

CENTRO UNIVERSITARIO DO DISTRITO FEDERAL – UDF

DISCIPLINA

Banco de Dados

BRASÍLIA

2024

Alunos

DÉLIO SANTANA DE SOUSA MARQUES - 32939680

ADRIAN COLEGARIO BATISTA – 33033862

GABRIEL DO CARMO CALDAS DE OLIVEIRA ANDRADE - 34620249

## 1. CONTEXTUALIZAÇÃO DO PROBLEMA

A Clínica Médica "Saúde em Foco" é uma instituição de saúde renomada na cidade, que oferece uma ampla gama de serviços médicos para seus pacientes. A clínica atende a uma grande variedade de necessidades médicas, desde consultas gerais até procedimentos especializados e cirurgias.

Com o objetivo de melhorar sua eficiência operacional e garantir um atendimento de alta qualidade, a clínica decidiu informatizar seus processos e implementar um sistema de gerenciamento completo. Esse sistema precisa abranger todas as áreas funcionais da clínica, desde o agendamento de consultas até o acompanhamento pós-tratamento dos pacientes.

Problema:

A Clínica Médica "Saúde em Foco" está enfrentando desafios relacionados à gestão eficaz de seus recursos e ao acompanhamento detalhado dos pacientes. Os problemas identificados incluem:

**Agendamento de Consultas:** O sistema atual de agendamento de consultas é manual e propenso a erros. Os pacientes muitas vezes enfrentam longas esperas e dificuldades para marcar consultas com os médicos desejados.

**Registro de Pacientes:** Não há um sistema centralizado para o registro e histórico médico dos pacientes. Isso dificulta o acesso rápido às informações do paciente durante as consultas e pode levar a diagnósticos imprecisos ou atrasados.

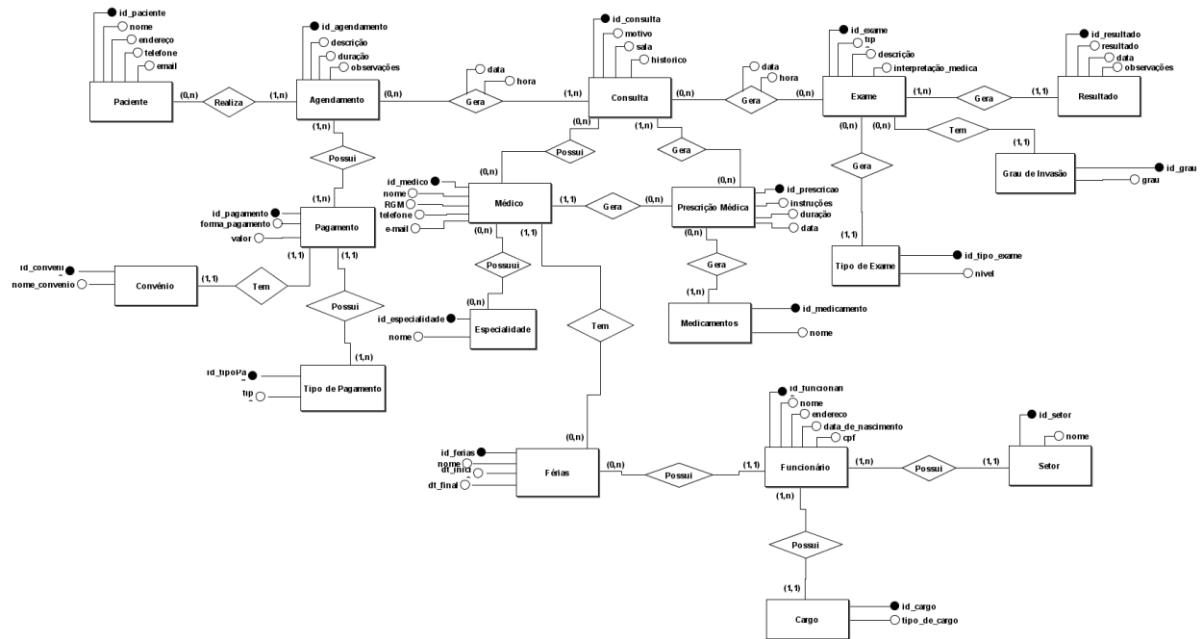
**Gestão de Recursos:** A clínica enfrenta dificuldades na alocação eficiente de recursos, como salas de consulta, equipamentos médicos e funcionários. Isso resulta em subutilização de recursos ou sobrecarga em determinados horários.

