



INTEGRANTES DO GRUPO:

- GABRIELLE RODRIGUES
- ROBSON LINS
- VALMIR JUNIOR
- TALITA PACHECO
- ANGELO THIAGO
- BRUNO VICENTE

PRONTOAR

ACREDITE SE QUISER: HOJE EM DIA AINDA UTILIZAM PRONTUÁRIOS FÍSICOS!



HGA Hospital Geral de Areias
Secretaria de Saúde do Estado de Pernambuco
Hospital Geral de Areias Sóny Santos Setor:

NOOME: <i>Pedro Nogueira Leite</i>	IDADE: <i>32</i>
DATA DE ADMISSÃO: <i>22-01-13-10-2021</i>	REGISTRO: <i>04</i>
DATA: <i>14-10-2021</i>	LEITO: <i>04</i>

EVOLUÇÃO DO TÉCNICO EM ENFERMAGEM

SINAIS VITais	Horário	PA	FC	FR	SatO2	T (C°)	HGT	DIURESE (SVD)	DRENAGENS	BALANÇO
DIURNO	12h	<i>100x80</i>	<i>81</i>	<i>19</i>	<i>98</i>	<i>35</i>	<i>141</i>			
	18h									
NOTURNO	24h									
	06h									

DIURNO		NOTURNO	
ESTADO GERAL	REGULAR ()	COMPROMETIDO ()	GRAVE ()
AFFIBIL ()	REGULAR ()	COMPROMETIDO ()	GRAVE ()
CONSCIENTE ()	CONSCIENTE ()	CONSCIENTE ()	CONSCIENTE ()
ORIENTADO ()	DISCENTE ()	DISCENTE ()	CONFUSO ()
SORRIDENTE ()	TOPIFÓRICO ()	TOPIFÓRICO ()	COMATOSO ()
NORMOCRÔNICO ()	HIPOCRÔNICO ()	HIPOCRÔNICO ()	HIPERCRÔNICO ()
ANACTÍRICO ()	ICTÉRICO ()	ICTÉRICO ()	
HIDRATADO ()	DESIDERATADO ()	DESIDERATADO ()	
ANASARCA ()	ANASARCA ()	ANASARCA ()	
INSONIA ()	INSONIA ()	INSONIA ()	
TURGOR E ELAST. DA PELE PRESERVADOS () CONCLUI SONO E REPOSO ()			
RESPIRAÇÃO			
EXPANSÃO TORÁCICA PRESERVADA ()	ARTIFICIAL ()	ARTIFICIAL ()	
TOsse seca ()	EUPNEICO ()	EUPNEICO ()	DISPNEICO ()
TOsse produtiva ()			
TOsse ausente ()			
CARDIACO	NORMOCARDICO ()	TAQUICARDICO ()	BRADICARDICO ()
NORMOTENSO ()	HIPERTENSO ()	HIPERTENSO ()	
PULSO	CHEIO ()	FRACO ()	
RITMO ()	AUSENTE ()	AUSENTE ()	
ACESSO	AVP ()	AVC ()	JUGULAR ()
MSE ()	MSE ()	MSE ()	SUBCLAVIA ()
ABDOMEM	PLANO ()	GLORIOSO ()	DISTENDIDO ()
ESCAVADO ()	DOLOROSO A PALPAÇÃO ()	GLORIOSO ()	
DIETA	Voi ()	DNE ()	SNG ()
COM BOA ACEITAÇÃO ()	COM BOA ACEITAÇÃO ()	COM POUCAS ACEITAÇÃO ()	
EVACUAÇÕES	PRESENTES ()	AUSENTES ()	
DIURESE	PRESENTES ()	AUSENTES ()	
MOBILIDADE	DEAMBULANDO ()	REPOSO RELATIVO ()	REPOSO ABSOLUTO ()
PASSIVO ()	ATIVO ()	ATIVO ()	

DRA. JÚLIA S. CORRÊA
Dra. Julia Souza Corrêa CRP 2025
99638-6452 | drjuliacorrêa.com.br
Qd. dos Chêries, 40 | Fluminense | Rio de Janeiro | RJ | CEP: 22710-000

