

BANCO DE DADOS I

Ciências da Computação

Gabriel Francisco Sahm Gallo RA:151025975

Samuel Cabral RA:181026171

Julio Cesar Benelli Varella RA:181024594

Rafael Nunes Caseiro RA:181024683

Prof. Dr:Aparecido Nilceu Marana

Relatório de Projeto - Andar de internação hospitalar

Bauru
2021

INTRODUÇÃO: O presente relatório tem por objetivo apresentar o “Projeto - Andar de internação hospitalar” , elaborado para a disciplina Banco de Dados I, ministrada pelo Professor Dr. Aparecido Nilceu Marana. O projeto possui como produto final um software de controle para setores de internação e tratamento hospitalar, que foi desenvolvido utilizando-se o banco de dados relacional a partir do PostgreSQL através de sua linguagem procedural SQL própria, plpgsql, sendo a interface gráfica baseada em linguagem Java.

SUMÁRIO

1 Descrição do minimundo	3
2 Modelo de entidade Relacional (MER)	4
3 Modelo Relacional	5
4 Scripts SQL do Oracle	5
5 Guia de instalação e uso do sistema	12
6 Link do GitHub com o código fonte	13

1 Descrição do minimundo

Deseja-se criar um banco de dados para o gerenciamento do setor de internação de um hospital.

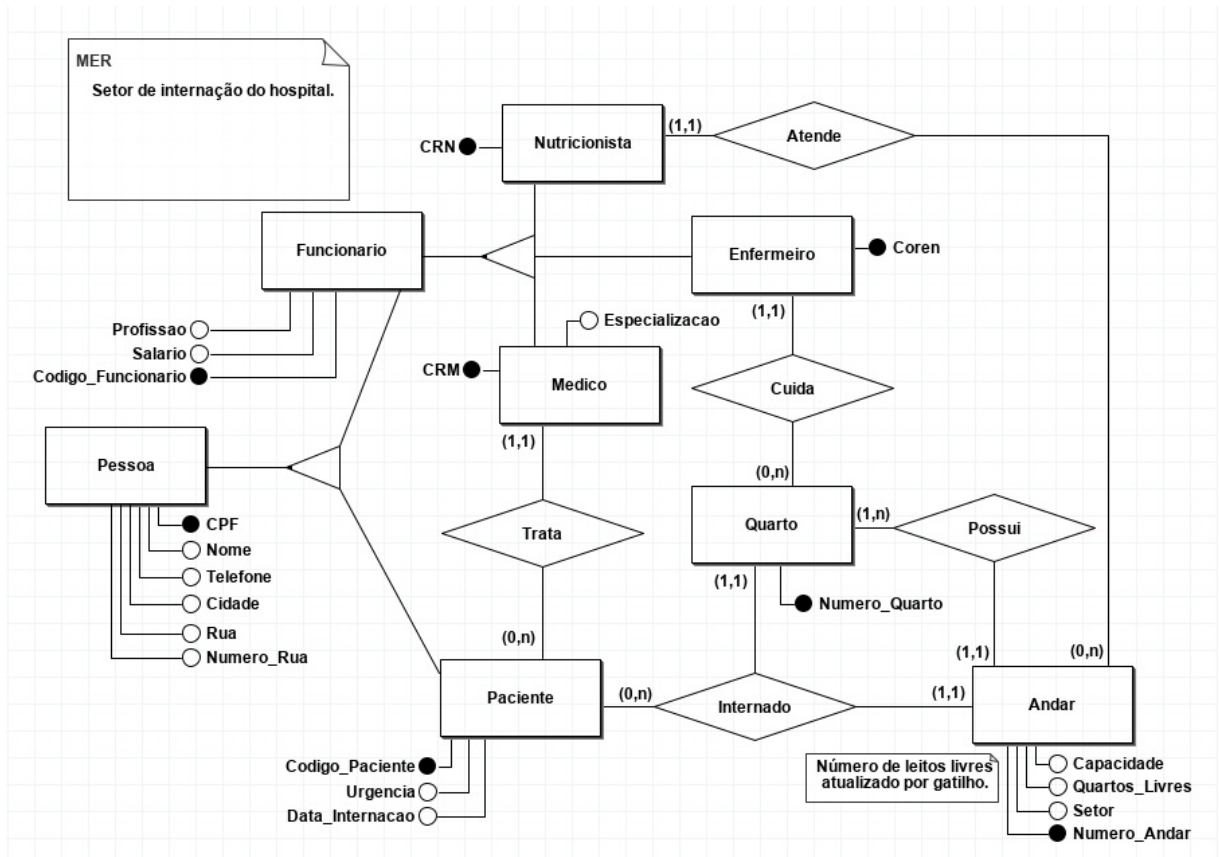
Todas as Pessoas presentes no hospital devem ter seus Nomes, CPFs, Telefones, Cidades e Ruas registrados, elas então serão alocadas entre Funcionários e/ou Pacientes, que irão herdar todas as informações já presentes no cadastro de pessoas.

