

UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
BANCO DE DADOS

Plataforma de Gestão de Cuidados para Idosos e Familiares

Victor André Lopes Brasileiro

Maceió

2025

RESUMO

ABSTRACT

Este projeto foi desenvolvido para a disciplina de Banco de Dados e propõe a criação de uma plataforma voltada à gestão dos cuidados de idosos no contexto da saúde e assistência domiciliar. O sistema tem como principal objetivo organizar e centralizar informações sobre agendamentos, registros de cuidadores e atendimentos, além de facilitar a comunicação entre os profissionais, os próprios idosos e seus familiares.

A solução pretende tornar o acompanhamento mais transparente e eficiente, permitindo que familiares tenham acesso rápido ao histórico de cuidados e possam avaliar os serviços prestados. O sistema oferece agendamento de visitas, controle de escalas, registro detalhado dos atendimentos e um canal de feedback para aprimorar continuamente o serviço. Com isso, busca-se promover mais confiança, organização e segurança para todos os envolvidos no processo de cuidado, garantindo atenção individualizada aos idosos e suporte direto às famílias.

Keywords: banco de dados; assistência domiciliar; idosos; gestão de cuidados.

SUMÁRIO

1. DESCRIÇÃO DO SISTEMA.....	5
2. REQUISITOS DO SISTEMA.....	6
3. DIAGRAMA ENTIDADE-RELACIONAMENTO (MER).....	7
4. DEPENDÊNCIAS FUNCIONAIS.....	8
4.2 ESQUEMATIZAÇÃO DO MER EM TABELAS.....	8
4.2.1. TABELA IDOSO.....	9
4.2.2 TABELA RESPONSÁVEL.....	10
4.2.3 TABELA CUIDADOR.....	11
4.2.4 TABELA AGENDAMENTO.....	12
4.2.5 TABELA AVALIAÇÃO.....	13
5. NORMALIZAÇÃO.....	14
5.1 PRIMEIRA FORMA NORMAL (1FN).....	14
5.1.1 TABELA IDOSO.....	14
5.1.2 TABELA RESPONSÁVEL.....	15
5.1.3 TABELA CUIDADOR.....	16
5.1.4 TABELA AGENDAMENTO E TABELA AVALIAÇÃO.....	17
5.2 SEGUNDA FORMA NORMAL (2FN).....	18
5.2.1 TABELA IDOSO.....	18
5.2.2 TABELA RESPONSÁVEL.....	19
5.2.3 TABELA CUIDADOR.....	19
5.2.4 TABELA AGENDAMENTO E AVALIAÇÃO.....	20
5.3 TERCEIRA FORMA NORMAL (3FN).....	21
5.3.1 TABELA IDOSO.....	21
5.3.2 TABELA RESPONSÁVEL.....	22
5.3.3 TABELA CUIDADOR.....	23
5.3.4 TABELA AGENDAMENTO.....	24
5.3.4 TABELA AVALIAÇÃO.....	24
5.4 FORMA NORMAL DE BOYCE-CODD (BCNF).....	25
5.4.1 TABELA IDOSO.....	25
5.4.2 TABELA RESPONSÁVEL.....	26
5.4.3 TABELA CUIDADOR.....	26
5.4.4 TABELA AGENDAMENTO.....	27
5.4.5 TABELA AVALIAÇÃO.....	27
5.5 RESULTADO DA NORMALIZAÇÃO.....	28
6. MODELO LÓGICO RELACIONAL.....	29
7. CONSULTAS SQL.....	30
7.1 INSERT, UPDATE, DELETE.....	30
7.2 SELECT E JOIN.....	31
7.3 AGREGAÇÕES.....	32
7.4 ORDENAÇÕES.....	33
7.5 SUBCONSULTAS.....	34
8. CONCLUSÃO.....	36
9. REFERÊNCIAS.....	37

INFORMAÇÕES DO ALUNO

Nome: Victor André Lopes Brasileiro

Matrícula: 202407269

Curso: Ciência da Computação

1. DESCRIÇÃO DO SISTEMA

O sistema será desenvolvido no contexto da saúde e da assistência domiciliar, com foco na gestão dos cuidados prestados a pessoas idosas e no apoio direto às suas famílias. A ideia é oferecer uma solução que ajude a organizar melhor os agendamentos, registrar cuidadores, acompanhar os atendimentos e, principalmente, facilitar a comunicação com os familiares responsáveis.