Prontuário Médico

Paciente	Data de nascimento	CPF
Nome	Sexo	
Endereço	CEP	UF

Assinatura / Histórico clínico progressivo

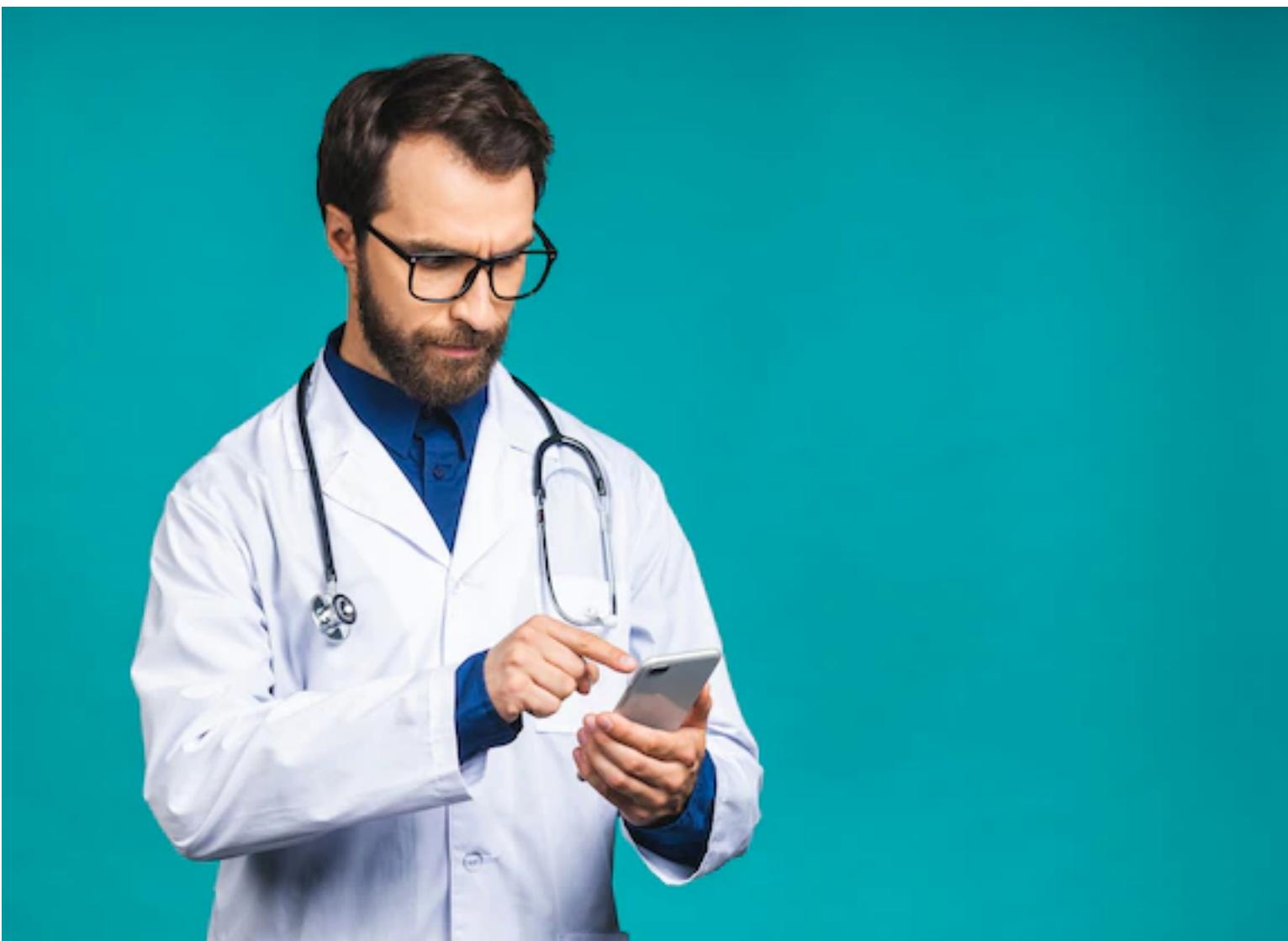
Doenças presentes	Doenças ausentes
Tosse seca	Dispepsia
Doenças crônicas	Doenças agudas
Alergias	Diabetes
Ingestão de álcool	Obesidade
Uso de antibióticos	Dificuldade de cicatrização
Uso de medicamentos	Exa. grávida
Presença arterial	Outras



Nome do Psicólogo
CRPXX/XXXX-X

PRONTOAR

UMA APLICAÇÃO WEB, PARA GARANTIR UM
MELHOR GERENCIAMENTO, DA SUA SAÚDE!