**Coordenação entre Departamentos:** A comunicação entre os diferentes departamentos da clínica, como recepção, enfermagem, laboratório e farmácia, é fragmentada e desorganizada. Isso leva a atrasos na execução de exames, coleta de amostras e dispensação de medicamentos.

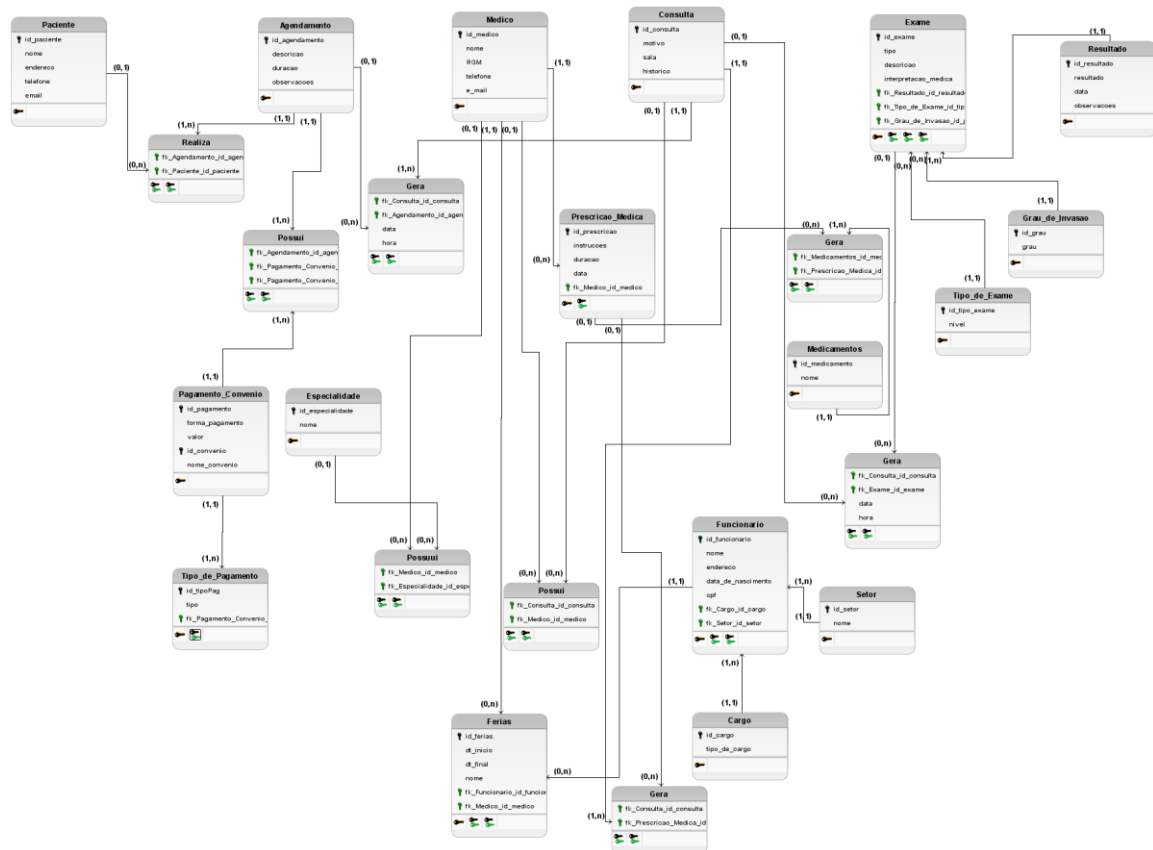
**Segurança dos Dados:** Os registros médicos dos pacientes não são armazenados de forma segura e não há medidas adequadas de segurança de dados para proteger as informações confidenciais dos pacientes.

Para resolver esses problemas e melhorar a eficiência geral da clínica, é necessário desenvolver um sistema de gerenciamento abrangente que integre todos os aspectos do funcionamento da clínica, desde o agendamento de consultas até o acompanhamento pós-tratamento dos pacientes. Isso exigirá a criação de um modelo entidade-relacionamento robusto que represente com precisão todas as entidades e seus relacionamentos na clínica.