Os principais envolvidos nesse processo são os próprios idosos, que merecem atenção e cuidado individualizado, e os cuidadores, profissionais ou auxiliares que realizam tarefas essenciais como higiene, alimentação e companhia no dia a dia. Também são parte fundamental os familiares ou responsáveis, que acompanham de perto a saúde e o bem-estar do idoso e precisam ter acesso claro e direto às informações sobre os atendimentos realizados.

Quando pensei neste projeto, foi justamente a partir de algumas situações que observei na vida real: a dificuldade em organizar os horários das visitas, a falta de um histórico claro sobre quem cuidou de quem, e a insegurança de muitos familiares que não conseguem acompanhar de perto o que está acontecendo. Por isso, meu principal objetivo é melhorar a comunicação e o relacionamento entre cuidadores, idosos e suas famílias, promovendo mais confiança, organização e clareza em todo o processo.

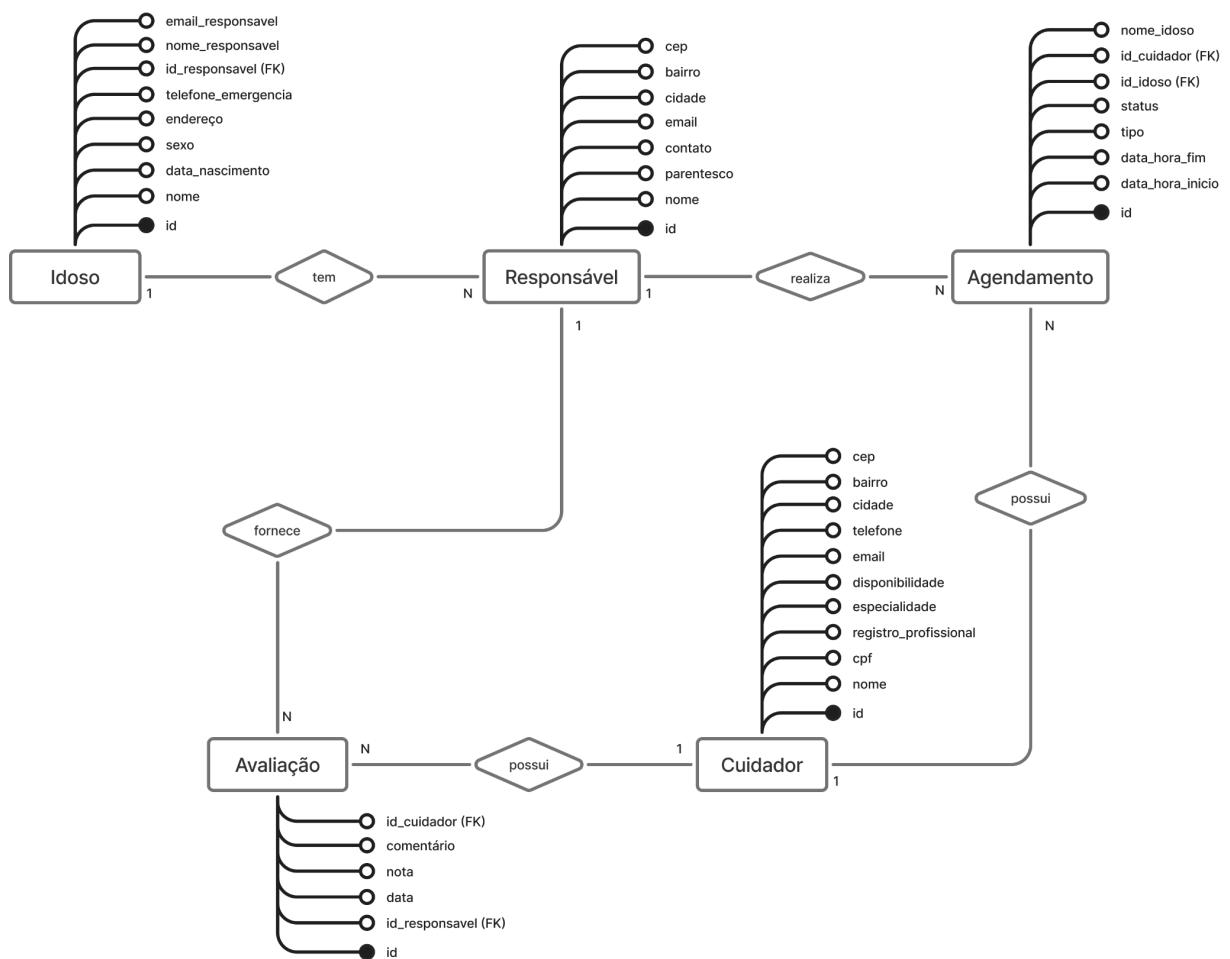
Com esse sistema, quero centralizar todas as informações em um só lugar, dados dos idosos, registros de atendimento e escalas dos cuidadores, de forma prática e acessível. A proposta é permitir agendamentos bem definidos, com histórico de quem realizou o cuidado, em que data e horário, trazendo mais transparência. Além disso, o sistema contará com um canal de feedback, para que cuidadores possam ser avaliados e os serviços aprimorados continuamente

