Universidade Católica de Brasília

Curso de Engenharia de Software
Trabalho de Laboratório de Banco de Dados

Aplicação para Acompanhamento de Doenças Crônicas

Autores:Lara Paiva, Karolline Sena, Paloma Vitória, Lucas Moreira, Ketlen Rebeca
Orientador: Jeferson

Brasília - DF
2025

1. Introdução ao Projeto

O presente projeto tem como escopo a modelagem e implementação de um banco de dados para um aplicativo de saúde móvel (mHealth), cujo objetivo primordial é o acompanhamento de Doenças Crônicas Não Transmissíveis (DCNTs), tais como diabetes, hipertensão e diversas doenças cardiovasculares.

Este aplicativo será uma ferramenta essencial para pacientes e profissionais de saúde, proporcionando uma base de dados estruturada, organizada e segura para o registro de informações clínicas, o monitoramento de tratamentos e o suporte à tomada de decisões clínicas.

1.1. Contexto

As DCNTs representam um dos maiores desafios para a saúde pública global, exigindo monitoramento contínuo e uma adesão rigorosa aos planos de tratamento. A falta de registro consistente de sintomas e o esquecimento na administração de medicamentos são obstáculos frequentes que dificultam a gestão eficaz dessas condições pelos profissionais de saúde.

Nesse cenário, um aplicativo de saúde móvel emerge como uma solução estratégica para centralizar informações de saúde, estabelecendo uma comunicação eficiente entre paciente e médico. O banco de dados, como alicerce desse sistema, garante que os dados sejam armazenados de forma estruturada, consistente e confiável, promovendo a segurança e a disponibilidade das informações.

1.2. Justificativa

A concepção e implementação deste banco de dados são justificadas pelos seguintes benefícios cruciais:

- Empoderamento do Paciente: Permite que os pacientes registrem seus dados de saúde diariamente, promovendo maior autonomia e engajamento no acompanhamento de sua própria evolução clínica.

- Acompanhamento Médico Preciso: Oferece aos profissionais de saúde acesso a históricos clínicos completos e organizados, possibilitando análises mais aprofundadas e a tomada de decisões terapêuticas mais informadas e personalizadas.
- Prevenção de Crises: Facilita a detecção precoce de padrões de risco e alterações significativas nos dados do paciente, permitindo intervenções preventivas antes da ocorrência de complicações agudas.
- Gestão Otimizada de Medicamentos: Através de lembretes programáveis e registros claros, minimiza a incidência de esquecimentos e erros de dosagem, contribuindo para a adesão ao tratamento.
- Organização Centralizada de Dados: Consolida todas as informações relevantes em um único sistema, otimizando o acesso e a gestão dos dados de saúde.

2. Modelagem Conceitual

A modelagem conceitual representa a estrutura abstrata dos dados, utilizando o Diagrama Entidade-Relacionamento (DER) para ilustrar as entidades e seus relacionamentos.

2.1. Entidades Principais

As entidades fundamentais identificadas para o sistema são:

- PACIENTE: Representa o indivíduo usuário do aplicativo, que possui uma ou mais doenças crônicas.
- MÉDICO: Representa o profissional de saúde responsável pelo acompanhamento dos pacientes.
- DOENÇA_CRÔNICA: Refere-se às condições de saúde monitoradas (e.g., Diabetes Tipo 2, Hipertensão).
- MEDICAMENTO: Corresponde aos fármacos que podem ser prescritos para o tratamento.
- PRESCRIÇÃO: Entidade associativa que estabelece a ligação entre paciente, médico e medicamento, detalhando o plano de tratamento.

- MEDIÇÃO: Entidade generalizada que engloba os registros clínicos realizados pelo paciente em determinado momento (e.g., glicemia, pressão arterial).
- LEMBRETE: Notificações programadas para alertar sobre a administração de medicamentos ou a realização de medições.

