and tmb_funcionario != 0)
    then delete from pacientes where pacientes.nif = :nprow.nif;
    end if;
end;
/
```


- Remover um funcionário. Remove apenas da tabela de funcionários se o funcionário também for um dador ou um paciente. No caso em que ele não é nem paciente nem dador remove da tabela de funcionários e da tabela de pessoas.

```
create or replace trigger remover_funcionario
before delete on funcionarios
referencing new as nfrow
for each row
declare existe_funcionario number; tmb_dador number; tmb_paciente number;
begin
    select count(*) into existe_funcionario
    from funcionarios
    where funcionarios.nif = :nfrow.nif;

    select count(*) into tmb_dador
    from dadores
    where dadores.nif = :nfrow.nif;

    select count(*) into tmb_paciente
    from pacientes
    where pacientes.nif = :nfrow.nif;

    if (existe_funcionario = 0)
    then Raise_Application_Error (-20100, 'Funcionario nao existente!');
    end if;

    if (existe_funcionario != 0 and tmb_dador = 0 and tmb_paciente = 0)
    then delete from funcionarios where funcionarios.nif = :nfrow.nif;
    end if;

    if (existe_funcionario != 0 and tmb_dador = 0 and tmb_paciente = 0)
    then delete from pessoas where pessoas.nif = :nfrow.nif;
    end if;

    if ((existe_funcionario != 0 and tmb_dador != 0) or (existe_funcionario != 0 and tmb_paciente != 0))
    then delete from funcionarios where funcionarios.nif = :nfrow.nif;
    end if;
end;
/
```

- Remover um dador. Remove apenas da tabela de dadores se o dador também for funcionário. No caso em que ele não é num funcionário remove da tabela de dadores e da tabela de pessoas.

```
create or replace trigger remover_dador
before delete on dadores
referencing new as ndrow
for each row
declare existe_dador number; tmb_funcionario number;
begin
    select count(*) into existe_dador
    from dadores
    where dadores.nif = :ndrow.nif;

    select count(*) into tmb_funcionario
    from funcionarios
    where funcionarios.nif = :ndrow.nif;

    if (existe_dador = 0)
    then Raise_Application_Error (-20100, 'Dador nao existente!');
    end if;

    if (existe_dador != 0 and tmb_funcionario = 0)
    then delete from dadores where dadores.nif = :ndrow.nif;
    end if;

    if (existe_dador != 0 and tmb_funcionario != 0)
    then delete from dadores where dadores.nif = :ndrow.nif;
    end if;
end;
/
```

- Dador passa a ser paciente

```
create or replace trigger dador_agora_paciente
before insert on pacientes
referencing new as nrow
for each row
declare exist_dador number;
begin
    select count(*) into exist_dador
    from dadores
    where dadores.nif = :nrow.nif;

    if (exist_dador != 0)
    then delete from dadores where dadores.nif = :nrow.nif;
    end if;
end;
/
```

- Sempre que é realizada uma doação adiciona-la á tabela de Histórico de Doações.

```
create or replace trigger adicionarHistoricoDoacoes
after insert on doacoes
referencing new as nrow
for each row
declare exist_dador number;
begin
    select count(*) into exist_dador
    from dadores
    where dadores.nif = :nrow.nif;

    if(exist_dador != 0)
    then insert into historicoDeDoacoes values (:nrow.nif, seq_num_doacao.nextval, :nrow.dataDoacao);
    end if;
end;
/
```

- Sempre que é realizada uma transfusão adiciona-la á tabela de Histórico de Transfusões.

```
create or replace trigger adicionarHistoricoTransfusoes
after insert on transfusoes
referencing new as nrow
for each row
declare exist_paciente number;
begin
    select count(*) into exist_paciente
    from pacientes
    where pacientes.nif = :nrow.nif;

    if(exist_paciente != 0)
    then insert into historicoDeTranfusoes values (:nrow.nif, seq_num_transfusao.nextval, :nrow.dataTransfusao);
    end if;
end;
/
```

- Antes de se inserir em doamEm verifica se o nif que vai ser adicionado pertence a um dador. Senão pertencer lança exceção caso contrário é adicionado com sucesso.

```
create or replace trigger doamEm_verificacao
before insert on doamEm
referencing new as nrow
for each row
declare exist number;
begin

    select count(*) into exist
    from dadores
    where dadores.nif = :nrow.nif;

    if (exist = 0)
    then RAISE_APPLICATION_ERROR (-20100, 'Utilizador nao e dador.');
```

- Antes de se inserir em recebemEm verifica se o nif que vai ser adicionado pertence a um paciente. Senão pertencer lança exceção caso contrário é adicionado com sucesso

```
create or replace trigger recebemEm_verificacao
before insert on recebemEm
referencing new as nrow
for each row
declare exist number;
begin

    select count(*) into exist
    from pacientes
    where pacientes.nif = :nrow.nif;

    if (exist = 0)
    then RAISE_APPLICATION_ERROR (-20100, 'Utilizador nao e paciente.');
```

- Verifica se o tipo de sangue na transfusão que vai ser efetuada corresponde com o tipo de sangue do paciente que o vai receber. Se for compatível a transfusão é executada caso contrário lança exceção.

```
create or replace trigger verCompatibilidade
before insert on transfusoes
referencing new as nrow
for each row
declare exist number; pessoa_sigla varchar(3);
begin

    select pacientes.sigla into pessoa_sigla
    from pacientes
    where pacientes.nif = :nrow.nif;

    select count(*) into exist
    from compativel
    where compativel.sigla = pessoa_sigla and compativel.siglaCompativel = :nrow.sigla;

    if (exist = 0)
    then RAISE_APPLICATION_ERROR (-20100, 'Tipos de sangue incompatíveis');
    end if;
end;
/
```

Views:

```
/* View: Todos os funcionarios tem acesso aos equipamentos Medicos */
create or replace view funcionarios_view as
    select *
    from equipamentosMedicos
;

/* View: os centros tem acesso aos funcionarios que lá trabalham */
create or replace view centros_view as
    select *
    from enfermeiros
;

/* View: os enfermeiros tem acesso a todas as doacoes feitas nos centros */
create or replace view enfermeiros_view as
    select *
    from historicoDeDoacoes natural inner join dadores
;

/* View: os hospitais tem acesso a todos os funcionarios que lá trabalham */
create or replace view hospitais_view as
    select *
    from funcionarios
;

/* View: os funcionarios tem acesso a todas as transfusões feitas nos hospitais */
create or replace view funcionarios_view as
    select *
    from historicoDeTranfusoes natural inner join pacientes
;
```

Funções:

```
/* Function: Calcula o número total de doações inseridas na base de dados */
create or replace FUNCTION calc_doacoes (nif NUMBER) RETURN NUMBER
IS
    d_count NUMBER;
BEGIN
    SELECT count(*)
    INTO d_count
    FROM doacoes d
    WHERE d.nif = nif;
    RETURN (d_count);
end calc_doacoes ;

/* Function: Calcula o número total de dadores que fizeram uma doação em cada
centro de doação inserido no sistema*/

create or replace FUNCTION calc_nr_dadores (nome varchar) RETURN NUMBER
IS
    d_count NUMBER;
begin
    select count(*)
    into d_count
    from doamEm d
    where nome = d.nomeEstabelecimento;
    return (d_count);
end calc_nr_dadores;

/* Function: Calcula o número total de pacientes que fizeram uma transfusão em cada
urgência de hospital inserida no sistema*/

create or replace FUNCTION calc_nr_pacientes (nome varchar) RETURN NUMBER
IS
    d_count NUMBER;
begin
    select count(*)
    into d_count
    from recebemEm r
    where nome = r.nomeEstabelecimento;
    return (d_count);
end calc_nr_pacientes;

/* Function: Calcula o número total de transfusões inseridas na base de dados */

create or replace FUNCTION calc_transfusoes (nif NUMBER) RETURN NUMBER
IS
    d_count NUMBER;
BEGIN
    SELECT count(*)
    INTO d_count
    FROM transfusoes t
    WHERE nif = t.nif;
    RETURN (d_count);
end calc_transfusoes ;
```

Inserção de dados na tabela

Inserção de Pessoas