2. REQUISITOS DO SISTEMA

- O sistema deve permitir cadastrar usuários com diferentes perfis: cuidador ou familiar.
- O sistema deve permitir cadastrar idosos com dados pessoais e contato de emergência.
- O sistema deve permitir cadastrar cuidadores, incluindo especialidade, disponibilidade e informações de contato.
- O sistema deve permitir registrar agendamentos de visitas, vinculando cuidador, idoso, tipo de atividade, data e horário.
- O sistema deve permitir editar e cancelar agendamentos existentes.
- O sistema deve permitir consultar o histórico de atendimentos realizados para cada idoso.
- O sistema deve permitir que familiares visualizem os agendamentos e visitas realizadas para o idoso sob sua responsabilidade.
- O sistema deve permitir o envio e visualização de feedbacks sobre os cuidadores, contendo nota e comentário.
- O sistema deve permitir filtrar agendamentos por data, cuidador ou idoso.
- O sistema deve gerar relatórios de visitas agendadas e realizadas, com filtros por período, status e profissional.

3. DIAGRAMA ENTIDADE-RELACIONAMENTO (MER)

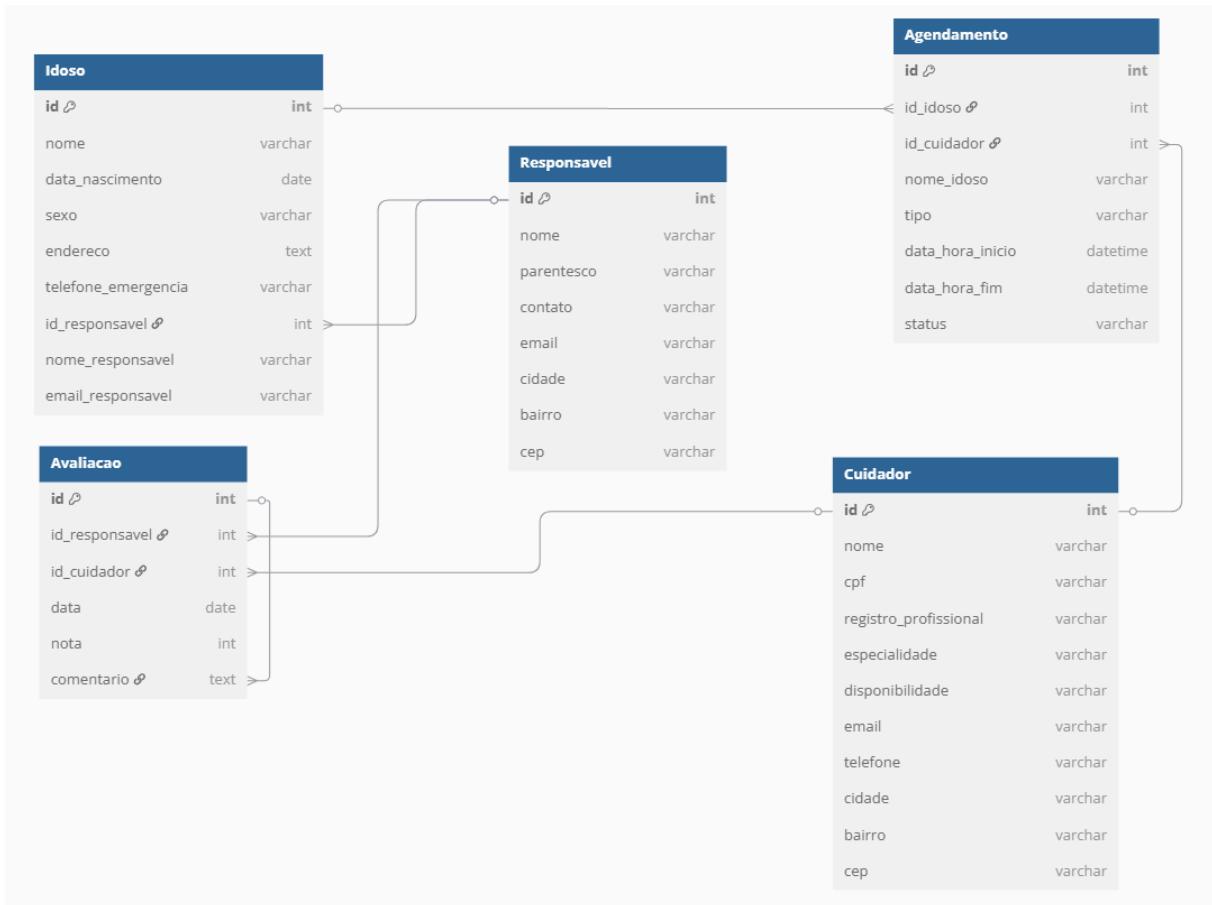
O MER deste projeto representa a estrutura de um sistema de gestão de cuidados domiciliares para idosos, com seis entidades principais: Idoso, responsável, cuidador, agendamento, avaliação e seus respectivos relacionamentos. Cada idoso é vinculado a um responsável, que realiza agendamentos de atendimentos executados por cuidadores. Após os atendimentos, o responsável pode registrar uma avaliação do cuidador.