2.2. Exemplos de Atributos

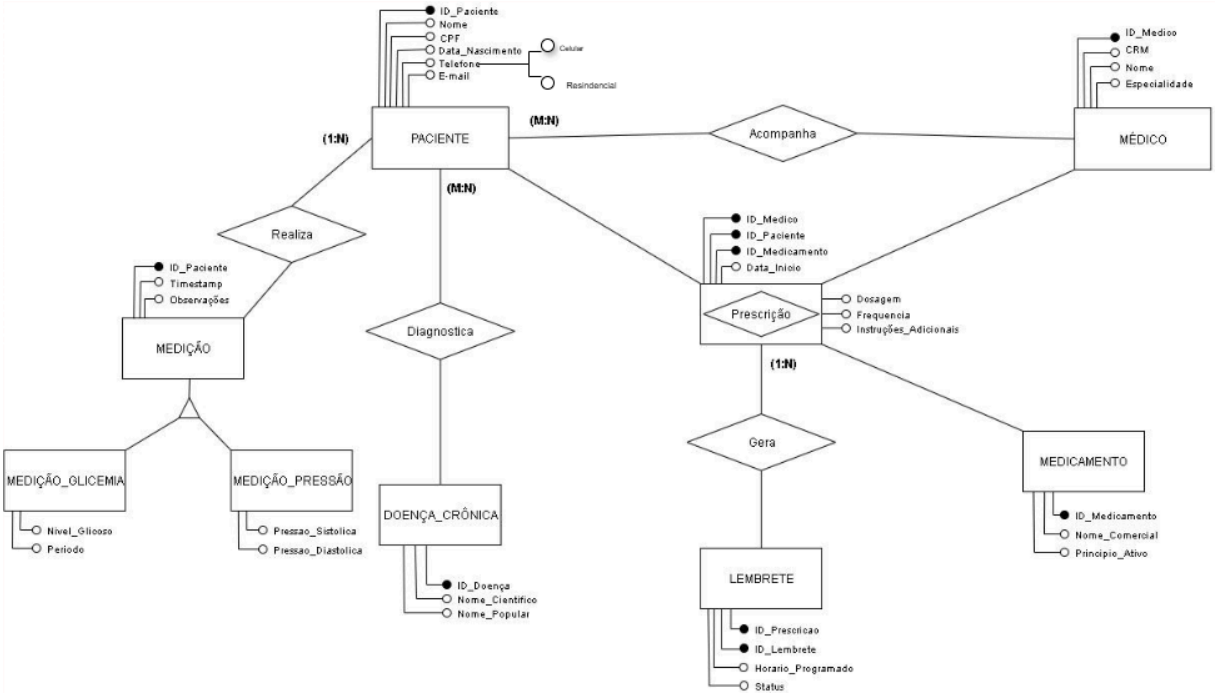
A seguir, são apresentados exemplos de atributos para cada entidade:

- PACIENTE: ID_Paciente (Chave Primária - PK), CPF, Email, Telefones (multivalorado).
- MÉDICO: ID_Medico (PK), CRM, Especialidade.
- DOENÇA_CRÔNICA: ID_Doenca (PK), Nome_Cientifico, Nome_Popular.
- MEDICAMENTO: ID_Medicamento (PK), Nome_Comercial, Principio_Ativo.
- PRESCRIÇÃO: Data_Inicio, Dosagem, Frequencia, Instrucoes_Adicionais.
- MEDIÇÃO_GLICEMIA: Nivel_Glicose, Periodo.
- MEDIÇÃO_PRESSÃO: Pressao_Sistolica, Pressao_Diastolica.
- LEMBRETE: Horario_Programado, Status.

2.3. Tratamento de Atributos Multivalorados

O atributo Telefones da entidade PACIENTE foi concebido como multivalorado, uma vez que um paciente pode possuir múltiplos números de contato (e.g., telefone pessoal, profissional, de emergência). A resolução deste atributo será abordada e detalhada na etapa de modelagem lógica, através da criação de uma entidade auxiliar.

2.4. Diagrama Entidade-Relacionamento (DER) Conceitual



3. Modelagem Lógica

A modelagem lógica consiste na tradução do DER conceitual para o modelo relacional, organizando as entidades em tabelas, definindo chaves primárias e estrangeiras, e resolvendo atributos multivalorados e especializações para garantir a integridade e a eficiência do banco de dados.

3.1. Regras de Conversão para o Modelo Relacional

As seguintes regras foram aplicadas na conversão do modelo conceitual para o lógico:

- Entidades Fortes: Convertidas diretamente em tabelas independentes (e.g., PACIENTE, MÉDICO, DOENÇA_CRÔNICA, MEDICAMENTO).
- Entidades Associativas: Transformadas em tabelas que incorporam as chaves primárias das entidades participantes como chaves estrangeiras, formando uma chave primária composta (e.g., PRESCRIÇÃO).
- Entidades Fracas: Herdam a chave primária da entidade forte à qual estão associadas, formando uma chave primária composta (e.g., MEDIÇÃO depende de PACIENTE).