Para os pacientes será gerado e guardado o código do paciente, CRM do médico responsável, o andar que o paciente se encontra, o número do quarto, a data de internação e a urgência de sua situação hospitalar.

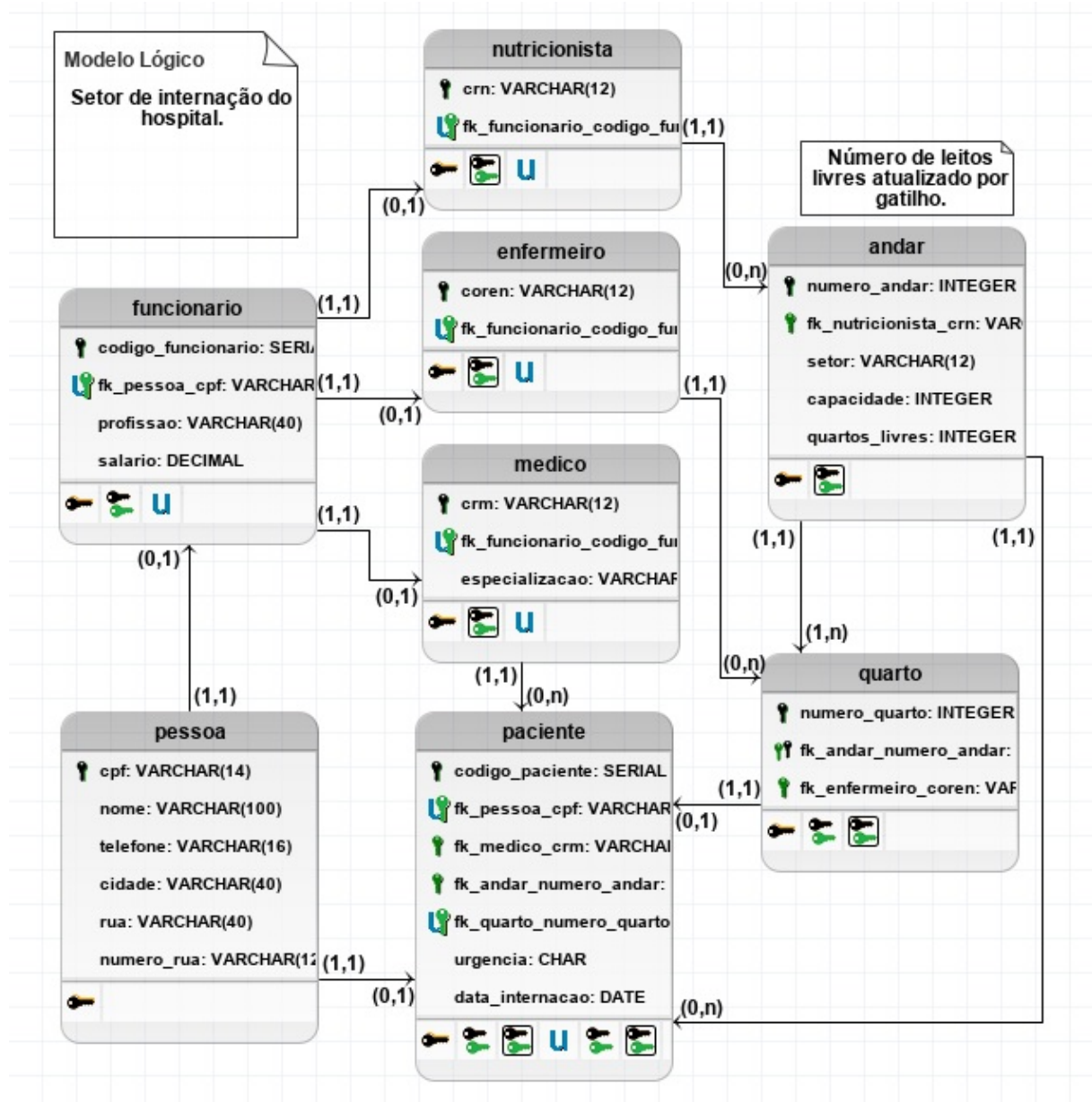
Para os funcionários será gerado e guardado o código de cada funcionário, a profissão que ele exercerá e o salário. Os funcionários ainda serão subdivididos em Nutricionistas, Enfermeiros e Médicos, que deverão herdar todas as informações presentes em funcionários. Nutricionistas ainda terão as informações de seus CRNs e função dentro do seu setor, sendo que os enfermeiros também terão seus CORENs e funções armazenadas, seguidos por Médicos que terão seus CRMs, especializações e funções armazenadas. Um funcionário pode possuir um cadastro como paciente se ele for internado no hospital.