```
insert into pessoas values(1, '1000-001', 'Ana', 961111111, 'Ana@gmail.com', to_date('1991-05-17','YYYY-MM-DD'));
insert into pessoas values(2, '1000-002', 'Beatriz', 961111112, 'Beatriz@gmail.com', to_date('1992-05-18','YYYY-MM-DD'));
insert into pessoas values(3, '1000-003', 'Carlos', 961111113, 'Carlos@gmail.com', to_date('1993-05-19','YYYY-MM-DD'));
insert into pessoas values(4, '1000-004', 'Duarte', 961111114, 'Duarte@gmail.com', to_date('1774-05-20','YYYY-MM-DD'));
insert into pessoas values(5, '1000-005', 'Eduarda', 961111115, 'Eduarda@gmail.com', to_date('1995-05-21','YYYY-MM-DD'));
insert into pessoas values(6, '1000-006', 'Francisca', 961111116, 'Francisca@gmail.com', to_date('1996-05-22','YYYY-MM-DD'));
insert into pessoas values(7, '1000-007', 'Gabriel', 961111117, 'Gabriel@gmail.com', to_date('1997-05-23','YYYY-MM-DD'));
insert into pessoas values(8, '1000-008', 'Herve', 961111118, 'Herve@gmail.com', to_date('1998-05-24','YYYY-MM-DD'));
insert into pessoas values(9, '1000-009', 'Ivo', 961111119, 'Ivo@gmail.com', to_date('1999-05-25','YYYY-MM-DD'));
insert into pessoas values(10, '1000-010', 'Joana', 961111120, 'Joana@gmail.com', to_date('2000-06-26','YYYY-MM-DD'));
insert into pessoas values(11, '1000-010', 'Karim', 961111120, 'Karim@gmail.com', to_date('1980-07-04','YYYY-MM-DD'));
insert into pessoas values(12, '1000-010', 'Lucas', 961111120, 'Lucas@gmail.com', to_date('1981-08-25','YYYY-MM-DD'));
insert into pessoas values(13, '1000-010', 'Mauro', 961111120, 'Mauro@gmail.com', to_date('1982-09-13','YYYY-MM-DD'));
insert into pessoas values(14, '1000-010', 'Nadia', 961111120, 'Nadia@gmail.com', to_date('1983-10-10','YYYY-MM-DD'));
insert into pessoas values(15, '1000-010', 'Olivia', 961111120, 'Olivia@gmail.com', to_date('1984-11-18','YYYY-MM-DD'));
```

Em todos os casos anterior a pessoa é adicionada.

1	0.05	insert into pessoas values(1, '1000-001', 'Ana', 961111111,	1 row(s) inserted.	1
2	0.13	insert into pessoas values(2, '1000-002', 'Beatriz', 9611111	1 row(s) inserted.	1
3	0.08	insert into pessoas values(3, '1000-003', 'Carlos', 96111111	1 row(s) inserted.	1
4	0.00	insert into pessoas values(4, '1000-004', 'Duarte', 96111111	1 row(s) inserted.	1
5	0.01	insert into pessoas values(5, '1000-005', 'Eduarda', 9611111	1 row(s) inserted.	1
6	0.00	insert into pessoas values(6, '1000-006', 'Francisca', 96111	1 row(s) inserted.	1
7	0.00	insert into pessoas values(7, '1000-007', 'Gabriel', 9611111	1 row(s) inserted.	1
8	0.02	insert into pessoas values(8, '1000-008', 'Herve', 961111118	1 row(s) inserted.	1
9	0.00	insert into pessoas values(9, '1000-009', 'Ivo', 961111119,	1 row(s) inserted.	1
10	0.00	insert into pessoas values(10, '1000-010', 'Joana', 96111112	1 row(s) inserted.	1
11	0.01	insert into pessoas values(11, '1000-010', 'Karim', 96111112	1 row(s) inserted.	1
12	0.00	insert into pessoas values(12, '1000-010', 'Lucas', 96111112	1 row(s) inserted.	1
13	0.00	insert into pessoas values(13, '1000-010', 'Mauro', 96111112	1 row(s) inserted.	1
14	0.02	insert into pessoas values(14, '1000-010', 'Nadia', 96111112	1 row(s) inserted.	1
15	0.00	insert into pessoas values(15, '1000-010', 'Olivia', 9611111	1 row(s) inserted.	1

Inserção de Dadores

```
insert into dadores values(1, 50, 'F');  
insert into dadores values(2, 60, 'F');  
insert into dadores values(3, 70, 'M');
```

Os dadores são inseridos com sucesso.

16	0.02	insert into dadores values(1, 50, 'F')	1 row(s) inserted.	1
17	0.00	insert into dadores values(2, 60, 'F')	1 row(s) inserted.	1
18	0.01	insert into dadores values(3, 70, 'M')	1 row(s) inserted.	1
20	0.00	insert into dadores values(5, 90, 'F')	1 row(s) inserted.	1

```
insert into dadores values(1, 50, 'F');
```

Já existe o dador com o nif dado logo é lançada a exceção 'Ja existe um dador com o NIF inserido!'.

22	0.00	insert into dadores values(1, 50, 'F')	ORA-20100: Ja existe um dador com o NIF inserido! ORA-06512: at "BD50511.DADOR_EXISTE", line 7 ORA-04088: error during execution of trigger 'BD50511.DADOR_EXISTE'	.
----	------	--	--	---

```
insert into dadores values(4, 80, 'M');
```

A pessoa com o nif dado tem mais de 65 anos logo é lançada a exceção 'Demasiado velho'.

19	0.00	insert into dadores values(4, 80, 'M')	ORA-20100: Demasiado velho ORA-06512: at "BD50511.MENOS_65", line 16 ORA-04088: error during execution of trigger 'BD50511.MENOS_65'	.
----	------	--	--	---

```
insert into dadores values(10, 100, 'M');
```

A pessoa com o nif dado tem menos de 18 anos logo é lançada a exceção 'Demasiado novo'.

21	0.02	insert into dadores values(10, 100, 'M')	ORA-20100: Demasiado novo ORA-06512: at "BD50511.MAIS_18", line 16 ORA-04088: error during execution of trigger 'BD50511.MAIS_18'	.
----	------	--	---	---

Inserção de Pacientes

```
insert into pacientes values(6, 'A+');  
insert into pacientes values(7, 'AB+');  
insert into pacientes values(8, 'O+');  
insert into pacientes values(9, 'A-');
```

Os pacientes são todos inseridos com sucesso.

39	0.00	insert into pacientes values(6, 'A+')	1 row(s) inserted.	1
40	0.01	insert into pacientes values(7, 'AB+')	1 row(s) inserted.	1
41	0.00	insert into pacientes values(8, 'O+')	1 row(s) inserted.	1
42	0.00	insert into pacientes values(9, 'A-')	1 row(s) inserted.	1

```
insert into pacientes values(6, 'B-');
```

O pessoa com o nif dado já é paciente logo é lançada a exceção 'Ja existe um paciente com o NIF inserido!'.

43

0.02 insert into pacientes values(6, 'B-')

ORA-20100: Ja existe um paciente com o NIF inserido! ORA-06512: at "BD50511.PACIENTE_EXISTE", line 7 ORA-04088: error during execution of trigger 'BD50511.PACIENTE_EXISTE'

Inserção de Tipos de Sangue

```
insert into tiposDeSangue values ('A+');  
insert into tiposDeSangue values ('O+');  
insert into tiposDeSangue values ('B+');  
insert into tiposDeSangue values ('A-');  
insert into tiposDeSangue values ('O-');  
insert into tiposDeSangue values ('B-');  
insert into tiposDeSangue values ('AB+');  
insert into tiposDeSangue values ('AB-');
```

São inseridos com sucesso todos os tipos de sangue.

23	0.00	insert into tiposDeSangue values ('A+')	1 row(s) inserted.	1
24	0.01	insert into tiposDeSangue values ('O+')	1 row(s) inserted.	1
25	0.00	insert into tiposDeSangue values ('B+')	1 row(s) inserted.	1
26	0.00	insert into tiposDeSangue values ('A-')	1 row(s) inserted.	1
27	0.02	insert into tiposDeSangue values ('O-')	1 row(s) inserted.	1
28	0.00	insert into tiposDeSangue values ('B-')	1 row(s) inserted.	1
29	0.00	insert into tiposDeSangue values ('AB+')	1 row(s) inserted.	1
30	0.01	insert into tiposDeSangue values ('AB-')	1 row(s) inserted.	1

Inserção de Doações