## 2. MODELO ENTIDADE RELACIONAMENTO (MER)



## 3. MODELO RELACIONAL



#### 4. MODELO FÍSICO

```
CREATE TABLE Paciente (
    id_paciente INTEGER PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR (255) NOT NULL,
    endereco VARCHAR (255),
    telefone CHAR (14),
    email VARCHAR (255) UNIQUE
);
```

```
CREATE TABLE Agendamento (
    id_agendamento INTEGER PRIMARY KEY AUTO_INCREMENT,
    descricao VARCHAR (255) NOT NULL,
    duracao TIME,
```

```
observacoes VARCHAR (500)
);
```

```
CREATE TABLE Consulta (
    id_consulta INTEGER PRIMARY KEY AUTO_INCREMENT,
    motivo VARCHAR (255) NOT NULL,
    sala INTEGER,
    historico VARCHAR (500)
);
```

```
CREATE TABLE Pagamento_Convenio (
    id_pagamento INTEGER PRIMARY KEY AUTO_INCREMENT,
    forma_pagamento VARCHAR (50) NOT NULL,
    valor DECIMAL (10, 2) NOT NULL,
    id_convenio INTEGER NOT NULL,
    nome_convenio VARCHAR (255),
    PRIMARY KEY (id_pagamento, id_convenio)
);
```

```
CREATE TABLE Tipo_de_Pagamento (
    id_tipoPag INTEGER PRIMARY KEY AUTO_INCREMENT,
    tipo VARCHAR (50),
    fk_Pagamento_Convenio_id_pagamento INTEGER
);
```

```
CREATE TABLE Exame (
    id_exame INTEGER PRIMARY KEY AUTO_INCREMENT,
    tipo VARCHAR (100) NOT NULL,
```

```
descricao VARCHAR (500),  
interpretacao_medica VARCHAR (500),  
fk_Resultado_id_resultado INTEGER,  
fk_Tipo_de_Exame_id_tipo_exame INTEGER,  
fk_Grau_de_Invasao_id_grau INTEGER  
);
```

```
CREATE TABLE Tipo_de_Exame (  
    id_tipo_exame INTEGER PRIMARY KEY AUTO_INCREMENT,  
    nivel VARCHAR (50) NOT NULL  
);
```

```
CREATE TABLE Grau_de_Invasao (  
    id_grau INTEGER PRIMARY KEY AUTO_INCREMENT,  
    grau VARCHAR (50) NOT NULL  
);
```

```
CREATE TABLE Resultado (  
    id_resultado INTEGER PRIMARY KEY AUTO_INCREMENT,  
    resultado VARCHAR (700) NOT NULL,  
    data DATE,  
    observacoes VARCHAR (500)  
);
```

```
CREATE TABLE Prescricao_Medica (  
    id_prescricao INTEGER PRIMARY KEY AUTO_INCREMENT,  
    instrucoes VARCHAR (255) NOT NULL,  
    duracao TIME NOT NULL,
```



```
data DATE NOT NULL,  
fk_Medico_id_medico INTEGER  
);
```

```
CREATE TABLE Medicamentos (  
id_medicamento INTEGER PRIMARY KEY AUTO_INCREMENT,  
nome VARCHAR (255)  
);
```

```
CREATE TABLE Ferias (  
id_ferias INTEGER PRIMARY KEY AUTO_INCREMENT,  
dt_inicio DATE NOT NULL,  
dt_final DATE NOT NULL,  
nome VARCHAR (255),  
fk_Funcionario_id_funcionario INTEGER,  
fk_Medico_id_medico INTEGER  
);
```

```
CREATE TABLE Funcionario (  
id_funcionario INTEGER PRIMARY KEY AUTO_INCREMENT,  
nome VARCHAR (255) NOT NULL,  
endereco VARCHAR (255),  
Data_de_nascimento DATE,  
cpf CHAR (11) UNIQUE NOT NULL,  
fk_Cargo_id_cargo INTEGER,  
fk_Setor_id_setor INTEGER  
);
```

```
CREATE TABLE Cargo (  
    id_cargo INTEGER PRIMARY KEY AUTO_INCREMENT,  
    tipo_de_cargo VARCHAR (55)  
);
```

```
CREATE TABLE Setor (  
    id_setor INTEGER PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR (55)  
);
```

```
CREATE TABLE Medico (  
    id_medico INTEGER PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR (255) NOT NULL,  