O hospital é dividido em andares, sendo que cada andar terá suas informações como numeração, nutricionista responsável, qual setor pertencem aquele andar, capacidade de pacientes e quantidade de quartos livres, sendo que os quartos terão as suas informações de numeração do quarto, numeração do andar que se encontra o quarto e o enfermeiro responsável.

2 Modelo de entidade Relacional (MER)



3 Modelo Relacional



4 Scripts SQL do Oracle

4.a Criação das tabelas

```
/* Banco de Dados 1 - Projeto Final: */
/* Hospital - Setor de Internação */
/* _____ */
/* Tabelas */
```

```
CREATE TABLE andar (
    numero_andar INTEGER UNIQUE PRIMARY KEY,
    setor VARCHAR(12),
    fk_nutricionista_crn VARCHAR(12),
    capacidade INTEGER,
```

```

leitos_livres INTEGER
);

CREATE TABLE quarto (
    numero_quarto INTEGER PRIMARY KEY,
    fk_andar_numero_andar INTEGER,
    fk_enfermeiro_coren VARCHAR(12)
);

CREATE TABLE pessoa (
    cpf VARCHAR(14) PRIMARY KEY,
    nome VARCHAR(100),
    telefone VARCHAR(16),
    cidade VARCHAR(40),
    rua VARCHAR(40),
    numero_rua VARCHAR(12)
);

CREATE TABLE paciente (
    codigo_paciente SERIAL PRIMARY KEY,
    fk_pessoa_cpf VARCHAR(14) UNIQUE,
    fk_medico_crm VARCHAR(12),
    fk_andar_numero_andar INTEGER,
    fk_quarto_numero_quarto INTEGER UNIQUE,
    urgencia CHAR,
    data_internacao DATE
);

CREATE TABLE funcionario (
    codigo_funcionario SERIAL PRIMARY KEY,
    fk_pessoa_cpf VARCHAR(14) UNIQUE,
    profissao VARCHAR(40),
    salario DECIMAL
);

CREATE TABLE medico (
    crm VARCHAR(12) PRIMARY KEY,
    fk_funcionario_codigo_funcionario INTEGER UNIQUE,
    especializacao VARCHAR(40)
);

CREATE TABLE enfermeiro (
    coren VARCHAR(12) PRIMARY KEY,
    fk_funcionario_codigo_funcionario INTEGER UNIQUE
);