```
insert into doacoes values (to_date('2018-05-17','YYYY-MM-DD'), 300, 1, 'A+');
insert into doacoes values (to_date('2018-05-19','YYYY-MM-DD'), 340, 2, 'B+');
insert into doacoes values (to_date('2019-05-20','YYYY-MM-DD'), 360, 2, 'B+');
insert into doacoes values (to_date('2018-05-22','YYYY-MM-DD'), 400, 3, 'O-');
insert into doacoes values (to_date('2018-05-25','YYYY-MM-DD'), 450, 5, 'O+');
insert into doacoes values (to_date('2019-05-26','YYYY-MM-DD'), 450, 5, 'O+');
```

As doações foram inseridas com sucesso.

44	0.00	insert into doacoes values (to_date('2018-05-17','YYYY-MM-DD	1 row(s) inserted.	1
46	0.00	insert into doacoes values (to_date('2018-05-19','YYYY-MM-DD	1 row(s) inserted.	1
47	0.00	insert into doacoes values (to_date('2019-05-20','YYYY-MM-DD	1 row(s) inserted.	1
48	0.02	insert into doacoes values (to_date('2018-05-22','YYYY-MM-DD	1 row(s) inserted.	1
50	0.00	insert into doacoes values (to_date('2018-05-25','YYYY-MM-DD	1 row(s) inserted.	1
51	0.01	insert into doacoes values (to_date('2019-05-26','YYYY-MM-DD	1 row(s) inserted.	1

```
insert into doacoes values (to_date('2018-05-23','YYYY-MM-DD'), 420, 3, 'O-');
```

O dador com nif dado que é do sexo masculino fez uma doação há menos de 90 dias logo este dador não pode fazer outra doação, portanto é lançada a exceção 'Limite mínimo de 90 dias não cumprido'.

49	0.00	insert into doacoes values (to_date('2018-05-23','YYYY-MM-DD	ORA-20100: Limite minimo de 90 dias nao cumprido ORA-06512: at "BD50511.TEMPOENTREDOACOESHOMENS", line 8 ORA-04088: error during execution of trigger 'BD50511.TEMPOENTREDOACOESHOMENS'
----	------	---	---

```
insert into doacoes values (to_date('2018-05-18','YYYY-MM-DD'), 320, 1, 'A+');
```

O dador com nif dado que é do sexo feminismo fez uma doação há menos de 120 dias logo este dador não pode fazer outra doação, portanto é lançada a exceção 'Limite mínimo de 120 dias não cumprido'.

45	0.01	insert into doacoes values (to_date('2018-05-18','YYYY-MM-DD	ORA-20100: Limite minimo de 120 dias nao cumprido ORA-06512: at "BD50511.TEMPOENTREDOACOESMULHERES", line 8 ORA-04088: error during execution of trigger 'BD50511.TEMPOENTREDOACOESMULHERES'
----	------	---	--

```
insert into doacoes values (to_date('2018-05-26','YYYY-MM-DD'), 450, 15, 'O+');
```

O nif dado não é dador logo não pode fazer uma doação por isso é lançada a exceção 'Dador não existente!'.

52	0.00	insert into doacoes values (to_date('2018-05-26','YYYY-MM-DD	ORA-20100: Dador nao existente! ORA-06512: at "BD50511.DOACAO_BY_DADOR", line 8 ORA-04088: error during execution of trigger 'BD50511.DOACAO_BY_DADOR'
----	------	---	--

Inserção na tabela de compatibilidade

```
insert into compativel values ('A+', 'A+');
insert into compativel values ('A+', 'O+');
insert into compativel values ('A-', 'O-');
insert into compativel values ('B+', 'O+');
insert into compativel values ('AB+', 'B+');
insert into compativel values ('O+', 'O+');
insert into compativel values ('A-', 'A-');
insert into compativel values ('AB+', 'AB+');
```

Todos os tuplos de sangue compatíveis são inseridos no sistema.

31	0.00	insert into compativel values ('A+', 'A+')	1 row(s) inserted.	1
32	0.00	insert into compativel values ('A+', 'O+')	1 row(s) inserted.	1
33	0.00	insert into compativel values ('A-', 'O-')	1 row(s) inserted.	1
34	0.02	insert into compativel values ('B+', 'O+')	1 row(s) inserted.	1
35	0.00	insert into compativel values ('AB+', 'B+')	1 row(s) inserted.	1
36	0.00	insert into compativel values ('O+', 'O+')	1 row(s) inserted.	1
37	0.02	insert into compativel values ('A-', 'A-')	1 row(s) inserted.	1
38	0.00	insert into compativel values ('AB+', 'AB+')	1 row(s) inserted.	1

Inserção de Transfusões

```
insert into transfusoes values (to_date('2018-05-17', 'YYYY-MM-DD'), 300, 6, 'A+');
insert into transfusoes values (to_date('2018-05-19', 'YYYY-MM-DD'), 340, 7, 'B+');
insert into transfusoes values (to_date('2018-05-20', 'YYYY-MM-DD'), 360, 7, 'AB+');
insert into transfusoes values (to_date('2018-05-21', 'YYYY-MM-DD'), 380, 7, 'B+');
insert into transfusoes values (to_date('2018-05-22', 'YYYY-MM-DD'), 400, 8, 'O+');
insert into transfusoes values (to_date('2018-05-23', 'YYYY-MM-DD'), 420, 8, 'O+');
insert into transfusoes values (to_date('2018-05-24', 'YYYY-MM-DD'), 440, 9, 'A-');
```

Transfusões inseridas com sucesso no sistema.

```
insert into transfusoes values (to_date('2018-05-18', 'YYYY-MM-DD'), 320, 6, 'B+');
```

O sangue do nif do paciente que vai receber a transfusão não é compatível com o sangue que lhe querem dar na transfusão logo é lançada a exceção 'Tipos de sangue incompatíveis'.

54	0.00	insert into transfusoes values (to_date('2018-05-18', 'YYYY-MM-DD'), 320, 6, 'B+');	ORA-20100: Tipos de sangue incompatíveis ORA-06512: at "BD50511.VERCOMPATIBILIDADE", line 13 ORA-04088: error during execution of trigger 'BD50511.VERCOMPATIBILIDADE'
----	------	--	--

```
insert into transfusoes values (to_date('2018-05-25','YYYY-MM-DD'), 450, 10, 'O+');
insert into transfusoes values (to_date('2018-05-26','YYYY-MM-DD'), 450, 10, 'O+');
```

O nif dado não corresponde a um paciente. Como a pessoa com o nif recebido não é um paciente não pode receber transfusões logo é lançada a exceção 'Paciente nao existente!'.

61	0.00	insert into transfusoes values (to_date('2018-05-25','YYYY-MM-DD'), 450, 10, 'O+');	ORA-20100: Paciente nao existente! ORA-06512: at "BD50511.TRANSFUSAO_BY_PACIENTE", line 8 ORA-04088: error during execution of trigger 'BD50511.TRANSFUSAO_BY_PACIENTE'
62	0.03	insert into transfusoes values (to_date('2018-05-26','YYYY-MM-DD'), 450, 10, 'O+');	ORA-20100: Paciente nao existente! ORA-06512: at "BD50511.TRANSFUSAO_BY_PACIENTE", line 8 ORA-04088: error during execution of trigger 'BD50511.TRANSFUSAO_BY_PACIENTE'

Inserção de Estabelecimentos

```
insert into estabelecimentos values ('Almada', 210000000, '2000-001');
insert into estabelecimentos values ('Benfica', 210000001, '2000-002');
insert into estabelecimentos values ('Porto', 210000002, '2000-003');
insert into estabelecimentos values ('Coimbra', 210000003, '2000-004');
```

Todos os estabelecimentos foram inseridos com sucesso no sistema.

63	0.00	insert into estabelecimentos values ('Almada', 210000000, '2000-001');	1 row(s) inserted.	1
64	0.00	insert into estabelecimentos values ('Benfica', 210000001, '2000-002');	1 row(s) inserted.	1
65	0.00	insert into estabelecimentos values ('Porto', 210000002, '2000-003');	1 row(s) inserted.	1
66	0.00	insert into estabelecimentos values ('Coimbra', 210000003, '2000-004');	1 row(s) inserted.	1

Inserção de Stocks

```
insert into stock values (1, 200000);
insert into stock values (2, 300000);
```

Todos os stocks são adicionados ao Sistema com sucesso.

67	0.00	insert into stock values (1, 200000);	1 row(s) inserted.	1
68	0.01	insert into stock values (2, 300000);	1 row(s) inserted.	1

Inserção de Urgências de Hospital

```
insert into urgenciasDeHospital values ('Almada',1);  
insert into urgenciasDeHospital values ('Porto',2);
```

Todas as urgências de hospitais são adicionadas com sucesso ao sistema.

69	0.00	insert into urgenciasDeHospital values ('Almada',1)	1 row(s) inserted.	1
70	0.00	insert into urgenciasDeHospital values ('Porto',2)	1 row(s) inserted.	1

Inserção de Centros

```
insert into centros values ('Benfica');  
insert into centros values ('Coimbra');
```

Todos os centros são adicionados com sucesso ao sistema.

71	0.02	insert into centros values ('Benfica')	1 row(s) inserted.	1
72	0.00	insert into centros values ('Coimbra')	1 row(s) inserted.	1

Inserção de Funcionários

```
insert into funcionarios values (11, 1000, 'Almada');  
insert into funcionarios values (12, 1100, 'Almada');  
insert into funcionarios values (13, 1200, 'Porto');  
insert into funcionarios values (14, 1300, 'Almada');  
insert into funcionarios values (15, 1400, 'Porto');
```

Foram inseridos com sucesso na tabela de funcionários.