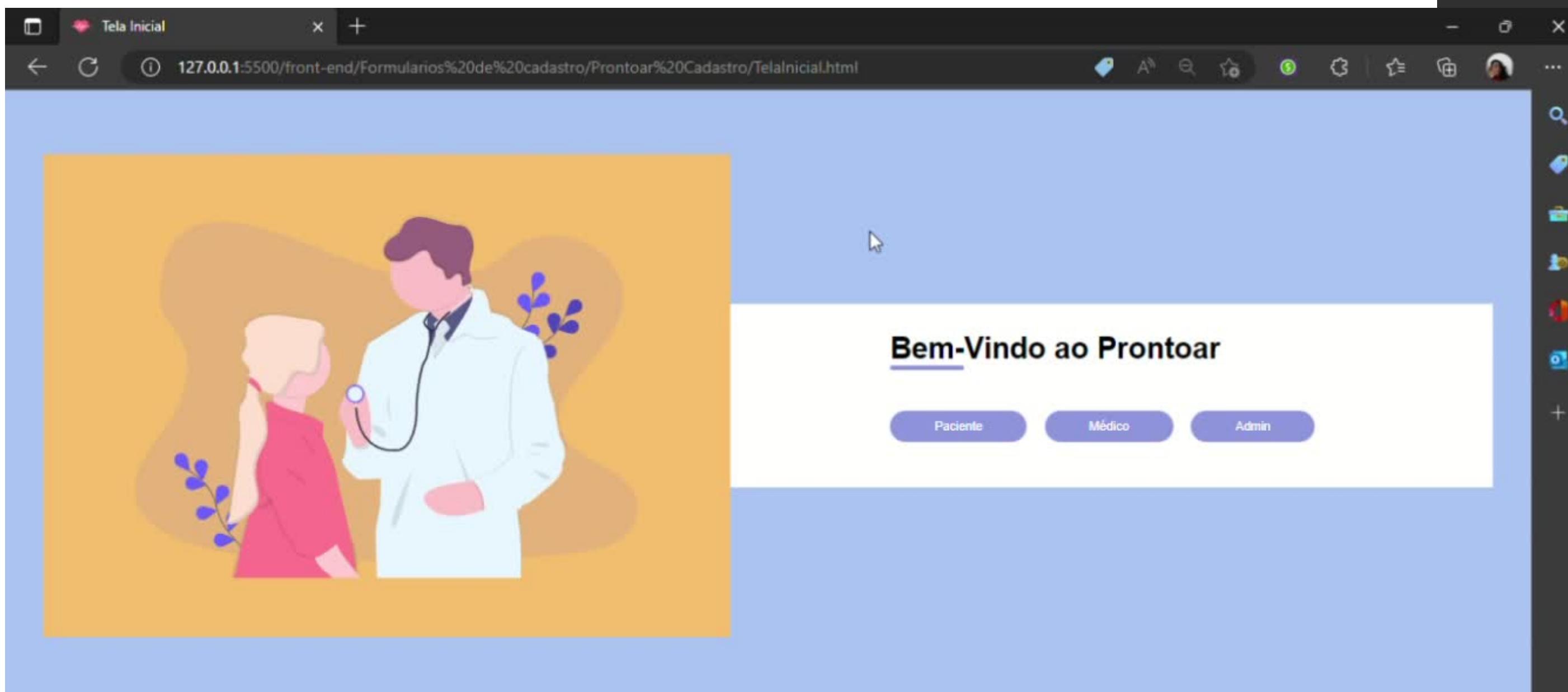


Casos de uso

- **Caso 1:** Como administrador da unidade de saúde, quero ter mais organização e agilidade, para acessar dados do paciente.
- **Caso 2:** Como médico desejo portabilidade para diagnosticar os pacientes.



Tela inicial



Cadastro Paciente



A screenshot of a web browser showing a patient registration form. The title bar reads "Paciente" and the address bar shows the URL "127.0.0.1:5500/front-end/Formularios%20de%20cadastro/Prontoar%20Cadastro/cadastroPaciente.html". The main content area is titled "Cadastro paciente". The form fields are as follows:

Primeiro Nome Gabrielle	Sobrenome Rodrigues	Nome Social(caso tenha) Gabrielle Rodrigues	
Celular (81) 98969-898	CPF do paciente 988.867.778-98	Confirme CPF 988.867.778-98	
Idade 21	Data de Nascimento 05/08/2001	E-mail gabriellerodriguesmd@gmail.com	
Altura 1.68	Peso 67	Endereço Avenida Boa Viagem	
Gêneros	<input type="radio"/> Feminino <input type="radio"/> Masculino <input type="radio"/> Outros <input type="radio"/> Prefiro não informar		

salvar

Login

The screenshot shows a web browser window titled "Tela de Login Médico". The URL in the address bar is "127.0.0.1:5500/front-end/Formularios%20de%20cadastro/Prontoar%20Cadastro/TelaLoginMedico.html". The page features a background illustration of a doctor in a white coat and a patient in a pink dress, both holding blue leafy branches. To the right, the word "Prontoar" is displayed in bold black text. Below it, there are two input fields: "Login" containing "josuecarlos@gmail.com" and "Senha" containing "Digite sua senha de acesso". At the bottom are two buttons: "Voltar" (Back) and "Acessar" (Access).