```

```
CREATE TABLE nutricionista (
    crn VARCHAR(12) PRIMARY KEY,
    fk_funcionario_codigo_funcionario INTEGER UNIQUE
);
```

4.b Inserção de tuplas

```
/* Banco de Dados 1 - Projeto Final: */
/* Hospital - Setor de Internação */
/* _____ */
/* Inserts */
```

```
INSERT INTO pessoa(cpf,nome,telefone,cidade,rua,numero_ rua)
VALUES ('111.222.333-44','nomePaciente1',(14) 99876-9181','Bauru','rua
Batista de Carvalho','10-20'),
('111.222.333-55','nomePaciente2',(14) 99876-9182','Bauru','rua Batista
de Carvalho','10-21'),
('111.222.333-66','nomePaciente3',(14) 99876-9183','Bauru','rua Batista
de Carvalho','10-22'),
('111.222.333-77','nomePaciente4',(14) 99876-9184','Bauru','rua Batista
de Carvalho','10-23'),
('111.222.333-88','nomePaciente5',(14) 99876-9185','Bauru','rua Batista
de Carvalho','10-24'),
('222.333.444-55','nomeMedico1',(14) 99876-9271','Bauru','rua Batista de
Carvalho','11-21'),
('222.333.444-56','nomeMedico2',(14) 99876-9272','Bauru','rua Batista de
Carvalho','11-22'),
('333.444.555-66','nomeEnfermeiro1',(14) 99876-9261','Bauru','rua
Batista de Carvalho','12-22'),
('333.444.555-67','nomeEnfermeiro2',(14) 99876-9262','Bauru','rua
Batista de Carvalho','12-23'),
('333.444.555-68','nomeEnfermeiro3',(14) 99876-9263','Bauru','rua
Batista de Carvalho','12-24'),
('444.555.666-77','nomeNutricionista1',(14) 99876-9251','Bauru','rua
Batista de Carvalho','13-23'),
('444.555.666-78','nomeNutricionista2',(14) 99876-9252','Bauru','rua
Batista de Carvalho','13-24'),
('444.555.666-79','nomeNutricionista3',(14) 99876-9253','Bauru','rua
Batista de Carvalho','13-25');
```

```
INSERT INTO funcionario(fk_pessoa_cpf,profissao,salario)
VALUES ('222.333.444-55','Médico', 4000.00),
('222.333.444-56','Médico', 4000.00),
```

```
('333.444.555-66','Enfermeiro', 2000.00),
('333.444.555-67','Enfermeiro', 2000.00),
('333.444.555-68','Enfermeiro', 2000.00),
('444.555.666-77','Nutricionista', 3000.00),
('444.555.666-78','Nutricionista', 3000.00),
('444.555.666-79','Nutricionista', 3000.00);
```

INSERT INTO

```
medico(crm, fk_funcionario_codigo_funcionario, especializacao)
SELECT '12345678', codigo_funcionario, 'urologista'
FROM funcionario WHERE fk_pessoa_cpf = '222.333.444-55';
```

INSERT INTO

```
medico(crm, fk_funcionario_codigo_funcionario, especializacao)
SELECT '12345679', codigo_funcionario, 'urologista'
FROM funcionario WHERE fk_pessoa_cpf = '222.333.444-56';
```

INSERT INTO enfermeiro(coren, fk_funcionario_codigo_funcionario)

```
SELECT '23456789', codigo_funcionario
FROM funcionario WHERE fk_pessoa_cpf = '333.444.555-66';
```

INSERT INTO enfermeiro(coren, fk_funcionario_codigo_funcionario)

```
SELECT '23456790', codigo_funcionario
FROM funcionario WHERE fk_pessoa_cpf = '333.444.555-67';
```

INSERT INTO enfermeiro(coren, fk_funcionario_codigo_funcionario)

```
SELECT '23456791', codigo_funcionario
FROM funcionario WHERE fk_pessoa_cpf = '333.444.555-68';
```

INSERT INTO nutricionista(crm, fk_funcionario_codigo_funcionario)

```
SELECT '34567890', codigo_funcionario
FROM funcionario WHERE fk_pessoa_cpf = '444.555.666-77';
```