```
insert into funcionarios values (11, 1000, 'Almada');
```

Já existe um funcionário com o nif dado logo é lançada a exceção 'Já tem emprego!'.

78	0.00	insert into funcionarios values (11, 1000, 'Almada')	ORA-20100: Já tem emprego! ORA-06512: at "BD50511.UM_EMPREGO", line 8 ORA-04088: error during execution of trigger 'BD50511.UM_EMPREGO'
----	------	--	---

Inserção de Médicos

```
insert into medicos values (15, 'Pediatra');
```

A inserção do médico foi realizada com sucesso.

79	0.00	insert into medicos values (15, 'Pediatra')	1 row(s) inserted.	1
----	------	---	--------------------	---

Inserção de Auxiliares de Limpeza

```
insert into auxiliaresDeLimpeza values (11);
```

A inserção do auxiliar de limpeza foi realizada com sucesso.

80	0.02	insert into auxiliaresDeLimpeza values (11)	1 row(s) inserted.	1
----	------	---	--------------------	---

Inserção de Técnicos

```
insert into tecnicos values (12);
```

```
insert into tecnicos values (11);
```

Todos os técnicos foram inseridos no sistema com sucesso.

81	0.00	insert into tecnicos values (12)	1 row(s) inserted.	1
82	0.00	insert into tecnicos values (11)	1 row(s) inserted.	1

Inserção de Enfermeiros

```
insert into enfermeiros values (13, 'Benfica');
```

```
insert into enfermeiros values (14, 'Coimbra');
```

Todos os enfermeiros foram inseridos no sistema com sucesso.

83	0.00	insert into enfermeiros values (13, 'Benfica')	1 row(s) inserted.	1
84	0.00	insert into enfermeiros values (14, 'Coimbra')	1 row(s) inserted.	1

Inserção de Equipamentos Médicos

```
insert into equipamentosMedicos values ('Maquina Raio-X', 111);
insert into equipamentosMedicos values ('Agulha', 222);
insert into equipamentosMedicos values ('Estetoscopio', 333);
insert into equipamentosMedicos values ('Microscopio', 444);
```

Todos os equipamentos médicos foram inseridos no sistema com sucesso.

85	0.00	insert into equipamentosMedicos values ('Maquina Raio-X', 111)	1 row(s) inserted.	1
86	0.00	insert into equipamentosMedicos values ('Agulha', 222)	1 row(s) inserted.	1
87	0.02	insert into equipamentosMedicos values ('Estetoscopio', 333)	1 row(s) inserted.	1
88	0.00	insert into equipamentosMedicos values ('Microscopio', 444)	1 row(s) inserted.	1

Inserção nos tipos que cada stock possui

```
insert into tipoEmStock values ('A+', 1);
insert into tipoEmStock values ('A-', 1);
insert into tipoEmStock values ('B+', 1);
insert into tipoEmStock values ('B-', 1);
insert into tipoEmStock values ('O+', 1);
insert into tipoEmStock values ('O-', 1);
insert into tipoEmStock values ('AB+', 1);
insert into tipoEmStock values ('AB-', 1);

insert into tipoEmStock values ('A+', 2);
insert into tipoEmStock values ('A-', 2);
insert into tipoEmStock values ('B+', 2);
insert into tipoEmStock values ('B-', 2);
insert into tipoEmStock values ('O+', 2);
insert into tipoEmStock values ('O-', 2);
insert into tipoEmStock values ('AB+', 2);
insert into tipoEmStock values ('AB-', 2);
```

Todos os tuplos anterior foram inseridos na tabela de tipoEmStock com sucesso.

89	0.00	insert into tipoEmStock values ('A+', 1)	1 row(s) inserted.	1
90	0.01	insert into tipoEmStock values ('A-', 1)	1 row(s) inserted.	1
91	0.00	insert into tipoEmStock values ('B+', 1)	1 row(s) inserted.	1
92	0.00	insert into tipoEmStock values ('B-', 1)	1 row(s) inserted.	1
93	0.00	insert into tipoEmStock values ('O+', 1)	1 row(s) inserted.	1
94	0.00	insert into tipoEmStock values ('O-', 1)	1 row(s) inserted.	1
95	0.01	insert into tipoEmStock values ('AB+', 1)	1 row(s) inserted.	1
96	0.00	insert into tipoEmStock values ('AB-', 1)	1 row(s) inserted.	1
97	0.00	insert into tipoEmStock values ('A+', 2)	1 row(s) inserted.	1
98	0.00	insert into tipoEmStock values ('A-', 2)	1 row(s) inserted.	1
99	0.00	insert into tipoEmStock values ('B+', 2)	1 row(s) inserted.	1
100	0.00	insert into tipoEmStock values ('B-', 2)	1 row(s) inserted.	1
101	0.00	insert into tipoEmStock values ('O+', 2)	1 row(s) inserted.	1
102	0.00	insert into tipoEmStock values ('O-', 2)	1 row(s) inserted.	1
103	0.04	insert into tipoEmStock values ('AB+', 2)	1 row(s) inserted.	1

Inserção de onde cada doador fez as doações

```
insert into doamEm values (1, 'Benfica');
insert into doamEm values (2, 'Coimbra');
insert into doamEm values (3, 'Benfica');
insert into doamEm values (5, 'Benfica');
```

Os tuplos foram inseridos com sucesso na tabela de doamEm.

105	0.00	insert into doamEm values (1, 'Benfica')	1 row(s) inserted.	1
106	0.00	insert into doamEm values (2, 'Coimbra')	1 row(s) inserted.	1
107	0.00	insert into doamEm values (3, 'Benfica')	1 row(s) inserted.	1
109	0.02	insert into doamEm values (5, 'Benfica')	1 row(s) inserted.	1

```
insert into doamEm values (4, 'Coimbra');
```

A pessoa não é dador logo não pode ser adicionado á tabela doamEm portanto é lançada a exceção 'Utilizador nao e dador.'.

108	0.00	insert into doamEm values (4, 'Coimbra')	ORA-20100: Utilizador nao e dador. ORA-06512: at "BD50511.DOAMEM_VERIFICACAO", line 9 ORA-04088: error during execution of trigger 'BD50511.DOAMEM_VERIFICACAO'	
-----	------	--	---	--

Inserção onde cada paciente recebeu uma transfusão

```
insert into recebemEm values (6, 'Almada');
insert into recebemEm values (7, 'Almada');
insert into recebemEm values (8, 'Almada');
insert into recebemEm values (9, 'Almada');
```

Os tuplos foram inseridos com sucesso na tabela de recebemEm.

110	0.00	insert into recebemEm values (6, 'Almada')	1 row(s) inserted.	1
111	0.00	insert into recebemEm values (7, 'Almada')	1 row(s) inserted.	1
112	0.01	insert into recebemEm values (8, 'Almada')	1 row(s) inserted.	1
113	0.00	insert into recebemEm values (9, 'Almada')	1 row(s) inserted.	1

