



UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO
CURSO DE GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

Victor Augusto Medeiros Balbino

**BANCO DE DADOS: SISTEMA GERENCIADOR DE BANCO
DE SANGUE**

JUAZEIRO - BA
2022

UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO
CURSO DE GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

Victor Augusto Medeiros Balbino

**BANCO DE DADOS: SISTEMA GERENCIADOR DE BANCO
DE SANGUE**

Esta trabalho teve como objetivo o projeto, desenvolvimento de implementação de um banco de dados de Gerenciamento de um Banco de Sangue, assim como realizar comandos e consultas abordados em sala de aula durante o decorrer da disciplina de Banco de Dados II.

Orientador: Prof. Mario Godoy

JUAZEIRO - BA
2022

SUMÁRIO

1	FUNCIONAMENTO DO SISTEMA	4
1.1	DER	4
1.2	DR	5
2	DESENVOLVIMENTO DO PROJETO	6
2.1	CRIAÇÃO DAS TABELAS	6
2.2	COMANDOS E CONSULTAS	8
2.2.1	Insert into	8
2.2.2	Alter table	9
2.2.2.1	ADD	9
2.2.2.2	RENAME	9
2.2.2.3	RENAME COLUMN	9
2.2.3	Funções de conjuntos	9
2.2.3.1	Max/Min	9
2.2.4	AVG	10
2.2.5	COUNT	10
2.2.6	SELECT	10
2.2.7	SELECT - Expressões aritméticas	10
2.2.8	SELECT - ALIASES	10
2.2.9	SELECT - Concatenação de colunas	11
2.2.10	SELECT - Distinct	11
2.2.11	SELECT - NULL	11
2.2.12	SELECT - Order By	11
2.2.13	SELECT - IN	11
2.2.14	SELECT - Between	12
2.2.15	SELECT - Like	12
2.2.16	SELECT - IS NULL	12
2.2.17	JOIN	12
2.2.18	UNION	13
2.2.19	Group by	13
2.2.20	Having	13
2.2.21	Intersect	14
2.2.22	Funções de manipulação	14
2.2.22.1	UPPER/INITCAP/LOWER	14
2.2.22.2	CONCAT	14

2.2.22.3 SUBSTR	14
2.2.22.4 LENGTH	15
2.2.22.5 LPAD	15
2.2.23 UPDATE	15
2.2.24 DELETE	15

1 FUNCIONAMENTO DO SISTEMA

O sistema proposto tem como objetivo simular a operação de um banco de dados de um Banco de Sangue, assim como as suas dependências e processos, como o cadastro do doador, receptor e contato com hospitais.

Podendo assim ser verificado quais tipos sanguíneos estão disponíveis ou ausentes em determinado banco de sangue, além de possibilitar todo o processo de cadastro de um usuário que deseja realizar uma doação. Outra função que pode ser executada é a de verificar quantas vezes a pessoa já doou sangue, além de possibilitar fazer outras consultas relevantes ao contexto ao qual se insere o projeto desenvolvido.

O projeto de SGBD desenvolvido, mesmo sendo um trabalho educacional, contempla todos os recursos e conhecimentos necessários para o desenvolvimento de um projeto maior com finalidades semelhantes, uma vez que fornece ao estudante uma visão geral e ampla do tipo de sistema que pode ser desenvolvido com os conhecimentos adquiridos na disciplina de Banco de Dados II.