4. DEPENDÊNCIAS FUNCIONAIS

4.2 ESQUEMATIZAÇÃO DO MER EM TABELAS

Abaixo está o diagrama MER, sem normalização, transformado em tabelas, para facilitar a visualização da estrutura do banco. Também mostro as chaves primárias, os atributos dependentes e como essas informações se conectam na álgebra relacional, ajudando a entender melhor a lógica por trás do modelo.



4.2.1. TABELA IDOSO

id	nome	data_nascimento	sexo	endereço	telefone_emergencia	id_responsavel	nome_responsavel	email_responsavel
1	Aldo Alves	19/01/1969	Masculino	Rua nascimento silva, 107, Ipanema, 22421-025 ,Rio de Janeiro, Brasil.	+55 21 99999-0000	1	Milton Jobim	jobim@jobim.com
2	Marcio Lopes	29/12/1952	Masculino	Rua São Francisco, 1500, Serraria, 57044-878, Alagoas, Maceió, Brasil	+55 82 11223-0220	2	José Silva	silva@gmail.com

Chaves primárias: id

Chaves estrangeiras: id_responsavel

Dependências funcionais:

id → nome, data_nascimento, sexo, endereço, telefone_emergencia, id_responsavel

id_responsavel → nome_responsavel, email_responsavel

Exemplo de álgebra relacional:

1. $\sigma_{\text{sexo} = \text{'Masculino'}}$ (IDOSO)
2. $\pi_{\text{endereço, telefone_emergencia}}$ (IDOSO)

Na tabela Idoso, id é a chave primária e determina todos os outros atributos diretamente. No entanto, existe uma dependência funcional transitiva: o campo id_responsavel determina nome_responsavel e email_responsavel.

4.2.2 TABELA RESPONSÁVEL

id	nome	parentesco	contato	email	cidade	bairro	cep
1	Milton Jobim	Filho	+55 21 12229-0000 +55 21 99999-0000	jobim@jobim.com	Rio de Janeiro	Ipanema	22421-025
2	José Silva	Irmão	+55 82 11223-0220	silva@gmail.com	Maceió	Jaraguá	12310-121

Chaves primárias: id

Dependências funcionais:

id → nome, parentesco, contato, email, cep

cep → bairro, cidade

Exemplo de álgebra relacional:

1. $\sigma_{\text{parentesco} = \text{'Filho'}}$ (RESPONSAVEL)
2. π_{nome} (RESPONSAVEL)

Na tabela Responsável, id é a chave primária e determina todos os outros atributos diretamente. No entanto, existe uma dependência funcional transitiva: o campo cep determina bairro e cidade.

4.2.3 TABELA CUIDADOR

id	nome	cpf	registro_profissional	especialidade	disponibilidade	email	telefone	cidade	bairro	cep
1	Maria José	999.999.999-99	12328829-12	Enfermagem Nutrição	Seg, Qua, Sex - 08h às 12h	maria@maria.com	+55 82 999987-2212	Maceió	Tabuleiro	23123-767

Chaves primárias: id

Dependências funcionais:

id → nome, cpf, registro_profissional, especialidade, disponibilidade, email, telefone, cep

cep → bairro, cidade

Exemplo de álgebra relacional:

1. $\sigma_{\text{especialidade} = \text{'Enfermagem'}}$ (CUIDADOR)
2. $\sigma_{\text{cidade} = \text{'Maceió'}}$ (CUIDADOR)

Na tabela Cuidador, o campo id é a chave primária e determina diretamente todos os demais atributos. No entanto, existe uma dependência funcional transitiva: o campo cep determina bairro e cidade.