    RGM CHAR (10) UNIQUE NOT NULL,  
    telefone CHAR (14) NOT NULL,  
    e_mail VARCHAR (255),  
    fk_Especialidade_id_especialidade INTEGER  
);
```

```
CREATE TABLE Especialidade (  
    id_especialidade INTEGER PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR (255)  
);
```

```
CREATE TABLE Realiza (  
    fk_Agendamento_id_agendamento INTEGER,  
    fk_Paciente_id_paciente INTEGER  
);
```

```
CREATE TABLE Gera (  
    fk_Consulta_id_consulta INTEGER,  
    fk_Agendamento_id_agendamento INTEGER,  
    data DATE NOT NULL,  
    hora DATE NOT NULL  
);
```

```
CREATE TABLE Gera (  
    fk_Medicamentos_id_medicamento INTEGER,  
    fk_Prescricao_Medica_id_prescricao INTEGER  
);
```

```
CREATE TABLE Gera (  
    fk_Consulta_id_consulta INTEGER,  
    fk_Exame_id_exame INTEGER,  
    data DATE NOT NULL,  
    hora DATE NOT NULL  
);
```

```
CREATE TABLE Possui (  
    fk_Agendamento_id_agendamento INTEGER,  
    fk_Pagamento_Convenio_id_pagamento INTEGER,  
    fk_Pagamento_Convenio_id_convenio INTEGER  
);
```

```
CREATE TABLE Possui (  
    fk_Consulta_id_consulta INTEGER,
```

```
fk_Medico_id_medico INTEGER  
);
```

```
CREATE TABLE Gera (  
    fk_Consulta_id_consulta INTEGER,  
    fk_Prescricao_Medica_id_prescricao INTEGER  
);
```

```
CREATE TABLE Possui (  
    fk_Medico_id_medico INTEGER,  
    fk_Especialidade_id_especialidade INTEGER  
);
```

```
ALTER TABLE Tipo_de_Pagamento ADD CONSTRAINT FK_Tipo_de_Pagamento_2  
    FOREIGN KEY (fk_Pagamento_Convenio_id_pagamento, ???)  
    REFERENCES Pagamento_Convenio (id_pagamento, ???)  
    ON DELETE RESTRICT;
```

```
ALTER TABLE Exame ADD CONSTRAINT FK_Exame_2  
    FOREIGN KEY (fk_Resultado_id_resultado)  
    REFERENCES Resultado (id_resultado)  
    ON DELETE RESTRICT;
```

```
ALTER TABLE Exame ADD CONSTRAINT FK_Exame_3  
    FOREIGN KEY (fk_Tipo_de_Exame_id_tipo_exame)  
    REFERENCES Tipo_de_Exame (id_tipo_exame)  
    ON DELETE CASCADE;
```

```
ALTER TABLE Exame ADD CONSTRAINT FK_Exame_4
```

```
FOREIGN KEY (fk_Grau_de_Invasao_id_grau)
```

```
REFERENCES Grau_de_Invasao (id_grau)
```

```
ON DELETE CASCADE;
```

```
ALTER TABLE Prescricao_Medica ADD CONSTRAINT FK_Prescricao_Medica_2
```

```
FOREIGN KEY (fk_Medico_id_medico)
```

```
REFERENCES Medico (id_medico)
```

```
ON DELETE CASCADE;
```

```
ALTER TABLE Ferias ADD CONSTRAINT FK_Ferias_2
```

```
FOREIGN KEY (fk_Medico_id_medico)
```

```
REFERENCES Medico (id_medico)
```

```
ON DELETE CASCADE;
```

```
ALTER TABLE Funcionario ADD CONSTRAINT FK_Funcionario_2
```

```
FOREIGN KEY (fk_Cargo_id_cargo)
```

```
REFERENCES Cargo (id_cargo)
```

```
ON DELETE RESTRICT;
```

```
ALTER TABLE Funcionario ADD CONSTRAINT FK_Funcionario_3
```

```
FOREIGN KEY (fk_Setor_id_setor)
```

```
REFERENCES Setor (id_setor)
```

```
ON DELETE RESTRICT;
```

```
ALTER TABLE Realiza ADD CONSTRAINT FK_Realiza_1
```

```
FOREIGN KEY (fk_Agendamento_id_agendamento)
```

```
REFERENCES Agendamento (id_agendamento)
```