1.1 DER

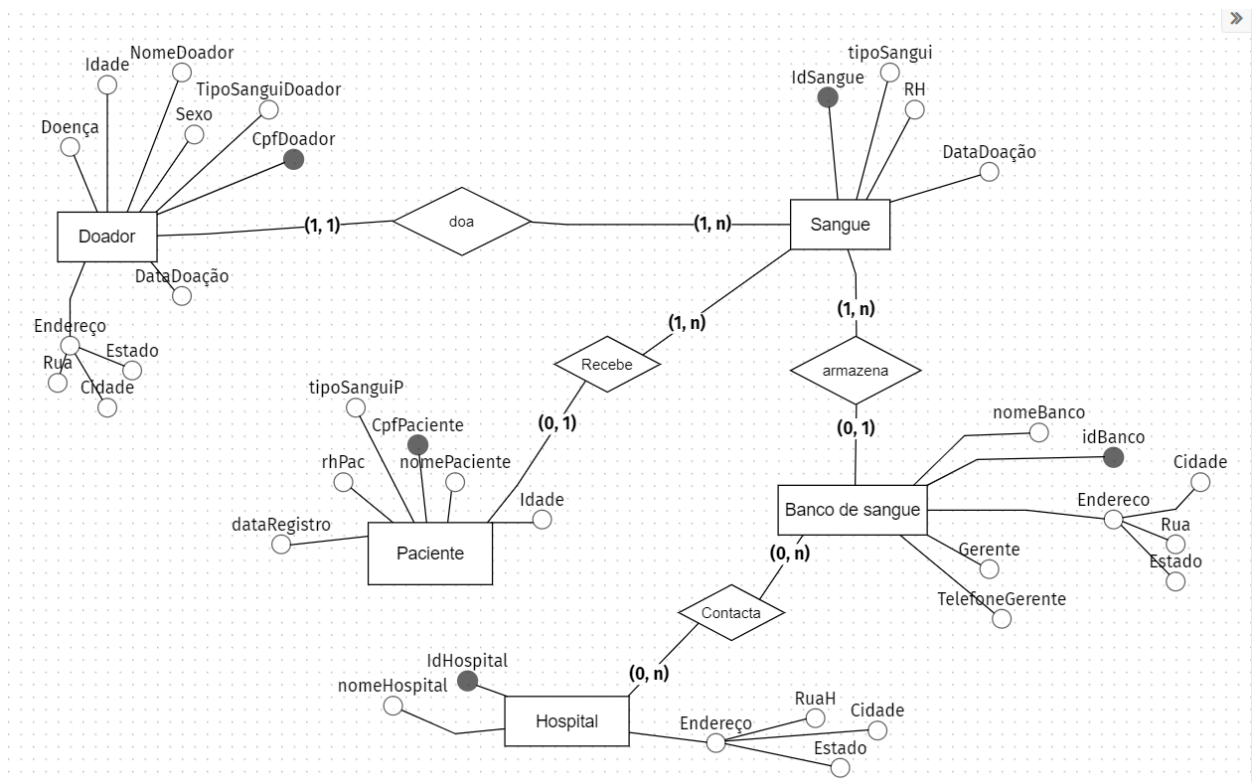


Figura 1 – Diagrama Entidade Relacionamento

1.2 DR

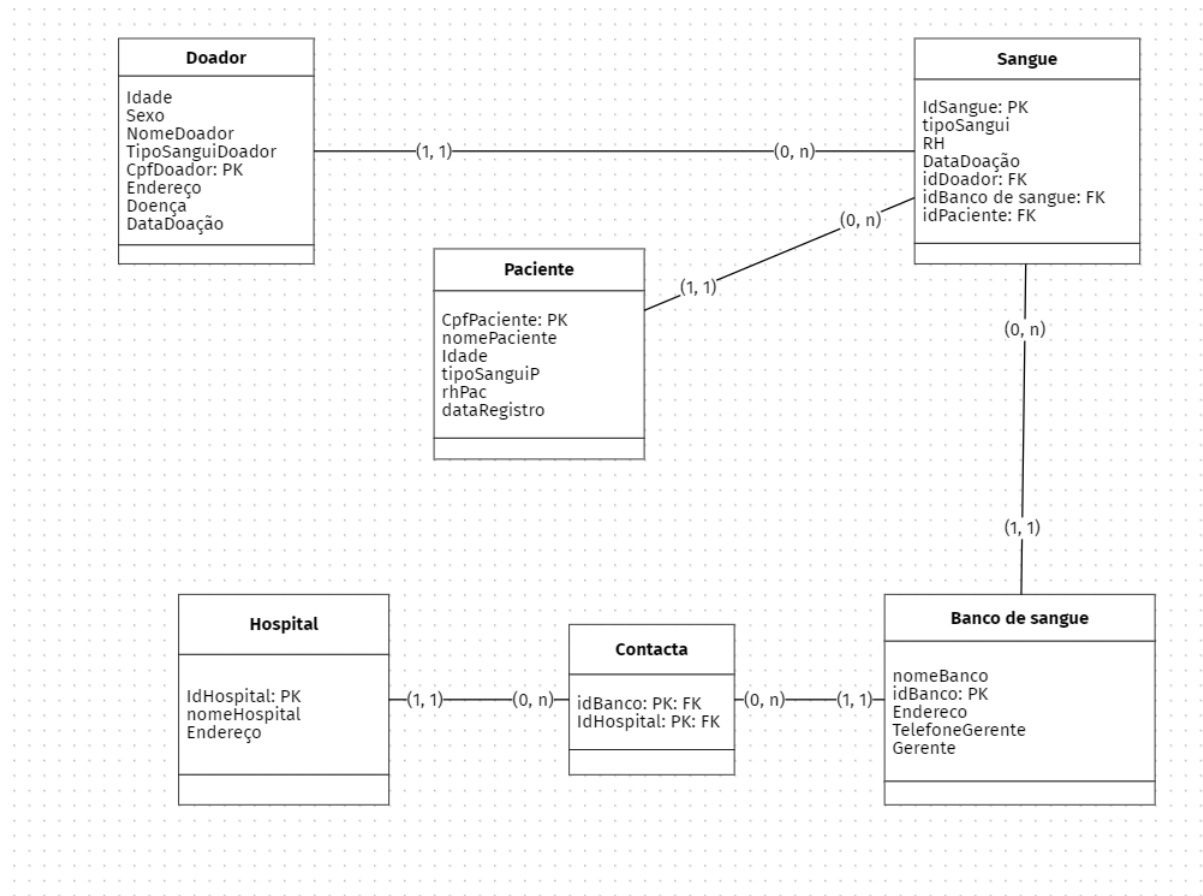


Figura 2 – Diagrama Relacional

2 DESENVOLVIMENTO DO PROJETO

2.1 CRIAÇÃO DAS TABELAS

Código 2.1 – Criação da tabela do Banco de Sangue

```
create table bancoSangue(  
    idBanco serial primary key,  
    nomeBanco varchar(30) not null,  
    rua varchar(40) not null,  
    cidade varchar(20) not null,  
    estado varchar(20) not null,  
    telefone numeric(12) not null,  
    nomeGerente varchar(20) not null,  
    telefoneGerente numeric(12)  
);
```

Código 2.2 – Criação da tabela do Doador

```
create table doador(  
    cpfDoador numeric(11) not null primary key,  
    nomeDoador varchar(50) not null,  
    tipoSanguiDoador varchar(2) not null,  
    rhDoador varchar(2) not null,  
    dataDoacao date not null,  
    sexo varchar(2) not null,  
    idade numeric(2) not null,  
    rua varchar(20) not null,  
    numero numeric(10) not null,  
    cidade varchar(20) not null,  
    estado varchar(20) not null,  
    doenca varchar(30),  
    idBanco serial,  
    constraint check_sexo check (sexo='M' or sexo='F'),  
    constraint checkRH check (rh='+' or rh='-'),  
    constraint check_tipoS check (tipoSanguiDoador='A' or  
        tipoSanguiDoador='B'  