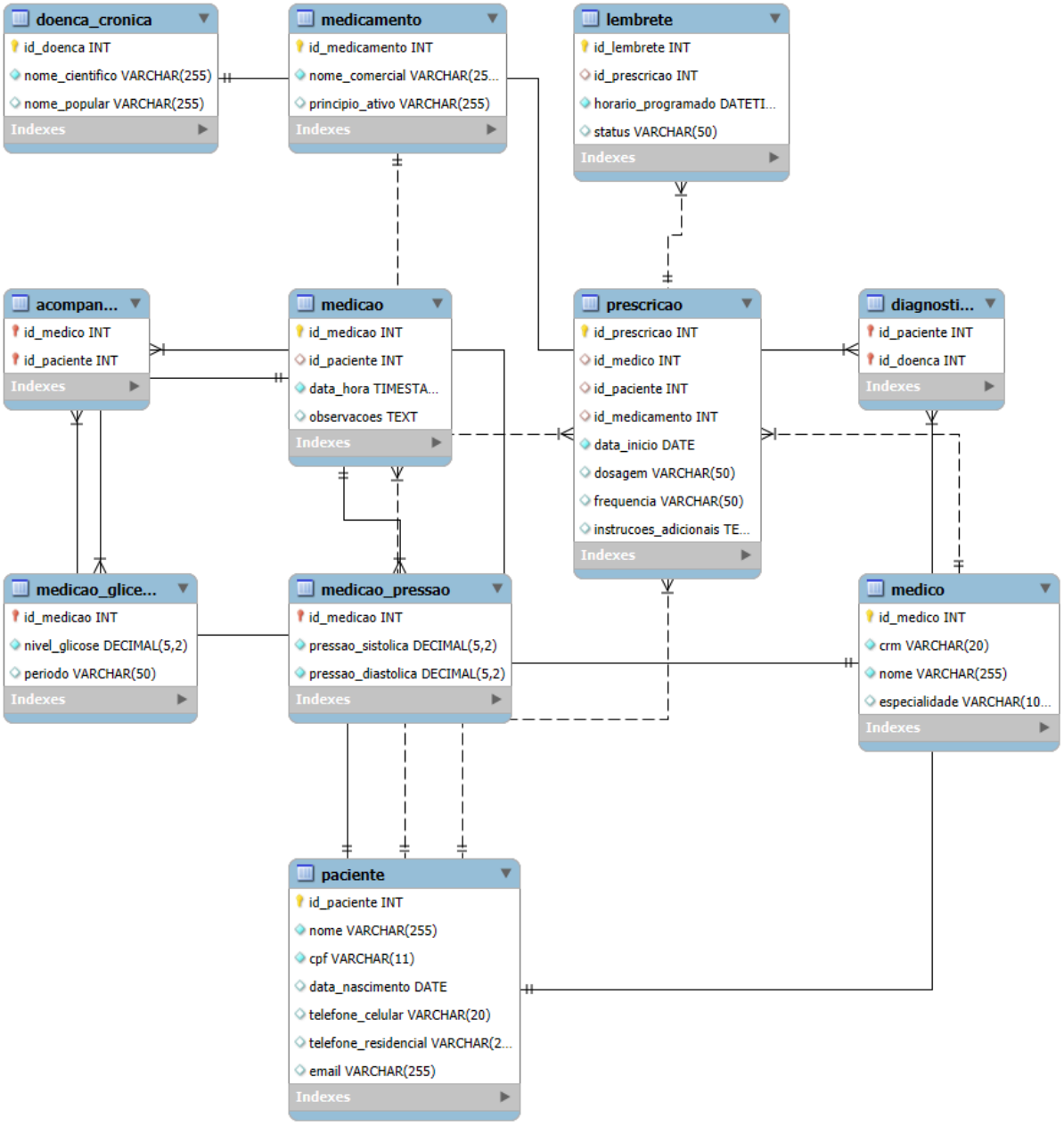
- Especializações: Cada tipo de medição (MEDIÇÃO_GLICEMIA e MEDIÇÃO_PRESSÃO) é representado como uma tabela separada, que herda a chave primária da entidade generalizada MEDIÇÃO.
- Atributos Multivalorados: Resolvidos através da criação de uma tabela auxiliar, que estabelece um relacionamento um-para-muitos com a entidade original (e.g., TELEFONE_PACIENTE para o atributo Telefones de PACIENTE).

3.2. Estrutura das Tabelas

A estrutura das tabelas no modelo lógico é apresentada a seguir, com a indicação das chaves primárias (PK) e estrangeiras (FK):

- PACIENTE (ID_Paciente PK, CPF, Email)
- TELEFONE_PACIENTE (ID_Paciente PK, FK; Telefone PK)
- MÉDICO (ID_Medico PK, CRM, Especialidade)
- DOENÇA_CRÔNICA (ID_Doenca PK, Nome_Cientifico, Nome_Popular)
- MEDICAMENTO (ID_Medicamento PK, Nome_Comercial, Principio_Ativo)
- PRESCRIÇÃO (ID_Paciente PK, FK; ID_Medico PK, FK; ID_Medicamento PK, FK; Data_Inicio PK; Dosagem, Frequencia, Instrucoes_Adicionais)
- MEDIÇÃO (ID_Paciente PK, FK; Timestamp PK; Observacoes)
- MEDIÇÃO_GLICEMIA (ID_Paciente PK, FK; Timestamp PK, FK; Nivel_Glicose, Periodo)
- MEDIÇÃO_PRESSÃO (ID_Paciente PK, FK; Timestamp PK, FK; Pressao_Sistolica, Pressao_Diastolica)
- LEMBRETE (ID_Lembrete PK; Horario_Programado, Status, ID_Prescricao FK)

3.3. Diagrama Lógico



4. Modelagem Física

A modelagem física adapta o modelo lógico para o Sistema Gerenciador de Banco de Dados (SGBD) MySQL, definindo tipos de dados específicos, restrições de integridade e a estrutura física das tabelas.