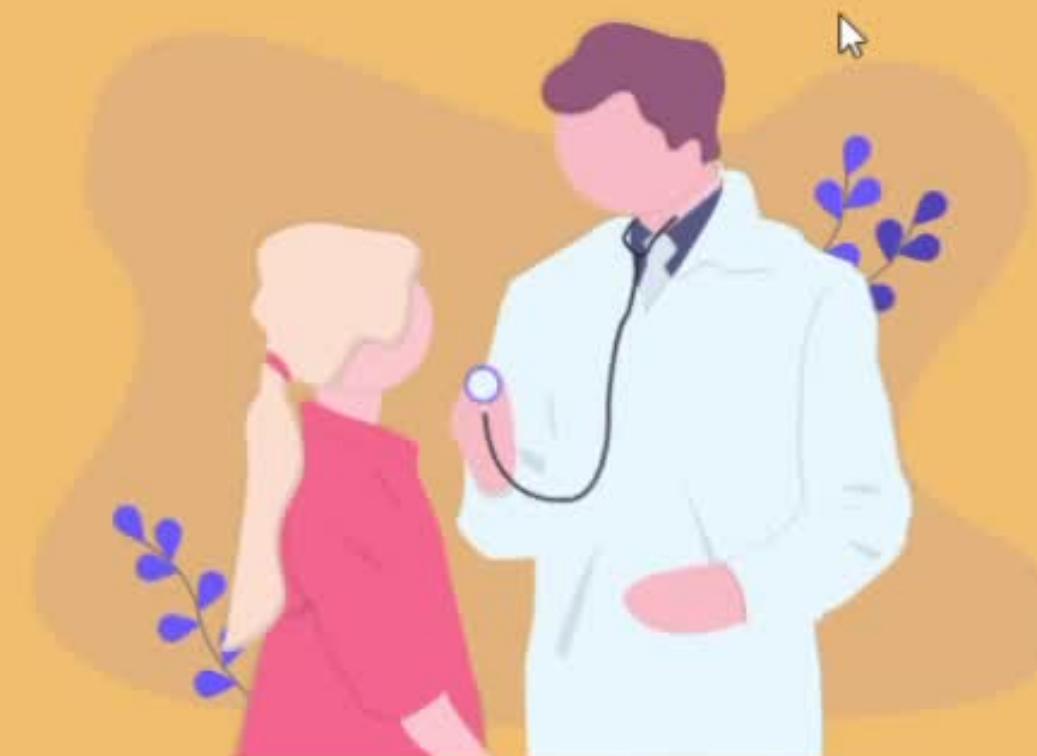
Atualizar Prontuário

Tela de diagnóstico

127.0.0.1:5500/front-end/Formularios%20de%20cadastro/Prontoar%20Cadastro/TelaProntuario.html?paciente=1

Diagnóstico do paciente

Paciente: Ana Maria



Pressão Arterial	Queixas	Data Consulta	Temperatura
120	Sentindo falta de ar.	20/11/2022	45

Prescrições Medicas	Ocorrências	Doenças presentes
Tomar 1 Dipirona a cada 4 h	Até o momento nada.	Covid-19