    or tipoSanguiDoador='O' or tipoSanguiDoador='AB'),  
    constraint fk_idBanco foreign key (idBanco)  
    references bancoSangue(idBanco)  
);
```

Código 2.3 – Criação da tabela do Sangue

```
create table sangue(
  idSangue serial primary key,
  tipoSangui varchar(2) not null,
  rh varchar(2) not null,
  idBanco serial not null,
  cpfDoador numeric(11) not null,
  dataDoacao date not null,

  constraint checkRH check (rh='+' or rh='-'),
  constraint check_tipoS check (tipoSangui='A' or tipoSangui='B'
  or tipoSangui='O' or tipoSangui='AB'),

  constraint fk_idBanco foreign key (idBanco)
  references bancoSangue(idBanco),
  constraint fk_cpfDoador foreign key (cpfDoador)
  references doador(cpfDoador)
);
```

Código 2.4 – Criação da tabela do Paciente

```
create table paciente(
  cpfPaciente numeric(11) primary key,
  nomePaciente varchar(100) not null,
  idade numeric(3) not null,
  telefone numeric(11) not null,
  tipoSanguiP varchar(3) not null,
  rhPac varchar(2) not null,
  dataRegistro date not null,
  idSangue serial not null,

  constraint fk_idSangue foreign key (idSangue)
  references sangue(idSangue),

  constraint checkRH check (rhPac='+' or rhPac='-'),
  constraint check_tipoS check (tipoSanguiP='A' or tipoSanguiP='
  B'
  or tipoSanguiP='O' or tipoSanguiP='AB')
);
```