4.1. Regras de Integridade e Otimização

Para garantir a consistência e a integridade dos dados, foram estabelecidas as seguintes regras:

- Chaves Estrangeiras: Implementadas para assegurar a integridade referencial, prevenindo a existência de dados órfãos (e.g., uma medição não pode existir sem um paciente associado).
- `ON DELETE CASCADE`: Utilizado em relacionamentos de dependência (e.g., se um registro de paciente for excluído, seus telefones e medições associadas serão automaticamente removidos).
- Restrições `UNIQUE`: Aplicadas a atributos críticos como `CPF` e `CRM` para garantir a unicidade dos registros.
- Normalização: O modelo foi projetado para atender às formas normais, minimizando redundâncias e anomalias de atualização.

4.2. Diagrama Físico

```
— cria o banco de dados e seleciona para uso
create database acompanhamento_doencas_cronicas;
use acompanhamento_doencas_cronicas;

— tabela paciente
create table paciente (
  id_paciente int primary key auto_increment,
  nome varchar(255) not null,
  cpf varchar(11) unique not null,
  data_nascimento date,
  telefone_celular varchar(20),
  telefone_residencial varchar(20),
  email varchar(255)
```

```
);

— tabela medico
create table medico (
    id_medico int primary key auto_increment,
    crm varchar(20) unique not null,
    nome varchar(255) not null,
    especialidade varchar(100)
);

— tabela doenca_cronica
create table doenca_cronica (
    id_doenca int primary key auto_increment,
    nome_cientifico varchar(255) unique not null,
    nome_popular varchar(255)
);

— tabela medicamento
create table medicamento (
    id_medicamento int primary key auto_increment,
    nome_comercial varchar(255) not null,
    principio_ativo varchar(255)
);

— tabela de relacionamento acompanha
```

```
create table acompanha (
    id_medico int,
    id_paciente int,
    primary key (id_medico, id_paciente),
    foreign key (id_medico) references medico(id_medico)
    on update cascade
    on delete no action,
    foreign key (id_paciente) references paciente(id_paciente)
    on update cascade
    on delete no action
);

— tabela de relacionamento diagnostica
create table diagnostica (
    id_paciente int,
    id_doenca int,
    primary key (id_paciente, id_doenca),
    foreign key (id_paciente) references paciente(id_paciente)
    on update cascade
    on delete no action,
    foreign key (id_doenca) references doenca_cronica(id_doenca)
    on update cascade
    on delete no action
);
```

```
— tabela prescricao
create table prescricao (
    id_prescricao int primary key auto_increment,
    id_medico int,
    id_paciente int,
    id_medicamento int,
    data_inicio date not null,
    dosagem varchar(50),
    frequencia varchar(50),
    instrucoes_adicionais text,
    foreign key (id_medico) references medico(id_medico)
    on update cascade
    on delete no action,
    foreign key (id_paciente) references paciente(id_paciente)
    on update cascade
    on delete no action,
    foreign key (id_medicamento) references medicamento(id_medicamento)
    on update cascade
    on delete no action
);

— tabela lembrete
create table lembrete (
    id_lembrete int primary key auto_increment,
    id_prescricao int,
```

```
    horario_programado datetime not null,
    status varchar(50),
    foreign key (id_prescricao) references prescricao(id_prescricao)
    on update cascade
    on delete no action
);

— tabela base para medições (COM A CORREÇÃO)
create table medicao (
    id_medicao int primary key auto_increment,
    id_paciente int,
    data_hora timestamp not null default current_timestamp,
    observacoes text,
    foreign key (id_paciente) references paciente(id_paciente)
    on update cascade
    on delete no action
);

— tabela para medição de glicemia
create table medicao_glicemia (
    id_medicao int primary key,
    nivel_glicose decimal(5, 2) not null,
    periodo varchar(50),
    foreign key (id_medicao) references medicao(id_medicao)
    on update cascade
```

```
    on delete cascade
);

— tabela para medição de pressão
create table medicao_pressao (
    id_medicao int primary key,
    pressao_sistolica decimal(5, 2) not null,
    pressao_diastolica decimal(5, 2) not null,
    foreign key (id_medicao) references medicao(id_medicao)
    on update cascade
    on delete cascade
);

— Inserindo dados na tabela PACIENTE
INSERT INTO paciente (nome, cpf, data_nascimento, telefone_celular, telefone_residencial, email)
VALUES
('Carlos Santana', '11122233344', '1955-07-20', '11987654321', '1145678901',
'carlos.santana@email.com'),
('Mariana Costa', '22233344455', '1982-03-15', '21998877665', '2123456789',
'mariana.costa@email.com'),
('João Pedro', '33344455566', '1990-11-01', '31988887777', '3134567890', 'joao.pedro@email.com'),
('Ana Oliveira', '44455566677', '1978-01-30', '41977776666', '4145678901', 'ana.oliveira@email.com');

— Inserindo dados na tabela MEDICO
INSERT INTO medico (crm, nome, especialidade) VALUES
('12345-SP', 'Dr. Roberto Alves', 'Cardiologia'),
```

('54321-RJ', 'Dra. Lúcia Martins', 'Endocrinologia'),

('67890-MG', 'Dr. Fernando Dias', 'Clínica Geral');