INSERT INTO nutricionista(crm, fk_funcionario_codigo_funcionario)

```
SELECT '34567891', codigo_funcionario
FROM funcionario WHERE fk_pessoa_cpf = '444.555.666-78';
```

INSERT INTO nutricionista(crm, fk_funcionario_codigo_funcionario)

```
SELECT '34567892', codigo_funcionario
FROM funcionario WHERE fk_pessoa_cpf = '444.555.666-79';
```

INSERT INTO andar(numero_andar, setor, fk_nutricionista_crm, capacidade)


```
VALUES (4,'Internação','34567890',3),
(5,'Internação','34567891',3);
```

```
INSERT INTO
quarto(numero_quarto,fk_andar_numero_andar,fk_enfermeiro_coren)
VALUES (401,4,'23456789'),
(402,4,'23456789'),
(403,4,'23456790'),
(501,5,'23456790'),
(502,5,'23456791'),
(503,5,'23456791');
```

```
INSERT INTO
paciente(fk_pessoa_cpf,fk_medico_crm,fk_andar_numero_andar,fk_quarto_numero_quarto,urgencia,data_internacao)
VALUES ('111.222.333-44','12345678',4,401,'A','2021-07-31'),
('111.222.333-55','12345679',4,402,'C','2021-08-01'),
('111.222.333-66','12345679',4,403,'C','2021-08-01'),
('111.222.333-77','12345678',5,502,'C','2021-08-01'),
('111.222.333-88','12345678',5,503,'E',CURRENT_DATE);
```

4.c Criação de Restrições

```
/* Banco de Dados 1 - Projeto Final: */
/* Hospital - Setor de Internação */
/* _____ */
/* Restrições */
```

```
ALTER TABLE andar ADD CONSTRAINT FK_andar_1
FOREIGN KEY (fk_nutricionista_crm)
REFERENCES nutricionista (crm)
ON DELETE RESTRICT;
```

```
ALTER TABLE quarto ADD CONSTRAINT FK_quarto_1
FOREIGN KEY (fk_andar_numero_andar)
REFERENCES andar (numero_andar)
ON DELETE RESTRICT;
```

```
ALTER TABLE quarto ADD CONSTRAINT FK_quarto_2
FOREIGN KEY (fk_enfermeiro_coren)
REFERENCES enfermeiro (coren)
```

ON DELETE RESTRICT;

```
ALTER TABLE paciente ADD CONSTRAINT FK_paciente_1
FOREIGN KEY (fk_pessoa_cpf)
REFERENCES pessoa (cpf)
ON DELETE RESTRICT;
```

```
ALTER TABLE paciente ADD CONSTRAINT FK_paciente_2
FOREIGN KEY (fk_andar_numero_andar)
REFERENCES andar (numero_andar)
ON DELETE RESTRICT;
```

```
ALTER TABLE paciente ADD CONSTRAINT FK_paciente_3
FOREIGN KEY (fk_quarto_numero_quarto)
REFERENCES quarto (numero_quarto)
ON DELETE RESTRICT;
```

```
ALTER TABLE paciente ADD CONSTRAINT FK_paciente_4
FOREIGN KEY (fk_medico_crm)
REFERENCES medico (crm)
ON DELETE RESTRICT;
```

```
ALTER TABLE funcionario ADD CONSTRAINT FK_funcionario_1
FOREIGN KEY (fk_pessoa_cpf)
REFERENCES pessoa (cpf)
ON DELETE RESTRICT;
```

```
ALTER TABLE medico ADD CONSTRAINT FK_medico_1
FOREIGN KEY (fk_funcionario_codigo_funcionario)
REFERENCES funcionario (codigo_funcionario)
ON DELETE CASCADE;
```

```
ALTER TABLE enfermeiro ADD CONSTRAINT FK_enfermeiro_1
FOREIGN KEY (fk_funcionario_codigo_funcionario)
REFERENCES funcionario (codigo_funcionario)
ON DELETE CASCADE;
```

```
ALTER TABLE nutricionista ADD CONSTRAINT FK_nutricionista_1
FOREIGN KEY (fk_funcionario_codigo_funcionario)
REFERENCES funcionario (codigo_funcionario)
ON DELETE CASCADE;
```