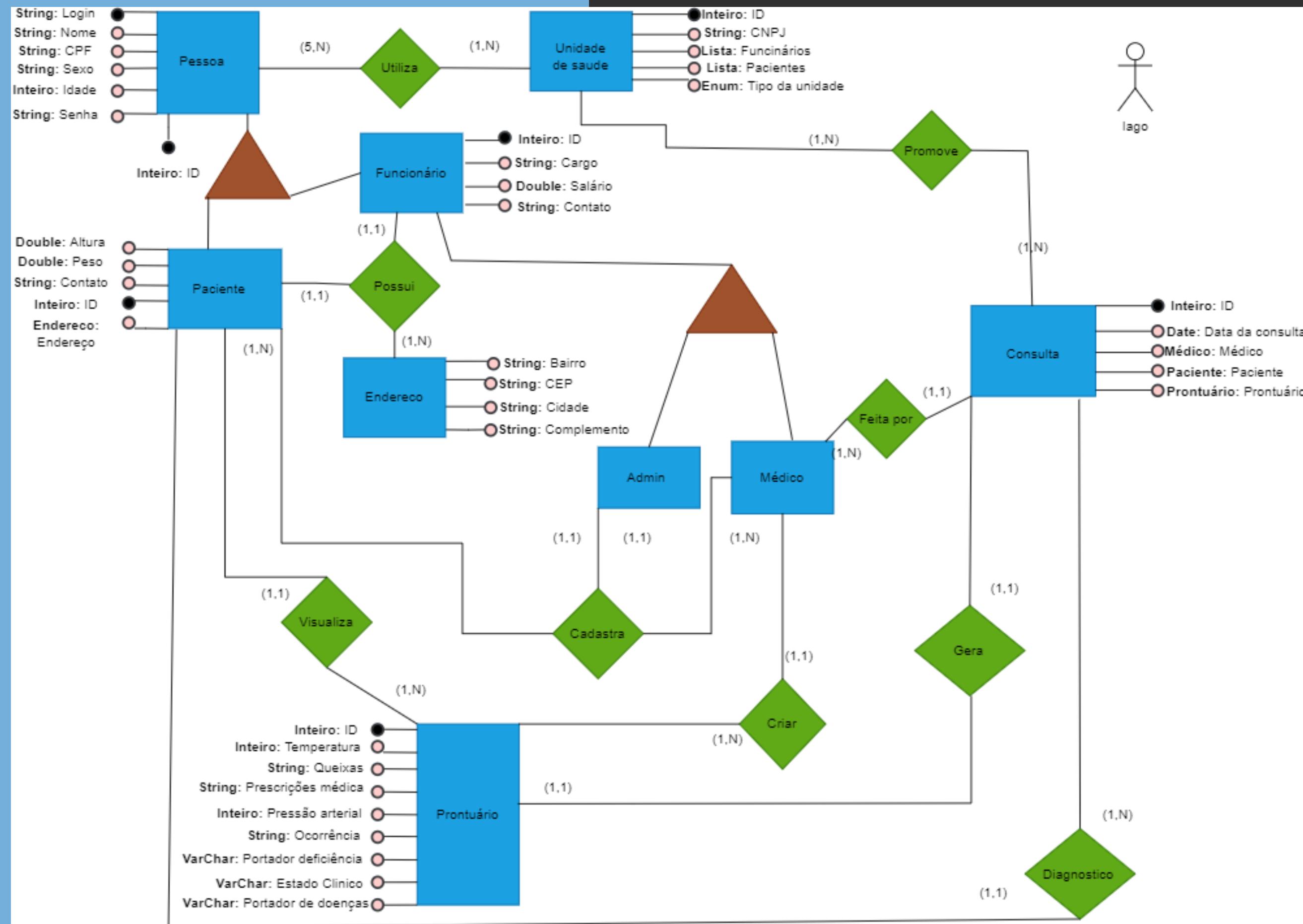
Estado geral
 Regular Comprometido Grave