— Inserindo dados na tabela DOENCA_CRONICA

INSERT INTO doenca_cronica (nome_cientifico, nome_popular) VALUES

('Hipertensão Arterial Sistêmica', 'Pressão Alta'),

('Diabetes Mellitus Tipo 2', 'Diabetes');

— Inserindo dados na tabela MEDICAMENTO

INSERT INTO medicamento (nome_comercial, principio_ativo) VALUES

('Losartana Potássica 50mg', 'Losartana'),

('Metformina 850mg', 'Cloridrato de Metformina'),

('Atenolol 25mg', 'Atenolol'),

('Glibenclamida 5mg', 'Glibenclamida');

— Relacionando médicos e pacientes na tabela ACOMPANHA

INSERT INTO acompanha (id_medico, id_paciente) VALUES

(1, 1), — Dr. Roberto acompanha Carlos

(2, 2), — Dra. Lúcia acompanha Mariana

(1, 3), — Dr. Roberto acompanha João Pedro

(2, 4); — Dra. Lúcia acompanha Ana

— Relacionando diagnósticos aos pacientes na tabela DIAGNOSTICA

INSERT INTO diagnostica (id_paciente, id_doenca) VALUES

(1, 1), — Carlos tem Hipertensão

(2, 2), — Mariana tem Diabetes

(3, 1), — João Pedro tem Hipertensão

(4, 2); — Ana tem Diabetes

— Criando prescrições na tabela PRESCRICAO

INSERT INTO prescricao (id_medico, id_paciente, id_medicamento, data_inicio, dosagem, frequencia, instrucoes_adicionais) VALUES

(1, 1, 1, '2024-01-10', '1 comprimido', '1 vez ao dia', 'Tomar pela manhã.'),

(2, 2, 2, '2024-02-15', '1 comprimido', '2 vezes ao dia', 'Tomar após o café da manhã e após o jantar.'),

(1, 3, 3, '2024-03-20', '1 comprimido', '1 vez ao dia', 'Tomar antes de dormir.'),

(2, 4, 4, '2024-04-05', '1 comprimido', '1 vez ao dia', 'Tomar 30 minutos antes do almoço.');

— Criando lembretes para as prescrições na tabela LEMBRETE

INSERT INTO lembrete (id_prescricao, horario_programado, status) VALUES

(1, '2025-09-28 08:00:00', 'Pendente'),

(2, '2025-09-28 09:00:00', 'Pendente'),

(2, '2025-09-28 20:00:00', 'Pendente'),

(3, '2025-09-28 22:00:00', 'Pendente');

— Inserindo medições genéricas na tabela MEDICAO

INSERT INTO medicao (id_paciente, data_hora, observacoes) VALUES

(1, '2025-09-27 09:00:00', 'Medição de rotina da pressão.'),

(2, '2025-09-27 08:30:00', 'Medição de glicose em jejum.'),

(1, '2025-09-28 09:05:00', 'Medição após caminhada leve.');

```
— Inserindo medições específicas de pressão na tabela MEDICAO_PRESSAO
INSERT INTO medicao_pressao (id_medicao, pressao_sistolica, pressao_diastolica) VALUES
(1, 13.8, 8.5),
(3, 13.2, 8.1);

— Inserindo medições específicas de glicemia na tabela MEDICAO_GLICEMIA
INSERT INTO medicao_glicemia (id_medicao, nivel_glicose, periodo) VALUES
(2, 99.00, 'Jejum');

— 3. MANIPULAÇÃO E CONSULTA DE DADOS (EXEMPLOS)

— Consulta 1: Listar todos os pacientes e seus respectivos médicos.
SELECT
    p.nome AS Nome_Paciente,
    m.nome AS Nome_Medico,
    m.especialidade AS Especialidade_Medico
FROM
    paciente p
INNER JOIN
    acompanha a ON p.id_paciente = a.id_paciente
INNER JOIN
    medico m ON a.id_medico = m.id_medico
ORDER BY
    Nome_Medico, Nome_Paciente;
```

```
— Consulta 2: Exibir todas as prescrições feitas para um paciente específico (ex: Mariana Costa).
SELECT
    p.nome AS Nome_Paciente,
    med.nome_comercial AS Medicamento,
    pr.dosagem,
    pr.frequencia,
    pr.data_inicio AS Inicio_Tratamento,
    pr.instrucoes_adicionais
FROM
    prescricao pr
INNER JOIN
    paciente p ON pr.id_paciente = p.id_paciente
INNER JOIN
    medicamento med ON pr.id_medicamento = med.id_medicamento
WHERE
    p.nome = 'Mariana Costa';

— Atualização 1: Atualizar o status de um lembrete para 'Tomado'.
UPDATE lembrete
SET
    status = 'Tomado'
WHERE
    id_lembrete = 1;

— Atualização 2: Alterar o número de telefone de um paciente.
```

```
UPDATE paciente
SET
  telefone_celular = '11999998888',
  email = 'c.santana.novo@email.com'
WHERE
  id_paciente = 1;

— Verificando a atualização do paciente Carlos Santana
SELECT * FROM paciente WHERE id_paciente = 1;
```

5. Implementação do Banco de Dados

5.1. Scripts SQL Necessários

A implementação do banco de dados será realizada através de scripts SQL, que incluirão:

- `CREATE TABLE`: Scripts para a criação de todas as dez tabelas, com suas respectivas definições de colunas, chaves primárias, chaves estrangeiras e outras restrições de integridade.
- `INSERT INTO`: Scripts para a inserção de dados fictícios, essenciais para a realização de testes e validação do modelo.
- `SELECT`: Scripts de consulta que demonstram a capacidade de recuperação de dados, utilizando `JOINs` múltiplos e funções de agregação para análises complexas.
- `UPDATE`: Scripts para a atualização de dados, como a modificação do status de lembretes ou a correção de registros de medições.