Código 2.5 – Criação da tabela do Hospital

```
create table hospital(
  idHospital serial primary key,
  nomeHospital varchar(50) not null,
  ruaH varchar(50) not null,
  cidade varchar(50) not null,
  estado varchar(10) not null
);
```

Código 2.6 – Criação da tabela entre o Hospital e o Banco

```
create table contactaHospBanco(
  idHospital serial not null,
  idBanco serial not null,
  constraint fk_idHospital foreign key (idHospital)
  references hospital(idHospital),
  constraint fk_idBanco foreign key (idBanco)
  references bancoSangue(idBanco),

  constraint pk_HospBanco primary key(idHospital, idBanco)
);
```

2.2 COMANDOS E CONSULTAS

2.2.1 Insert into

Código 2.7 – Insert into

```
insert into hospital(nomeHospital, ruaH, cidade, estado) values(
  'Hospital Regional de Juazeiro', 'Tv. do Hospital, s/n ', '
  Juazeiro', 'BA');

insert into doador(cpfDoador, nomeDoador, tipoSangui, rhDoador,
  dataDoacao, sexo, idade, rua,
  numero, cidade, estado, idBanco)
values(46764157578, 'Rodrigo Armando Neves', 'A', '-', '
  2022-05-15', 'F', 19,
  'Rua Tiradentes' , 713, 'Juazeiro', 'BA', 1);
```

2.2.2 Alter table

2.2.2.1 ADD

Código 2.8 – ADD

```
Alter table paciente ADD sexo varchar(2) NULL;  
Alter table paciente add doencaPac varchar(20) null;
```

2.2.2.2 RENAME

Código 2.9 – RENAME

```
ALTER TABLE bancoSangue  
  RENAME TO bancodeSangue;  
  
ALTER TABLE paciente  
  RENAME TO pacientes;
```

2.2.2.3 RENAME COLUMN

Código 2.10 – RENAME COLUMN

```
ALTER TABLE doador  
  RENAME COLUMN rh TO rhDoador;  
  
ALTER TABLE sangue  
  RENAME COLUMN rh TO rhSangue;
```

2.2.3 Funções de conjuntos

2.2.3.1 Max/Min

Código 2.11 – Max/Min

```
select max(datadoacao) from sangue;  
  
select min(idade) from doador;
```

2.2.4 AVG

Código 2.12 – Avg

```
select avg(idade) from doador;  
  
select avg(idade) from paciente;
```

2.2.5 COUNT

Código 2.13 – COUNT

```
SELECT COUNT (*) FROM hospital;  
  
SELECT COUNT (*) FROM doador  
WHERE idade > 20;
```

2.2.6 SELECT

Código 2.14 – SELECT

```
select * from bancodeSangue;  
  
select tipoSanguini, rhSanguine, idBanco from sangue;
```

2.2.7 SELECT - Expressões aritméticas

Código 2.15 – SELECT com expressões aritméticas

```
select idade-5 from doador;  
  
select count (*) / 12  
from doador  
where tiposanguine = '0';
```

2.2.8 SELECT - ALIASES

Código 2.16 – SELECT com ALIASES

```
Select telefone as fone, cpfpaciente as Id from pacientes;  
  
Select Idbanco as BancoID from bancodesanguine;
```

2.2.9 SELECT - Concatenação de colunas

Código 2.17 – SELECT com concatenação

```
Select idBanco ||nomebanco from bancodesangue;  
  
Select cpfpaciente ||tiposanguip from pacientes;
```

2.2.10 SELECT - Distinct

Código 2.18 – SELECT com distinct

```
Select DISTINCT cpfDoador from sangue;  
  
Select DISTINCT tipoSanguip from pacientes;
```

2.2.11 SELECT - NULL

Código 2.19 – SELECT - NULL

```
insert into doador  
values( 26515025256, 'Pedro Alves Lima', 'AB', '-', '2022-08-04',  
      'M', 26,  
      'Rua Americo Farias' , 262, 'Juazeiro', 'BA', NULL, 1);  
  
select coalesce(doenca, 'n/a')  
from doador;
```

2.2.12 SELECT - Order By

Código 2.20 – SELECT com Order By

```
Select nomeDoador from doador ORDER BY cpfDoador DESC;  
  
Select idsangue, tipoSanguip from sangue ORDER BY dataDoacao ASC;
```