ON DELETE RESTRICT;

ALTER TABLE Realiza ADD CONSTRAINT FK\_Realiza\_2

FOREIGN KEY (fk\_Paciente\_id\_paciente)

REFERENCES Paciente (id\_paciente)

ON DELETE SET NULL;

ALTER TABLE Gera ADD CONSTRAINT FK\_Gera\_1

FOREIGN KEY (fk\_Consulta\_id\_consulta)

REFERENCES Consulta (id\_consulta)

ON DELETE RESTRICT;

ALTER TABLE Gera ADD CONSTRAINT FK\_Gera\_2

FOREIGN KEY (fk\_Agendamento\_id\_agendamento)

REFERENCES Agendamento (id\_agendamento)

ON DELETE SET NULL;

ALTER TABLE Gera ADD CONSTRAINT FK\_Gera\_1

FOREIGN KEY (fk\_Medicamentos\_id\_medicamento)

REFERENCES Medicamentos (id\_medicamento)

ON DELETE RESTRICT;

ALTER TABLE Gera ADD CONSTRAINT FK\_Gera\_2

FOREIGN KEY (fk\_Prescricao\_Medica\_id\_prescricao)

REFERENCES Prescricao\_Medica (id\_prescricao)

ON DELETE SET NULL;

ALTER TABLE Possui ADD CONSTRAINT FK\_Possui\_1

```
FOREIGN KEY (fk_Funcionario_id_funcionario)
REFERENCES Funcionario (id_funcionario)
ON DELETE RESTRICT;
```

```
ALTER TABLE Possui ADD CONSTRAINT FK_Possui_2
FOREIGN KEY (fk_Ferias_id_ferias)
REFERENCES Ferias (id_ferias)
ON DELETE SET NULL;
```

```
ALTER TABLE Gera ADD CONSTRAINT FK_Gera_1
FOREIGN KEY (fk_Consulta_id_consulta)
REFERENCES Consulta (id_consulta)
ON DELETE SET NULL;
```

```
ALTER TABLE Gera ADD CONSTRAINT FK_Gera_2
FOREIGN KEY (fk_Exame_id_exame)
REFERENCES Exame (id_exame)
ON DELETE SET NULL;
```

```
ALTER TABLE Possui ADD CONSTRAINT FK_Possui_1
FOREIGN KEY (fk_Agendamento_id_agendamento)
REFERENCES Agendamento (id_agendamento)
ON DELETE RESTRICT;
```

```
ALTER TABLE Possui ADD CONSTRAINT FK_Possui_2
FOREIGN KEY (fk_Pagamento_Convenio_id_pagamento,
fk_Pagamento_Convenio_id_convenio)
REFERENCES Pagamento_Convenio (id_pagamento, id_convenio)
```

ON DELETE RESTRICT;

ALTER TABLE Possui ADD CONSTRAINT FK\_Possui\_1

FOREIGN KEY (fk\_Consulta\_id\_consulta)

REFERENCES Consulta (id\_consulta)

ON DELETE SET NULL;

ALTER TABLE Possui ADD CONSTRAINT FK\_Possui\_2

FOREIGN KEY (fk\_Medico\_id\_medico)

REFERENCES Medico (id\_medico)

ON DELETE SET NULL;

ALTER TABLE Gera ADD CONSTRAINT FK\_Gera\_1

FOREIGN KEY (fk\_Consulta\_id\_consulta)

REFERENCES Consulta (id\_consulta)

ON DELETE RESTRICT;

ALTER TABLE Gera ADD CONSTRAINT FK\_Gera\_2

FOREIGN KEY (fk\_Prescricao\_Medica\_id\_prescricao)

REFERENCES Prescricao\_Medica (id\_prescricao)

ON DELETE SET NULL;

ALTER TABLE Possuii ADD CONSTRAINT FK\_Possuii\_1

FOREIGN KEY (fk\_Medico\_id\_medico)

REFERENCES Medico (id\_medico)

ON DELETE SET NULL;

ALTER TABLE Possuii ADD CONSTRAINT FK\_Possuii\_2



FOREIGN KEY (fk\_Especialidade\_id\_especialidade)

REFERENCES Especialidade (id\_especialidade)

ON DELETE SET NULL;