```
insert into recebemEm values (10, 'Almada');
```

A pessoa não é paciente logo não pode ser adicionado á tabela recebemEm portanto é lançada a exceção 'Utilizador nao e paciente.'.

114	0.00	insert into recebemEm values (10, 'Almada')	ORA-20100: Utilizador nao e paciente. ORA-06512: at "BD50511.RECEBEMEM_VERIFICACAO", line 9 ORA-04088: error during execution of trigger 'BD50511.RECEBEMEM_VERIFICACAO'	
-----	------	---	--	--

Inserção de cada utilização de equipamentos médicos

```
insert into usado values (11, 111);  
insert into usado values (12, 222);  
insert into usado values (14, 333);  
insert into usado values (15, 444);
```

Os tuplos foram inseridos com sucesso na tabela de usado.

115	0.02	insert into usado values (11, 111)	1 row(s) inserted.	1
116	0.00	insert into usado values (12, 222)	1 row(s) inserted.	1
117	0.00	insert into usado values (14, 333)	1 row(s) inserted.	1
118	0.01	insert into usado values (15, 444)	1 row(s) inserted.	1

Funcionalidades da Interface – Implementação

1. Existência de uma página de entrada onde existam ligações para as páginas subsequentes

A página de entrada da aplicação, contém uma página de início que liga todas as outras páginas.

Existem ao todo 37 páginas em toda a aplicação, tendo em conta a página de início e a página de login. Algumas páginas são reports e estão normalmente identificadas com o nome da tabela a que pertencem, outras páginas são forms e estão normalmente identificadas por “Inserir nome da tabela” dependendo da tabela a que correspondem.



1 - Início



2 - Pacientes



3 - Inserir Pacientes



4 - Inserir Estabelecimentos



5 - Inserir Estabelecimentos



6 - Tiposdesangue



7 - Historicodedoacoes



9 - Historicodetransfusoes



11 - DADORES



12 - Centros



14 - Medicos



15 - Auxiliaresdelimpeza



16 - Tecnicos



17 - Enfermeiros



19 - Stock



20 - Compativel



21 - Tipoemstock



22 - Doamem



23 - Recebemem



24 - Usado



25 - Pessoas



26 - Inserir Pessoas



27 - Equipamentos Médicos



28 - Inserir Equipamentos Médicos



29 - Dadores

1 - 25 ►



30 - Inserir Dadores



31 - Doações



32 - Inserir Doações



33 - Transfusões



34 - Inserir Transfusões



3 - Urgências de Hospital



44 - Inserir Urgências de Hospital



49 - Funcionários



50 - Inserir Funcionários



55 - Transfusoes



56 - Master Detail



101 - Login

2. Listagem de dados da Base de Dados onde códigos referentes a chaves externas sejam substituídos pelo valor de outros atributos de fácil compreensão:

Esta funcionalidade foi utilizada maioritariamente para acrescentar o valor nomePessoa em todas as tabelas que apenas tinham o nif da pessoa, visto nomePessoa ser de mais fácil identificação do que o seu nif.

Início > Inserir DADORES



Go

Actions ▾

Create

	<u>Nif</u>	<u>Peso</u>	<u>Sexo</u>	<u>Nomepessoa</u>
	1	50	F	Ana
	2	60	F	Beatriz
	3	70	M	Carlos
	5	90	F	Eduarda

1 - 4

Legenda: Página dados onde foi acrescentado NomePessoa através do seu nif.

Source

Region Source

```
select
"NIF",
"NOMEPESSOA",
"PESO",
"SEXO",
calc_doacoes ("NIF") "Número de Doações"
from "#OWNER#". "DADORES" natural inner join "PESSOAS"
```

Legenda: Código final para que esta mudança fosse possível.

3. Listagem de dados da Base de Dados onde sejam apresentados valores derivados:

Esta funcionalidade permite criar valores que não se conseguem obter estaticamente (através da inserção nas tabelas). Assim foram utilizados para fazer somas de dados inseridos, aqui apresentamos um dos casos:



Início > Centros	
	<input type="text"/> <input type="button" value="Go"/> <input type="button" value="Actions ▼"/>
<u>Nomeestabelecimento</u>	<u>Número De Dadores</u>
 Benfica	3
 Coimbra	1
1 - 2	

Legenda: Número de Dadores é um valor derivado que permite saber quantos dadores diferentes é que fizeram uma doação em cada centro disponível.

```
create or replace FUNCTION calc_nr_pacientes (nome varchar) RETURN NUMBER
IS
    d_count NUMBER;
begin
    select count(*)
    into d_count
    from recebemEm r
    where nome = r.nomeEstabelecimento;
    return (d_count);
end calc_nr_pacientes;
```

Legenda: Código SQL da função que permite descobrir o valor derivado.

4. Possibilidade de inserir, remover e atualizar tuplos da Base de Dados:

Em todas as tabelas (exceto aquelas que são preenchidas automaticamente) é preciso a possibilidade criar e modificar tuplos, isto é algo que o APEX faz quase automaticamente e não é preciso código específico.

Início > Inserir PESSOAS > Inserir PESSOAS

Inserir PESSOAS Cancel Create


Nif

*Codigopostal

*Nomepessoa

*Ntelefone

*Email

*Datanasc 

Legenda: Criar um novo tuplo.

Início > Inserir PESSOAS > Inserir PESSOAS

Inserir PESSOAS Cancel Delete Apply Changes


Nif

*Codigopostal

*Nomepessoa

*Ntelefone

*Email

*Datanasc 

Legenda: Atualizar/Remover um tuplo existente.

5. Possibilidade de preencher valores de atributos correspondentes a relações (chaves externas) sem se ter conhecimento de códigos (e.g. seleccionando um valor de uma LOV baseada numa consulta) e/ou restringindo o domínio dos valores seleccionados):

Esta funcionalidade poderia ser usada na aplicação em diversos contextos, mas neste caso foi utilizada para apenas permitir que seja inserida uma doação com um tipo de sangue predefinido.

The image shows a screenshot of a web application interface. At the top, there is a form titled "Inserir DOACOES" with a green header bar containing "Cancel" and "Create" buttons. The form contains several input fields: "Datadoacao" (a date picker), "Nif" (a dropdown menu), "*Quantidaderecolhida" (a text input), and "*Sigla" (a dropdown menu). The "*Sigla" dropdown is open, showing a list of blood types: A+, A-, AB+, AB-, B+, B-, O+, and O-. Below the form, there is a horizontal bar with several tabs: "Lists of Values", "Edit Null Text", "Locally Defined", "Grid Edit", "Conditional Entries", "Subscription", "Utilization", and "History". The "Lists of Values" tab is selected. Below this bar, there is a search bar with a magnifying glass icon, a "Go" button, and an "Actions" dropdown menu. At the bottom, there are four icons representing different lists: "LIST DADORES NOME", "LIST PACIENTES NOME", "LIST SANGUE", and "Report Row Per Page".

Legenda: List Sangue é uma List Of Values (LOV) que permite seleccionar apenas os tipos de sangue inserido na base.

Source: Lists of Values Query

```
Query
select sigla d, sigla r
from tiposDeSangue
order by 1
```

Legenda: Código usado na LOV acima.

6. Existência de links de navegação (breadcrumbs) nas várias páginas:

Todas as páginas foram criadas com breadcrumbs, sejam estas reports ou forms.

Page Rendering

Page

Page Name: [Pacientes](#) Template: [Application default](#)
Title: [Pacientes](#) Header Text:
HTML Header: Footer Text:
HTML Body: Build Option:
Help Text: [No help is available for this](#) Authorization: [No](#)
Page Group: Cached: [No](#)

Regions

Display Point: [Page Template Body \(3\)](#)
10 [Inserir Pacientes](#) [Interactive Report](#) [Saved Report](#)

Display Point: [Region Position 01](#)
1 >>> [Breadcrumb](#) [Breadcrumb Entry](#)

Legenda: Exemplo da página Report Pacientes.

	<u>Name</u>	<u>Sequence</u>	<u>Page</u>	<u>Parent</u>	<u>Page Exists</u>
	Inserir Pacientes	10	<u>2</u>	1. Início	Yes

1 - 1

7. Existência de dois reports interligados, onde um apresente detalhes do outro (drill-down):

Esta funcionalidade permite que ao selecionar um atributo duma tabela, o utilizador seja guiado para uma página com detalhes sobre esse atributo. No caso da nossa aplicação decidimos usar esta funcionalidade para saber informações sobre as doações feitas por um dador específico.

Para ativar esta funcionalidade basta seguir instruções do APEX não sendo necessário nenhum código extra.

Início > Inserir DADORES



	Nif	Peso	Sexo	Nomepessoa
	1	50	F	Ana
	2	60	F	Beatriz
	3	70	M	Carlos
	5	90	F	Eduarda

1 - 4


Legenda: Ao clicar com o mouse num dos nifs na tabela dadores, o utilizador é guiado para a página de Doações com informações sobre as doações feitas por este dador.



Início > Inserir DOACOES

Doações feitas

Número de Doações: 2

Selecione NIF



	Datadoacao	Quantidaderecolhida	Nif	Sigla
	19-MAY-18	340	2	B+
	20-MAY-19	360	2	B+

1 - 2

Legenda: Ao clicar com o mouse no nif 2, o utilizador é redirecionado para esta página.

8. Existência de um detalhe condicional.

O detalhe condicional desenvolvido nesta aplicação permite listar apenas as doações feitas por cada dador (sendo possível selecionar qual o dador pretendido, usando o seu nif) e mostra o número total de doações feitas por este.

Início > Inserir DOACOES

Donações feitas

Número de Doações: 2

Selecione NIF

	<u>Datadoacao</u>	<u>Quantidaderecolhida</u>	<u>Nif</u>	<u>Sigla</u>
	19-MAY-18	340	2	B+
	20-MAY-19	360	2	B+

1 - 2

Legenda: Na tabela doações foi selecionado o dador com o nif 2 e devido ao detalhe condicional é possível ver todas as doações feitas por este bem como o número total de doações desde que se inscreveu como dador. Código necessário para implementar esta funcionalidade:








```
Source
Region Source
SELECT count (*) "Número de Doações:"
FROM dadores inner join doacoes using (nif)
WHERE nvl(nif,'-1') = nvl(:P31_DOACOES,'-1')
```

NOTA: Embora não implementado, esta funcionalidade também teria sido bastante útil para o caso do histórico de doações e histórico de transfusões.

9. Existência de um form master-detail, onde seja possível inserir, remover e alterar dados (pelo menos) do detail:

A tabela Transfusões foi a que escolhemos para criar um form master-detail de forma a poder inserir, remover e alterar os seus dados.

Assim será possível editar apenas alguns atributos das transfusões (neste caso a quantidade e sigla) mas mostrando alguma informação adicional sobre este paciente, nomeadamente o tipo de sangue que tem de receber.

Transfusoes			
Transfusoes			Create
Edit	Quantidaderecolhida	Nif	Sigla
	300	6	A+
	340	7	B+
	360	7	AB+
	380	7	B+
	400	8	O+
	420	8	O+
	440	9	A-

Transfusoes > Master Detail

Edit Transfusoes
Cancel
Delete
Apply Changes
>

Datatransfusao 17-MAY-18

Nif 6

*Quantidaderecolhida

*Sigla

1 of 7

Pacientes Detail
Delete Checked
Add Row

	<u>Sigla</u>
<input type="checkbox"/>	<input type="text" value="A+"/>

1 - 1

10. Implementações que apresentem funcionalidades adicionais:

- Views

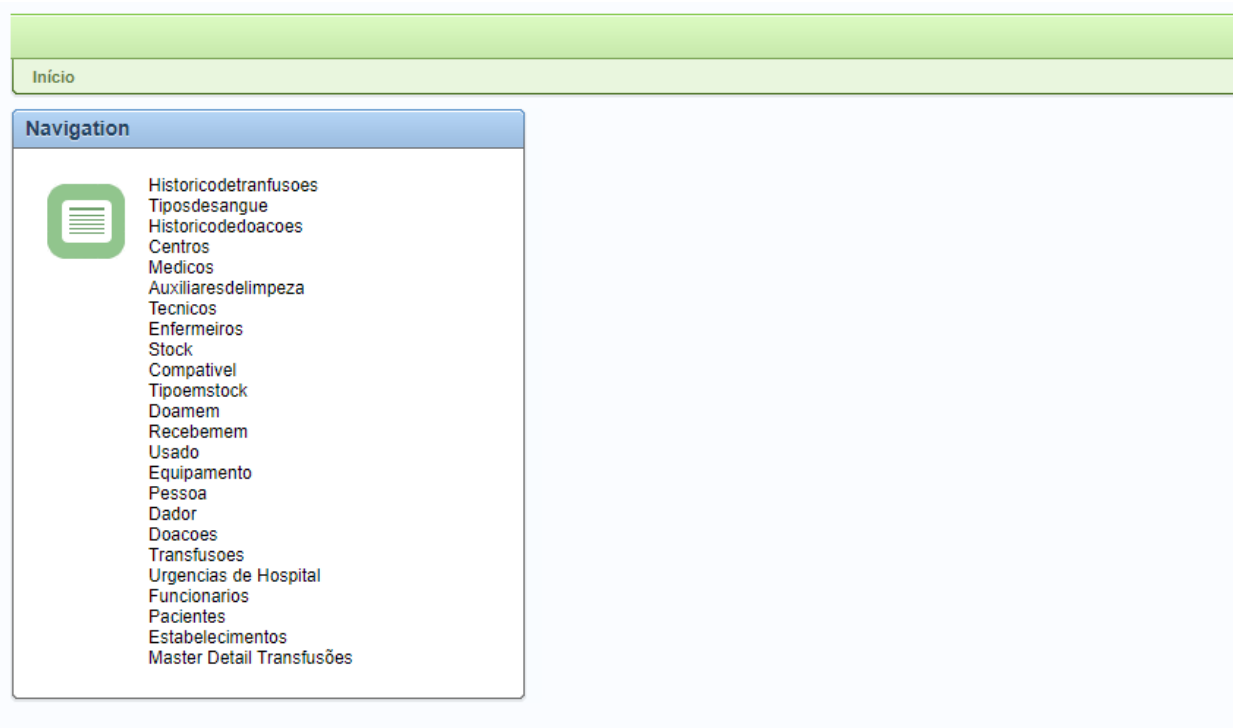
Os códigos correspondentes a todas as views criadas na aplicação já foram apresentados acima na parte do código SQL, aqui é apresentado o resultado que uma das views produz na base de dados.

ENFERMEIROS_VIEW			Create ▼
View	Code	Data	Grants UI Defaults Dependencies SQL
Compile Drop			
Column Name	Data Type	Nullable	
NIF	NUMBER(9,0)	No	
NUMERODADOACAO	NUMBER	No	
DATADOACAO	DATE	No	
PESO	NUMBER	No	
SEXO	CHAR(1)	No	

ENFERMEIROS_VIEW					Create ▼
View	Code	Data	Grants	UI Defaults	Dependencies SQL
Query	Count Rows	Insert Row			
NIF	NUMERODADOACAO	DATADOACAO	PESO	SEXO	
1	1	05/17/2018	50	F	
2	2	05/19/2018	60	F	
2	3	05/20/2019	60	F	
3	4	05/22/2018	70	M	
5	5	05/25/2018	90	F	
5	6	05/26/2019	90	F	
row(s) 1 - 6 of 6					
Download					

Manual do Utilizador

Menu inicial – O utilizador pode escolher quais as páginas que quer consultar. Todas as linhas apresentadas no menu são links que levam o utilizador à página/tabela com o nome correspondente.



Devido ao facto de esta aplicação não corresponder à interface completa, as diversas páginas podem ter funcionalidades diferentes.

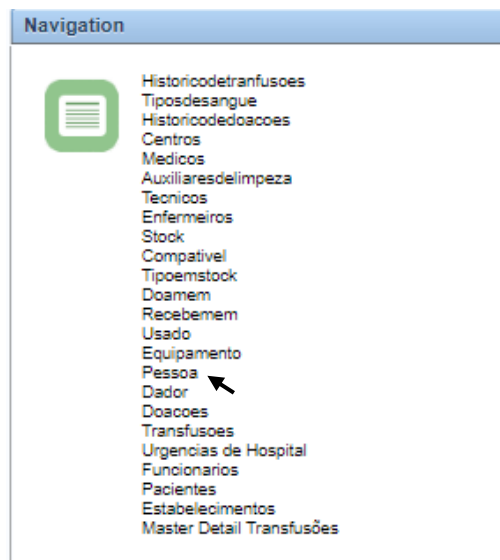
Assim as páginas “TiposDeSangue”, “Centros”, “Médicos”, “AuxiliaresDeLimpeza”, “Técnicos”, “Enfermeiros”, “Stock”, “Compatível”, “TipoEmStock”, “DoamEm”, “RecebemEm” e “Usado” são apenas de consulta. Podendo ser consultados os dados já inseridos na base de dados.

As páginas “HistóricoDeDoações” e “HistóricoDeTransfusões” também são apenas de consulta, mas os dados destas vão ser inseridos automaticamente sempre que uma doação é inserida na aplicação.

As restantes páginas têm funcionalidades extras que serão descritas abaixo.

Adicionar dados – O utilizador pode adicionar dados (tuplos) novos às páginas “Pacientes”, “Estabelecimentos”, “EquipamentosMédicos”, “Dadores”, “Doações”, “Transfusões”, “UrgênciasdeHospital” e “Funcionários” usando o botão CREATE presente em todas estas páginas. Para isto é necessário seguir os seguintes passos:

1. Selecionar a página pretendida:



2. De seguida selecionar o botão CREATE:

Início > Inserir PESSOAS						
		<input type="text"/>		Go	Actions ▼	Create
	Nif	Codigopostal	Nomepessoa	Ntelefone	Email	Datanasc
	1	1000-001	Ana	961111111	Ana@gmail.com	17-MAY-91
	2	1000-002	Beatriz	961111112	Beatriz@gmail.com	18-MAY-92
	3	1000-003	Carlos	961111113	Carlos@gmail.com	19-MAY-93
	4	1000-004	Duarte	961111114	Duarte@gmail.com	20-MAY-74
	5	1000-005	Eduarda	961111115	Eduarda@gmail.com	21-MAY-95
	6	1000-006	Francisca	961111116	Francisca@gmail.com	22-MAY-96