4.d Criação de Funções e Gatilhos

```

/* Banco de Dados 1 - Projeto Final: */
/* Hospital - Setor de Internação */
/* _____ */
/* Gatilhos e Funções */

--Inicialização de leitos livres após inserção de novo andar
CREATE FUNCTION tr_func_leitos_livres_ini()
RETURNS TRIGGER LANGUAGE 'plpgsql' AS $BODY$
BEGIN
    RAISE NOTICE 'trigger "tr_leitos_livres_ini" ativado';
    UPDATE andar
    SET leitos_livres = capacidade;
    RETURN NULL;
END; $BODY$;

CREATE TRIGGER tr_leitos_livres_ini
AFTER INSERT
ON andar
FOR EACH ROW
EXECUTE PROCEDURE tr_func_leitos_livres_ini();

--Decremento de leitos livres após inserção em paciente
CREATE FUNCTION tr_func_leitos_livres_dec()
RETURNS TRIGGER LANGUAGE 'plpgsql' AS $BODY$
BEGIN
    RAISE NOTICE 'trigger "tr_leitos_livres_dec" ativado';
    UPDATE andar
    SET leitos_livres = leitos_livres - 1
    FROM paciente
    WHERE numero_andar = NEW.fk_andar_numero_andar;
    RETURN NULL;
END; $BODY$;

CREATE TRIGGER tr_leitos_livres_dec
AFTER INSERT
ON paciente
FOR EACH ROW
EXECUTE PROCEDURE tr_func_leitos_livres_dec();

```

```
--Incremento de leitos livres antes de deleção em paciente
CREATE FUNCTION tr_func_leitos_livres_inc()
RETURNS TRIGGER LANGUAGE 'plpgsql' AS $BODY$
BEGIN
    RAISE NOTICE 'trigger "tr_leitos_livres_inc" ativado';
    UPDATE andar
    SET leitos_livres = leitos_livres + 1
    FROM paciente
    WHERE numero_andar = OLD.fk_andar_numero_andar;
    RETURN NULL;
END; $BODY$;

CREATE TRIGGER tr_leitos_livres_inc
AFTER DELETE
ON paciente
FOR EACH ROW
EXECUTE PROCEDURE tr_func_leitos_livres_inc();
```

5 Guia de instalação e uso do sistema

4.a Guia de instalação

Para a utilização do programa são necessários estar instalado na máquina os seguintes softwares:

1-JDK, disponível para download em:

<https://www.oracle.com/java/technologies/javase-jdk16-downloads.html>

2-PostgreSQL, disponível em:

<https://www.postgresql.org/download/>

Durante a instalação do PostgreSQL quando for requisitado o login e senha para o banco de dados utilizar para o login “postgresql” e para a senha “admin”.

Após a instalação do banco de dados do PostgreSQL, é necessário abrir o banco de dados criado por ele, gerar base de dados com o nome “hospital” e fazer uma *query* com o arquivo disponível na pasta do GitHub no caminho BD1-Projeto\Arquivos\SQLs com o nome BD1-Projeto-AllQuerys.sql.

Agora para utilizar o programa, navegar até BD1-Projeto\ProjetoBD\dist e executar o programa ProjetoBD.jar

4.b Uso do Sistema

A utilização do sistema é muito intuitiva, apenas navegar com o mouse para a coluna desejada, Funcionário ou Paciente, selecionar o botão que deseja, dentre das opções, Cadastrar, Consultar e Deletar, preencher com as informações pessoais da pessoa que está sendo cadastrada e clicar no botão salvar ou cancelar, feito isto você será levado novamente ao menu principal para executar uma nova ação.

6 Link do GitHub com o código fonte

Todo o código fonte e aplicativos necessários são encontrados em
<https://github.com/Avatark0/BD1-Projeto>.