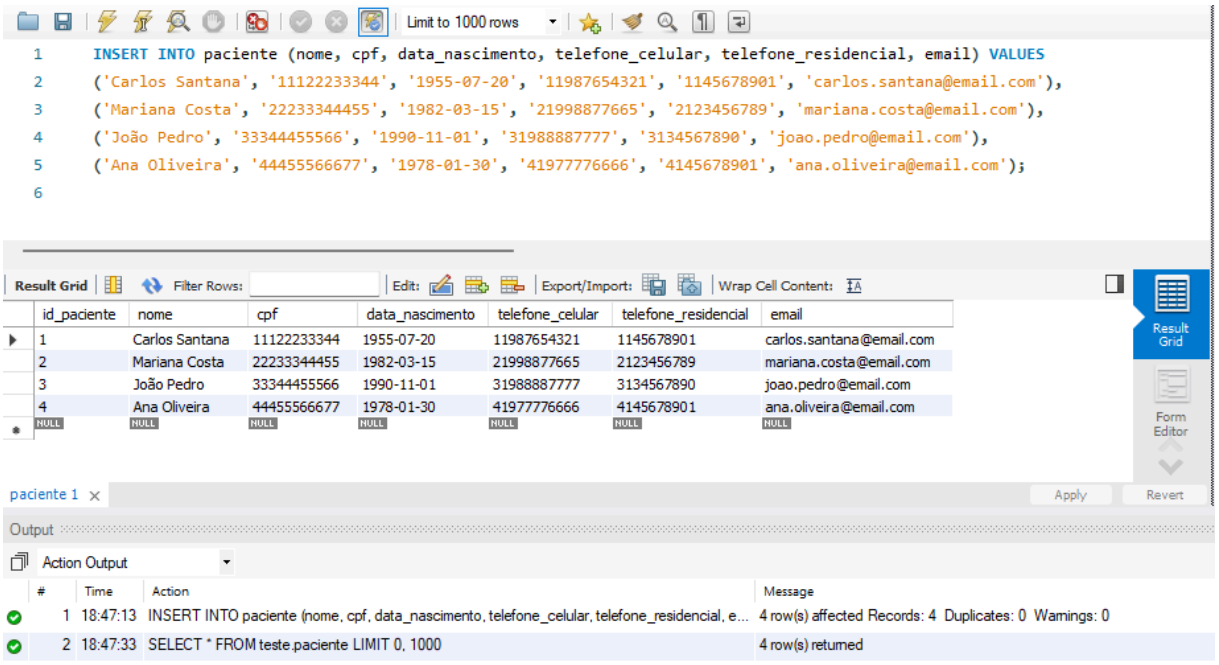
5.2. Repositório de Código

O código-fonte completo para a criação e manipulação do banco de dados está hospedado no seguinte repositório:

https://github.com/lucas0mp/BancoDeDadosGrupo.git

5.3. Evidências de Funcionamento

- Inserindo dados na tabela PACIENTE



- Inserindo dados na tabela MEDICO

```sql
INSERT INTO medico (crm, nome, especialidade) VALUES
('12345-SP', 'Dr. Roberto Alves', 'Cardiologia'),
('54321-RJ', 'Dra. Lúcia Martins', 'Endocrinologia'),
('67890-MG', 'Dr. Fernando Dias', 'Clínica Geral');
```

Result Grid

id_medico	crm	nome	especialidade
1	12345-SP	Dr. Roberto Alves	Cardiologia
2	54321-RJ	Dra. Lúcia Martins	Endocrinologia
3	67890-MG	Dr. Fernando Dias	Clínica Geral

- Inserindo dados na tabela DOENCA_CRONICA

```sql
INSERT INTO doenca_cronica (nome_cientifico, nome_popular) VALUES
('Hipertensão Arterial Sistêmica', 'Pressão Alta'),
('Diabetes Mellitus Tipo 2', 'Diabetes');
```

Result Grid

id_doenca	nome_cientifico	nome_popular
1	Hipertensão Arterial Sistêmica	Pressão Alta
2	Diabetes Mellitus Tipo 2	Diabetes

- Relacionando médicos e pacientes na tabela ACOMPANHA

```sql
INSERT INTO acompanha (id_medico, id_paciente) VALUES
(1, 1), -- Dr. Roberto acompanha Carlos
(2, 2), -- Dra. Lúcia acompanha Mariana
(1, 3), -- Dr. Roberto acompanha João Pedro
(2, 4); -- Dra. Lúcia acompanha Ana
```