	7	1000-007	Gabriel	961111117	Gabriel@gmail.com	23-MAY-97
	8	1000-008	Herve	961111118	Herve@gmail.com	24-MAY-98
	9	1000-009	Ivo	961111119	Ivo@gmail.com	25-MAY-99
	10	1000-010	Joana	961111120	Joana@gmail.com	26-JUN-00
	11	1000-010	Karim	961111120	Karim@gmail.com	04-JUL-80
	12	1000-010	Lucas	961111120	Lucas@gmail.com	25-AUG-81
	13	1000-010	Mauro	961111120	Mauro@gmail.com	13-SEP-82

3. Preencher os campos:

Inserir PESSOAS

Cancel

Create

Nif

*Codigopostal

*Nomepessoa

*Ntelefone

*Email

*Datanasc

Por fim clicar de novo no botão CREATE nesta página.

Atualizar/Remover dados – O utilizador pode atualizar/remover de todas as tabelas nas quais pode inserir novos tuplos. Para isso:













1. Selecionar o tuplo que se quer remover/atualizar clicando no ícone ao lado do nif correspondente ao tuplo selecionado.

Início > Inserir PESSOAS

Go

Actions

Create

	Nif	Codigopostal	Nomepessoa	Ntelefone	Email	Datanasc
	1	1000-001	Ana	981111111	Ana@gmail.com	17-MAY-91
	2	1000-002	Beatriz	981111112	Beatriz@gmail.com	18-MAY-92
	3	1000-003	Carlos	981111113	Carlos@gmail.com	19-MAY-93
	4	1000-004	Duarte	981111114	Duarte@gmail.com	20-MAY-74
	5	1000-005	Eduarda	981111115	Eduarda@gmail.com	21-MAY-95
	6	1000-006	Francisca	981111116	Francisca@gmail.com	22-MAY-96
	7	1000-007	Gabriel	981111117	Gabriel@gmail.com	23-MAY-97
	8	1000-008	Herve	981111118	Herve@gmail.com	24-MAY-98
	9	1000-009	Ivo	981111119	Ivo@gmail.com	25-MAY-99
	10	1000-010	Joana	981111120	Joana@gmail.com	26-JUN-00
	11	1000-010	Karim	981111120	Karim@gmail.com	04-JUL-80
	12	1000-010	Lucas	981111120	Lucas@gmail.com	25-AUG-81
	13	1000-010	Mauro	981111120	Mauro@gmail.com	13-SEP-82

2. Para atualizar os dados basta escrever novas informações nos respectivos campos e de seguida clicar no botão APPLY CHANGES.

Para remover o tuplo clicar no botão DELETE.

The screenshot shows a web form titled "Inserir PESSOAS" with a green header bar. On the right side of the header are three buttons: "Cancel", "Delete", and "Apply Changes". The form contains several input fields, each preceded by a red asterisk indicating a required field. The fields are: "Nif" with the value "1", "*Codigopostal" with "1000-001", "*Nomepessoa" with "Ana", "*Ntelefone" with "961111111", "*Email" with "Ana@gmail.com", and "*Datanasc" with "17-MAY-91". A small calendar icon is visible next to the date field.

Ao inserir ou atualizar tuplos na base de dados pode acontecer que determinado atributo esteja condicionado por uma LOV (list of values). Nestes casos a única coisa a fazer é seleccionar qual o valor pretendido naquele parâmetro, de entre as opções dadas.

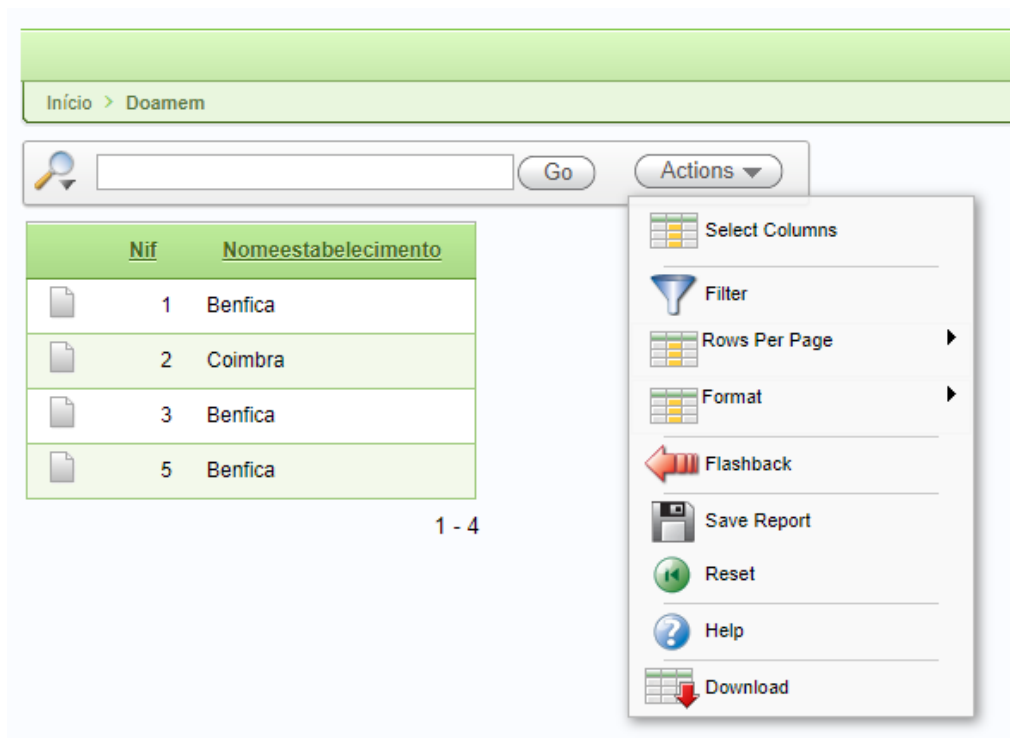
The screenshot shows a web form titled "INSERIR TRANSFUSOES" with a green header bar. On the right side of the header are two buttons: "Cancel" and "Create". The form contains several input fields. The "Datatransfusao" field has a calendar icon. The "Nif" field has a dropdown menu open, showing a list of values: "--", "6", "7", "8", and "9". The "6" option is currently selected. Other fields include "*Quantidaderecolhida" and "*Sigla", both with dropdown menus.

No exemplo acima, apenas podem ser seleccionados NIF's de pessoas que estejam inscritas no sistema como pacientes, pois apenas essas podem receber transfusões.

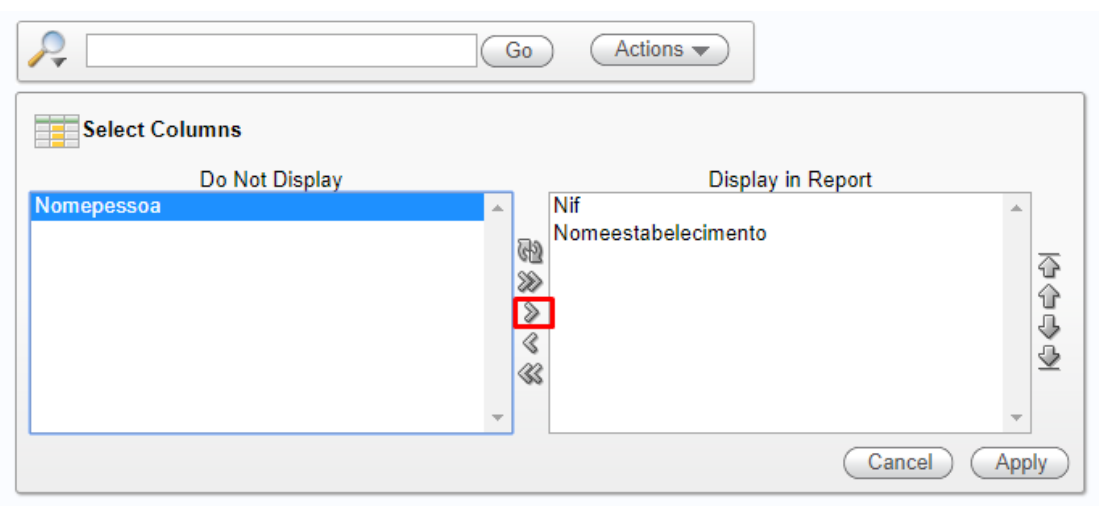
Ver os atributos de fácil compreensão – Por vezes os atributos de fácil compreensão que foram adicionados com base noutros já existentes podem não ser apresentados de imediato ao abrir uma página, nesta situação é necessário seguir os seguintes passos:

(Tomemos por exemplo a página DoamEm, onde se quer mostrar o nome do Dador)

1. Clicar em ACTIONS e de seguida em “Select Columns”:



2. Selecionar o atributo que se pretende ver na tabela e pressionar o botão rodeado a vermelho. Por fim fazer APPLY.