Portador de doenças transmissíveis
 Sim Não Não sei informar

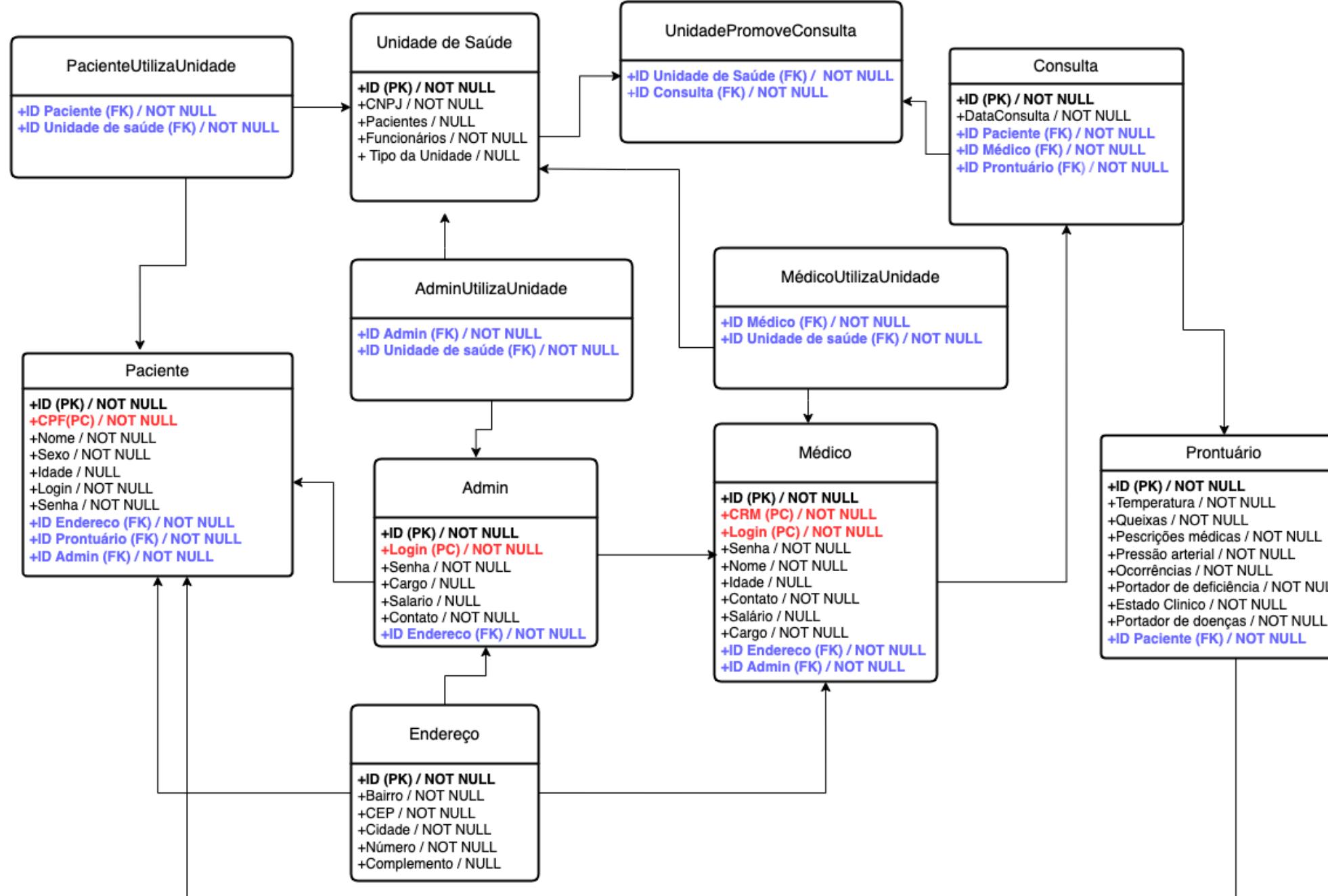
Portador de deficiência
 Sim Não Talvez

Voltar

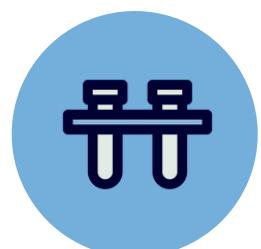
Modelagem ER



Modelagem Relacional



Principais Tabelas



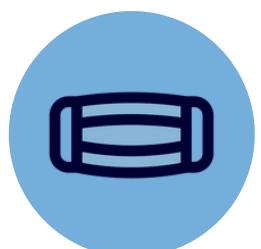
- medico



- prontuario



- administrador



- paciente



- unidade_saude



- consulta



Comandos SQL



Principais tabelas

-- Deletando o banco de dados para criar novamente

```
DROP DATABASE IF EXISTS prontoar;  
CREATE DATABASE prontoar;  
USE prontoar;
```

-- Criação das tabelas

-- medico

```
CREATE TABLE medico(  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    crm VARCHAR(8) NOT NULL,  
    login VARCHAR(50) NOT NULL,  
    senha VARCHAR(50) NOT NULL,  
    nome VARCHAR(255) NOT NULL,  
    idade INT(3) NOT NULL,  
    contato VARCHAR(15) NOT NULL,  
    cargo VARCHAR(50) NOT NULL,  
    salario DOUBLE NOT NULL,  
    id_endereco INT(11) NOT NULL,  
    CONSTRAINT fk_id_endereco3 FOREIGN KEY (id_endereco) REFERENCES  
        endereco(id),  
    id_administrador INT(11) NOT NULL,  
    CONSTRAINT fk_id_administrador2 FOREIGN KEY (id_administrador)  
        REFERENCES administrador(id)  
);
```



Comandos SQL



Principais tabelas

-- prontuario

```
CREATE TABLE prontuario (
    id INT AUTO_INCREMENT PRIMARY KEY,
    temperatura FLOAT(10) NOT NULL,
    queixas TEXT NOT NULL,
    prescicoes TEXT NOT NULL,
    pressao_arterial FLOAT(10) NOT NULL,
    ocorrencias TEXT NOT NULL,
    portador_deficiencia ENUM('sim','nao'),
    estado_clinico ENUM('grave', 'estavel', 'normal'),
    portador_doenca_transmissivel ENUM('sim','nao'),
    id_paciente INT(11) NOT NULL,
    CONSTRAINT fk_paciente FOREIGN KEY (id_paciente) REFERENCES paciente(id)
);
```

-- consulta

```
CREATE TABLE consulta (
    id INT AUTO_INCREMENT PRIMARY KEY,
    data_consulta DATETIME NOT NULL,
    id_paciente INT(11) NOT NULL,
    CONSTRAINT fk_id_paciente FOREIGN KEY (id_paciente) REFERENCES paciente(id),
    id_medico INT(11) NOT NULL,
    CONSTRAINT fk_id_medico FOREIGN KEY (id_medico) REFERENCES medico(id),
    id_prontuario INT(11) NOT NULL,
    CONSTRAINT fk_id_prontuario_2 FOREIGN KEY (id_prontuario)
    REFERENCES prontuario(id)
);
```