Result Grid

id_medico	id_paciente
1	1
2	2
1	3
2	4

- Relacionando diagnósticos aos pacientes na tabela DIAGNOSTICA

```
INSERT INTO diagnostica (id_paciente, id_doenca) VALUES
(1, 1), -- Carlos tem Hipertensão
(2, 2), -- Mariana tem Diabetes
(3, 1), -- João Pedro tem Hipertensão
(4, 2); -- Ana tem Diabetes
```

Result Grid

id_paciente	id_doenca
1	1
3	1
2	2
4	2
NULL	NULL

diagnostica 1

- Criando lembretes para as prescrições na tabela LEMBRETE

```
INSERT INTO lembrete (id_prescricao, horario_programado, status) VALUES
(1, '2025-09-28 08:00:00', 'Pendente'),
(2, '2025-09-28 09:00:00', 'Pendente'),
(2, '2025-09-28 20:00:00', 'Pendente'),
(3, '2025-09-28 22:00:00', 'Pendente');
```

Result Grid

id_lembrete	id_prescricao	horario_programado	status
1	1	2025-09-28 08:00:00	Pendente
2	2	2025-09-28 09:00:00	Pendente
3	2	2025-09-28 20:00:00	Pendente
4	3	2025-09-28 22:00:00	Pendente
NULL	NULL	NULL	NULL

- Inserindo medições genéricas na tabela MEDICAO

```
INSERT INTO medicao (id_paciente, data_hora, observacoes) VALUES
(1, '2025-09-27 09:00:00', 'Medição de rotina da pressão.'),
(2, '2025-09-27 08:30:00', 'Medição de glicose em jejum.'),
(1, '2025-09-28 09:05:00', 'Medição após caminhada leve.');
```

Result Grid

id_medicao	id_paciente	data_hora	observacoes
1	1	2025-09-27 09:00:00	Medição de rotina da pressão.
2	2	2025-09-27 08:30:00	Medição de glicose em jejum.
3	1	2025-09-28 09:05:00	Medição após caminhada leve.
NULL	NULL	NULL	NULL

- Criando prescrições na tabela PRESCRICAO

```
INSERT INTO prescricao (id_medico, id_paciente, id_medicamento, data_inicio, dosagem, frequencia, instrucoes_adicionais)
(1, 1, 1, '2024-01-10', '1 comprimido', '1 vez ao dia', 'Tomar pela manhã.'),
(2, 2, 2, '2024-02-15', '1 comprimido', '2 vezes ao dia', 'Tomar após o café da manhã e após o jantar.'),
(1, 3, 3, '2024-03-20', '1 comprimido', '1 vez ao dia', 'Tomar antes de dormir.'),
(2, 4, 4, '2024-04-05', '1 comprimido', '1 vez ao dia', 'Tomar 30 minutos antes do almoço.');
```

Result Grid

id_prescricao	id_medico	id_paciente	id_medicamento	data_inicio	dosagem	frequencia	instrucoes_adicionais
1	1	1	1	2024-01-10	1 comprimido	1 vez ao dia	Tomar pela manhã.
2	2	2	2	2024-02-15	1 comprimido	2 vezes ao dia	Tomar após o café da manhã e após o jantar.
3	1	3	3	2024-03-20	1 comprimido	1 vez ao dia	Tomar antes de dormir.
4	2	4	4	2024-04-05	1 comprimido	1 vez ao dia	Tomar 30 minutos antes do almoço.
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Inserindo medições específicas de pressão na tabela MEDICAO_PRESSAO

```
INSERT INTO medicao_pressao (id_medicao, pressao_sistolica, pressao_diastolica) VALUES
(1, 13.8, 8.5),
(3, 13.2, 8.1);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id_medicao	pressao_sistolica	pressao_diastolica
1	13.80	8.50
3	13.20	8.10
NULL	NULL	NULL

- Consulta 1: Listar todos os pacientes e seus respectivos médicos.

```
SELECT
    p.nome AS Nome_Paciente,
    m.nome AS Nome_Medico,
    m.especialidade AS Especialidade_Medico
FROM
    paciente p
INNER JOIN
    acompanha a ON p.id_paciente = a.id_paciente
INNER JOIN
    medico m ON a.id_medico = m.id_medico
ORDER BY
    Nome_Medico, Nome_Paciente;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Nome_Paciente	Nome_Medico	Especialidade_Medico
Carlos Santana	Dr. Roberto Alves	Cardiologia
João Pedro	Dr. Roberto Alves	Cardiologia
Ana Oliveira	Dra. Lúcia Martins	Endocrinologia
Mariana Costa	Dra. Lúcia Martins	Endocrinologia

- Consulta 2: Exibir todas as prescrições feitas para um paciente específico (ex: Mariana Costa).

```sql
SELECT
    p.nome AS Nome_Paciente,
    med.nome_comercial AS Medicamento,
    pr.dosagem,
    pr.frequencia,
    pr.data_inicio AS Inicio_Tratamento,
    pr.instrucoes_adicionais
FROM
    prescricao pr
INNER JOIN
    paciente p ON pr.id_paciente = p.id_paciente
INNER JOIN
    medicamento med ON pr.id_medicamento = med.id_medicamento
WHERE
    p.nome = 'Mariana Costa';
```

Result Grid

Nome_Paciente	Medicamento	dosagem	frequencia	Inicio_Tratamento	instrucoes_adicionais
Mariana Costa	Metformina 850mg	1 comprimido	2 vezes ao dia	2024-02-15	Tomar após o café da manhã e após o jantar.