Resultado Final:

	<input type="text"/>	<input type="button" value="Go"/>	<input type="button" value="Actions ▼"/>
	<u>Nif</u>	<u>Nomeestabelecimento</u>	<u>Nomepessoa</u>
	1	Benfica	Ana
	2	Coimbra	Beatriz
	3	Benfica	Carlos
	5	Benfica	Eduarda

1 - 4

DRILL-DOWN – No caso da página “Dadores” o utilizador pode ver todos os dadores da aplicação e ao clicar num dos nifs de um dos dadores, vai ser redirecionado para a página “Doações” onde estará informação sobre todas as doações do dador selecionado. Basta seguir os passos:

1. Entrar na página “Dadores” através do Menu Inicial e seleccionar o nif do dador que se pretende consultar.

Início > Inserir DADORES				
	<input type="text"/>	<input type="button" value="Go"/>	<input type="button" value="Actions ▼"/>	<input type="button" value="Create"/>
Search Report				
	<u>Nif</u>	<u>Peso</u>	<u>Sexo</u>	<u>Nomepessoa</u>
	1	50	F	Ana
	2	60	F	Beatriz
	3	70	M	Carlos
	5	90	F	Eduarda

1 - 4


Resultado Final:


[Início](#) > [Inserir DOACOES](#)

Doações feitas

Número de Doações: 1

Selecione NIF 1



	<u>Data doacao</u>	<u>Quantidade recolhida</u>	<u>Nif</u>	<u>Sigla</u>	<u>Nome pessoa</u>
	17-MAY-18	300	1	A+	Ana


1 - 1

Se um utilizador quiser ver as doações feitas por cada dador, basta ir à página das Doações e seleccionar o dador pretendido numa caixa branca que diz “Selecione NIF”:


Doações feitas

Número de Doações: 1

Selecione NIF 3



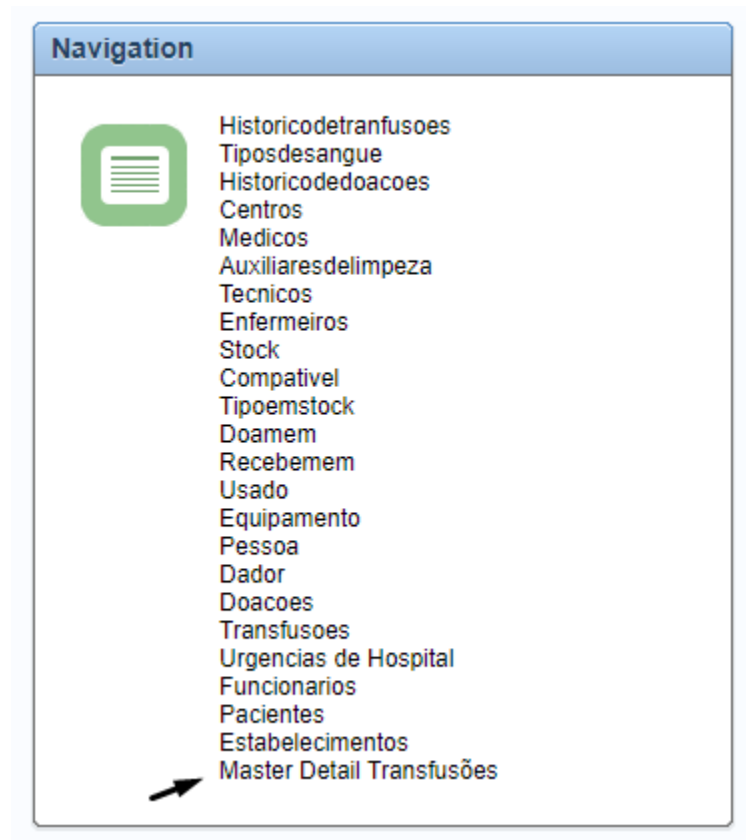
Sem Doacao
1
2
3
5

	<u>Data</u>	<u>Quantidade recolhida</u>	<u>Nif</u>	<u>Sigla</u>	<u>Nome pessoa</u>
	22-MAY-18	400	3	O-	Carlos








1 - 1

Form master-detail – Se o utilizador quiser aceder ao form master-detail (que no caso da nossa aplicação apenas está disponível para Transfusões) precisa de seguir os seguintes passos:

1. Selecionar Master Detail Transfusões no Menu Inicial



2. Clicar no ícone de edição junto do nif da transfusão que se quer seleccionar.

Transfusoes				Create
Edit	Quantidade recolhida	Nif	Sigla	
	300	6	A+	
	340	7	B+	
	360	7	AB+	
	380	7	B+	
	400	8	O+	
	420	8	O+	
	440	9	A-	
				1 - 7

3. Editar o que for preciso e clicar em APPLY CHANGES.

The image shows two overlapping windows from a database application. The top window, titled 'Edit Transfusoes', has a green header bar with buttons for 'Cancel', 'Delete', 'Apply Changes', and a right arrow. The main area shows a record for 'Datatransfusao' dated '17-MAY-18' with 'Nif 6'. It contains two input fields: '*Quantidaderecolhida' with the value '300' and '*Sigla' with the value 'A+'. A '1 of 7' indicator is at the bottom. The bottom window, titled 'Pacientes Detail', also has a green header bar with 'Delete Checked' and 'Add Row' buttons. It displays a table with one row. The first column is 'Sigla Paciente' with a checkbox. The second column contains the value 'A+' with another checkbox. A '1 - 1' indicator is at the bottom.

Edit Transfusoes	
Datatransfusao 17-MAY-18	
Nif 6	
*Quantidaderecolhida	300
*Sigla	A+
1 of 7	

Pacientes Detail	
<input type="checkbox"/> Sigla Paciente	
<input type="checkbox"/> A+	
1 - 1	

Alterações relativamente à 1ª Fase

Foram feitas poucas alterações em relação à 1ª Fase, alguns erros foram corrigidos no diagrama relacional e subsequentemente em todo o que depende deste.

Foi feito o acréscimo de dois “ISA” um para pessoas que engloba funcionários, pacientes e dadores e outro para estabelecimentos que inclui centros e urgências de hospitais.

A mudança mais drástica foi a eliminação da agregação pacientes/transfusões e a adição de uma nova relação com tipos de sangue.

Por fim a entidade Stock deixou de ser uma entidade fraca e ganhou uma nova chave primária nrSérie.

Todas estas mudanças foram devidamente explicadas na parte da Modelação de Dados deste relatório.