2.2.13 SELECT - IN

Código 2.21 – SELECT - IN

```
Select nomedoador from doador where nomedoador in ('Luiza Beatriz
Stefany Moura');

Select tipoSangui, rhSangue from sangue where idBanco in (1);
```

2.2.14 SELECT - Between

Código 2.22 – SELECT - BETWEEN

```
Select nomeDoador, tipoSangui, rhDoador from doador where idade
BETWEEN 18 and 30;

Select nomepACIENTE, tipoSanguip from pacientes where
dataRegistro BETWEEN '2022-08-01' and '2022-09-30';
```

2.2.15 SELECT - Like

Código 2.23 – SELECT - Like

```
Select nomeBanco from bancodeSangue where nomeBanco like 'Hemo%';

Select nomeDoador from doador where nomeDoador like 'A%';
```

2.2.16 SELECT - IS NULL

Código 2.24 – SELECT - IS NULL

```
Select nomeDoador, tipoSangui from doador
where doenca IS NULL;

Select nomeGerente from bancodeSangue
where telefoneGerente IS NULL;
```

2.2.17 JOIN

Código 2.25 – JOIN

```
select pacientes.nomePaciente, doador.nomeDoador, sangue.idBanco,
bancodeSangue.nomeBanco
from pacientes, doador, sangue, bancodeSangue
where sangue.cpfDoador = doador.cpfDoador and
```

```

    sangue.idBanco = bancodeSangue.idbanco and
    pacientes.idSangue = sangue.idSangue;

select bancodeSangue.nomeBanco, hospital.nomeHospital
from contactaHospBanco, bancodeSangue, hospital
where bancodeSangue.idBanco = contactaHospBanco.idBanco and
    hospital.idHospital = contactaHospBanco.idHospital;

```

2.2.18 UNION

Código 2.26 – UNION

```

Select nomedoador as id From doador UNION Select nomebanco from
bancodeSangue;

Select cpfDoador as idD From doador UNION Select ALL cpfPaciente
    from
    pacientes;

```

2.2.19 Group by

Código 2.27 – Group By

```

select idBanco from bancodeSangue
group by idBanco;

select estado, count(estado) from bancodeSangue
group by estado;

```

2.2.20 Having

Código 2.28 – Having

```

select tiposanguis, count(idade) from doador
group by tiposanguis
having count(idade) > 20

select estado, sum(idade) from doador
group by estado
having sum (idade) >50

```

2.2.21 Intersect

Código 2.29 – Intersect

```
Select cpfdoador from doador INTERSECT select cpfdoador from
    sangue;
Select idbanco from bancodeSangue where estado= 'BA' INTERSECT
    select
idhospital from hospital;
```

2.2.22 Funções de manipulação

2.2.22.1 UPPER/INITCAP/LOWER

Código 2.30 – UPPER/INITCAP/LOWER

```
Select nomedoador from doador where nomedoador = INITCAP('Luiza
    Beatriz Stefany Moura');

Select UPPER(nomedoador) from doador;

Select LOWER(nomehospital) from hospital;
```

2.2.22.2 CONCAT

Código 2.31 – CONCAT

```
Select CONCAT (nomedoador, cpfdoador) from doador;

Select CONCAT (tipoSanguinP, rhPac) from pacientes;
```

2.2.22.3 SUBSTR

Código 2.32 – SUBSTR

```
Select SUBSTR (nomehospital, 1,5) from hospital;

Select SUBSTR (nomepaciente, 7,10) from pacientes;
```

2.2.22.4 LENGTH

Código 2.33 – LENGTH

```
Select LENGTH (nomehospital) from hospital;  
  
Select LENGTH (nomepaciente) from pacientes;
```

2.2.22.5 LPAD

Código 2.34 – LPAD

```
SELECT LPAD(nomeDoador,15,'*') from doador;  
  
SELECT LPAD(nomeBanco,5,'*') from bancodeSangue;
```

2.2.23 UPDATE

Código 2.35 – UPDATE

```
UPDATE bancodesangue  
SET estado = 'PE'  
WHERE estado = 'Pernambuco';  
  
UPDATE hospital  
SET nomeHospital = 'Hospital SOTE'  
WHERE idHospital = '5';
```

2.2.24 DELETE

Código 2.36 – DELETE

```
DELETE FROM bancodeSangue  
WHERE nomeBanco = 'Hemocentro Regional Palmares';  
  
DELETE FROM doador  
WHERE cpfDoador = 84459192454;
```