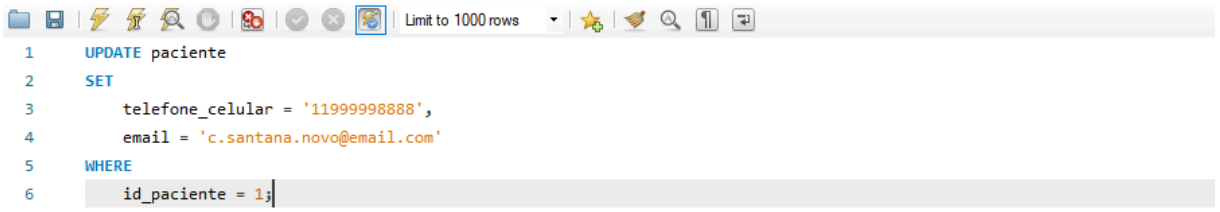
- Atualização 1: Atualizar o status de um lembrete para 'Tomado'.

```sql
UPDATE lembrete
SET
    status = 'Tomado'
WHERE
    id_lembrete = 1;
```

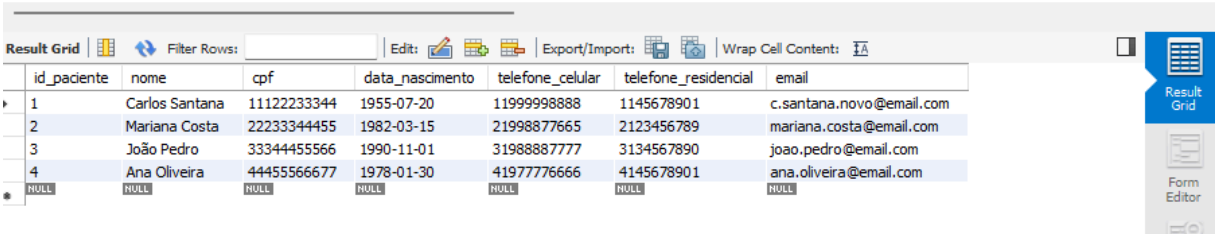
Result Grid

id_lembrete	id_prescricao	horario_programado	status
1	1	2025-09-28 08:00:00	Tomado
2	2	2025-09-28 09:00:00	Pendente
3	2	2025-09-28 20:00:00	Pendente
4	3	2025-09-28 22:00:00	Pendente
NULL	NULL	NULL	NULL

- Atualização 2: Alterar o número de telefone de um paciente.



```
UPDATE paciente
SET
    telefone_celular = '11999998888',
    email = 'c.santana.novo@email.com'
WHERE
    id_paciente = 1;
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id_paciente	nome	cpf	data_nascimento	telefone_celular	telefone_residencial	email
1	Carlos Santana	11122233344	1955-07-20	11999998888	1145678901	c.santana.novo@email.com
2	Mariana Costa	22233344455	1982-03-15	21998877665	2123456789	mariana.costa@email.com
3	João Pedro	33344455566	1990-11-01	31988887777	3134567890	joao.pedro@email.com
4	Ana Oliveira	44455566677	1978-01-30	41977776666	4145678901	ana.oliveira@email.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor

6. Conclusão

Este trabalho detalhou o processo completo de modelagem de dados para um aplicativo de acompanhamento de doenças crônicas, abrangendo as etapas de modelagem conceitual, lógica e física, culminando na implementação do banco de dados.

O modelo desenvolvido assegura:

- A organização eficiente das informações do paciente.
- O suporte robusto ao acompanhamento médico e à tomada de decisões clínicas.
- A flexibilidade necessária para futuras evoluções e expansões do sistema.
- A consistência e a segurança dos dados armazenados.

7. Referências Bibliográficas

- WIKIPEDIA. Modelo entidade–relacionamento. Disponível em: https://pt.wikipedia.org/wiki/Modelo_entidade_relacionamento. Acesso em: [Data de Acesso, 25 ago. 2025].
- CLICKUP. Entidade fraca em diagramas ER: um guia completo. Disponível em: https://clickup.com/pt-BR/blog/264674/entidade-fraca. Acesso em: [Data de Acesso, ex: 30 ago. 2025].
- ELMASRI, R.; NAVATHE, S. B. Sistemas de Banco de Dados. 7. ed. Pearson Education do Brasil, 2017.
- DATE, C. J. An Introduction to Database Systems. 8. ed. Addison-Wesley, 2003.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Database System Concepts. 7. ed. McGraw-Hill Education, 2019.