Comandos SQL



Principais tabelas

```
-- unidade_saude
CREATE TABLE unidade_saude (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cnpj VARCHAR(14) NOT NULL,
    tipo_unidade ENUM('clinica', 'hospital', 'consultorio') NOT NULL
);

-- endereco
CREATE TABLE endereco (
    id INT AUTO_INCREMENT PRIMARY KEY,
    bairro VARCHAR(50) NOT NULL,
    cep VARCHAR(10) NOT NULL,
    cidade VARCHAR(50) NOT NULL,
    numero VARCHAR(6) NOT NULL,
    complemento VARCHAR(100)
);
```



Comandos SQL



Principais tabelas

-- Administrador

```
CREATE TABLE administrador(
    id INT AUTO_INCREMENT PRIMARY KEY,
    login VARCHAR(50) NOT NULL,
    senha VARCHAR(50) NOT NULL,
    cargo VARCHAR(50) NOT NULL,
    salario DECIMAL(5,3) NOT NULL,
    contato VARCHAR(15) NOT NULL,
    id_endereco INT(11) NOT NULL,
    CONSTRAINT fk_id_endereco FOREIGN KEY (id_endereco) REFERENCES
    endereço(id)
);
```

-- Paciente

```
CREATE TABLE paciente (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    idade INT(3) NOT NULL,
    sexo ENUM('M', 'F', 'Outro') NOT NULL,
    login VARCHAR(50) NOT NULL,
    senha VARCHAR(50) NOT NULL,
    cpf VARCHAR(15) NOT NULL UNIQUE,
    id_endereco INT(11) NOT NULL,
    CONSTRAINT fk_id_endereco_2 FOREIGN KEY (id_endereco) REFERENCES
    endereço(id),
    id_administrador INT(11) NOT NULL,
    CONSTRAINT fk_id_administrador FOREIGN KEY (id_administrador)
    REFERENCES administrador(id)
);
```



Tabelas de Relacionamento



- **unidade_promove_consulta**



- **admin_utiliza_unidade**



- **paciente_utiliza_unidade**



- **medico_utiliza_unidade**



Comandos SQL



Tabelas de relacionamento

```
-- unidade_promove_consulta
CREATE TABLE unidade_promove_consulta(
    id_unidade_saude INT(11) NOT NULL,
    CONSTRAINT fk_id_unidade_saude FOREIGN KEY (id_unidade_saude)
    REFERENCES unidade_saude(id),
    id_consulta INT(11) NOT NULL,
    CONSTRAINT fk_id_consulta FOREIGN KEY (id_consulta)
    REFERENCES consulta(id)
);

-- admin_utiliza_unidade
CREATE TABLE admin_utiliza_unidade (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_administrador INT(11) NOT NULL,
    CONSTRAINT fk_id_administrador_3 FOREIGN KEY
    (id_administrador) REFERENCES administrador(id),
    id_unidade INT(11) NOT NULL,
    CONSTRAINT fk_id_unidade_2 FOREIGN KEY
    (id_unidade) REFERENCES unidade_saude(id)
);
```



Comandos SQL



Tabelas de relacionamento

```
-- paciente_utiliza_unidade
CREATE TABLE paciente_utiliza_unidade(
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_paciente INT(11) NOT NULL,
    CONSTRAINT fk_id_paciente_2 FOREIGN KEY
        (id_paciente) REFERENCES paciente(id),
    id_unidade INT(11) NOT NULL,
    CONSTRAINT fk_id_unidade_3 FOREIGN KEY
        (id_unidade) REFERENCES unidade_saude(id)
);
```

```
-- medico_utiliza_unidade
CREATE TABLE medico_utiliza_unidade(
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_medico INT(11) NOT NULL,
    CONSTRAINT fk_id_medico_2 FOREIGN KEY (id_medico) REFERENCES
        medico(id),
    id_unidade INT(11) NOT NULL,
    CONSTRAINT fk_unidade_saude_2 FOREIGN KEY (id_unidade)
        REFERENCES unidade_saude(id)
);
```



Consultas complexas



- Fazer consulta de qual admin cadastrou qual paciente (em casos de dados inconsistentes cadastrados).



- Listar todos os pacientes que são portadores de doença transmissível.



- Obter os pacientes que possuem deficiência e que estão com a pressão arterial maior que 12



- Selecionar o prontuário completo junto com o nome do paciente.



- Listar todas as consultas agendadas para um determinado paciente



- Encontrar a quantidade de consultas realizadas por cada médico.



Comandos SQL



Consultas Complexas

-- Fazer consulta de qual admin cadastrou qual paciente (em casos de dados inconsistentes cadastrados):

```
SELECT t1.login AS login_administrador, t2.* FROM administrador AS t1 JOIN paciente  
AS t2 ON t1.id = t2.id_administrador;
```

-- Selecionar o prontuário completo junto com o nome do paciente e nome do médico.

```
SELECT t1.nome AS nome_paciente, t2.* FROM paciente AS t1 JOIN prontuario AS t2  
ON t1.id = t2.id_paciente;
```

-- Listar todos os pacientes que são portadores de doença transmissível.

```
SELECT p.nome, p.cpf FROM paciente p INNER JOIN prontuario pt ON p.id =  
pt.id_paciente WHERE pt.portador_doenca_transmissivel = 'sim';
```



Comandos SQL



Consultas Complexas

--Obter os pacientes que possui deficiência e que estão com a pressão arterial maior que 12.

```
SELECT t1.* FROM prontuario AS t1 JOIN consulta AS t2 ON t2.id_paciente =  
t1.id_paciente JOIN medico AS t3 ON t3.id = t2.id_medico WHERE  
portador_deficiencia = 'sim' AND pressao_arterial > 12 ORDER BY t2.data_consulta  
DESC;
```

--Encontrar a quantidade de consultas realizadas por cada médico.

```
SELECT m.id, m.nome, COUNT(c.id) AS total_consultas FROM medico m LEFT JOIN  
consulta c ON m.id = c.id_medico GROUP BY m.id, m.nome;
```

--Listar todas as consultas agendada para um determinado paciente.

```
SELECT c.id, c.data_consulta, m.nome AS nome_medico FROM consulta c  
INNER JOIN medico m ON c.id_medico = m.id WHERE c.id_paciente = (SELECT id  
FROM paciente WHERE nome = 'Vanessa');
```



Triggers



- Ao cadastrar administrador, vinculá-lo com uma unidade de saúde



- Quando um novo registro é inserido na tabela prontuário e atualiza automaticamente o estado clínico do paciente na tabela paciente com base no estado clínico informado no prontuário



- Quando houver uma tentativa de alteração de consulta do novo paciente e deseja remanejar para uma outra data ou cancelar consulta



- Não permitir que o CRM de um médico seja alterado



Comandos SQL



Triggers

-- Esta Trigger é acionada para quando cadastrar administrador, vinculá-lo com uma unidade de saúde.

DELIMITER //

```
CREATE TRIGGER trg_adiciona_administrador
AFTER INSERT ON administrador
FOR EACH ROW
BEGIN
    DECLARE unidade_id INT;

    -- Recupera o ID da unidade a partir da tabela de unidades
    SELECT id INTO unidade_id FROM unidade_saude ORDER BY id DESC LIMIT 1;

    -- Insere o registro na tabela admin_utiliza_unidade
    INSERT INTO admin_utiliza_unidade (id_administrador, id_unidade)
    VALUES (NEW.id, unidade_id);
END //
```

DELIMITER :

```
SELECT * FROM admin_utiliza_unidade;

INSERT INTO administrador
( login, senha, cargo, salario, contato, id_endereco)
VALUES ('alan01@prontoar', 'alan2568', 'receptionista', 2.568, 8111111111, 1);
```



Comandos SQL



Triggers

-- Este trigger é acionado sempre que um novo registro é inserido na tabela prontuário e atualiza automaticamente o estado clínico do paciente na tabela paciente com base no estado clínico informado no prontuário.

```
DROP TRIGGER update_estado_clinico;

DELIMITER //
CREATE TRIGGER update_estado_clinico
AFTER INSERT ON prontuario
FOR EACH ROW
BEGIN
    UPDATE paciente
    SET estado_clinico = NEW.estado_clinico
    WHERE id = NEW.id_paciente;
END //
DELIMITER ;

SELECT * FROM prontuario;
```



Comandos SQL



Triggers

-- Esta Trigger é acionada quando houver um tentativa de alteração de consulta do novo paciente e deseja remanejar para uma outra data ou cancelar consulta.

```
DELIMITER //
CREATE TRIGGER alter_data_consulta
AFTER INSERT ON consulta
FOR EACH ROW
BEGIN
    ALTER consulta
    SET data_consulta = NEW.data_consulta WHERE id = NEW.id_consulta;
    OR DELETE FROM consulta WHERE id = 1;
END //
DELIMITER;
SELECT * FROM consulta;
```



Comandos SQL



Triggers

-- Esta Trigger não permite que o CRM de um médico seja alterado

```
DELIMITER //
CREATE TRIGGER medico_update
BEFORE UPDATE ON medico
FOR EACH ROW
BEGIN
    IF NEW.crm <> OLD.crm THEN
        SIGNAL SQLSTATE = '45000'
        SET MESSAGE_TEXT = 'A alteração do valor do CRM não é permitida.';
    END IF;
END //
DELIMITER;
UPDATE medico SET CRM = '12345' WHERE id = 1;
```





The End!!!