4.2.4 TABELA AGENDAMENTO

id	data_hora_inicio	data_hora_fim	tipo	status	id_idoso	nome_idoso	id_cuidador
1	12/04/2024-12:00	12/05/2024-12:00	Acompanhamento diário	Pendente	1	Aldo Alves	1

Chaves primárias: id

Dependências funcionais:

id → data_hora_inicio, data_hora_fim, tipo, status, id_idoso, id_cuidador

id_idoso → nome_idoso

Exemplo de álgebra relacional:

1. $\sigma_{\text{status} = \text{'Concluido'}}$ (AGENDAMENTO)

Na tabela Agendamento, o atributo id é a chave primária e determina todos os demais atributos. No entanto, existe uma dependência funcional transitiva: o campo nome_idoso depende de id_idoso.

4.2.5 TABELA AVALIAÇÃO

id	data	nota	comentario	id_responsavel	id_cuidador
1	31/12/2004	5	Ótimo profissional!	1	1

Chaves primárias: id

Dependências funcionais:

id → data, nota, comentario, id_responsavel, id_cuidador

Exemplo de álgebra relacional:

1. $\sigma_{\text{nota} \geq 4}(\text{AVALIACAO})$

Na tabela Agendamento, o atributo id é a chave primária e determina todos os demais atributos.

5. NORMALIZAÇÃO

5.1 PRIMEIRA FORMA NORMAL (1FN)

A 1FN garante que cada tabela em um banco de dados tenha apenas valores atômicos em suas colunas [1].

5.1.1 TABELA IDOSO

id	nome	data_nascimento	sexo	endereço	telefone_emergencia	id_responsavel	nome_responsavel	email_responsavel
1	Aldo Alves	19/01/1969	Masculino	Rua nascimento silva, 107, Ipanema, 22421-025 ,Rio de Janeiro, Brasil.	+55 21 99999-0000	1	Milton Jobim	jobim@jobim.com
2	Marcio Lopes	29/12/1952	Masculino	Rua São Francisco, 1500, Serraria, 57044-878, Alagoas, Maceió, Brasil	+55 82 11223-0220	2	José Silva	silva@gmail.com

A tabela Idoso tem uma chave primária (id), garantindo o pilar de chave primária da 1FN. Mas o campo endereço guarda várias infos juntas (rua, número, cidade e cep), e isso quebra a regra de valor atômico por campo. Então, mesmo com chave primária, ela ainda não está 100% na 1FN.

Para resolver isso, vamos criar uma nova tabela exclusiva para armazenar as informações de endereço dos idosos.



Com essa correção, a tabela Idoso passa a ficar assim, garantindo a 1FN:

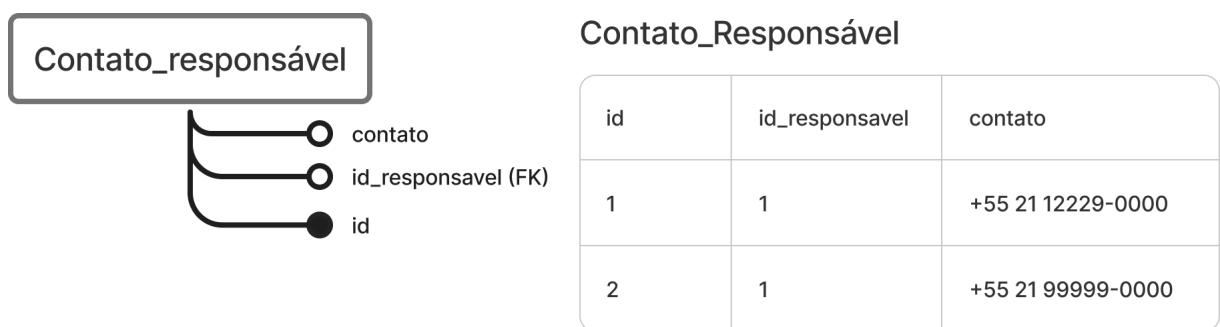
id	nome	data_nascimento	sexo	telefone_emergencia	id_responsavel	nome_responsavel	email_responsavel
1	Aldo Alves	19/01/1969	Masculino	+55 21 99999-0000	1	Milton Jobim	jobim@jobim.com
2	Marcio Lopes	29/12/1952	Masculino	+55 82 11223-0220	2	José Silva	silva@gmail.com

5.1.2 TABELA RESPONSÁVEL

id	nome	parentesco	contato	email	cidade	bairro	cep
1	Milton Jobim	Filho	+55 21 12229-0000 +55 21 99999-0000	jobim@jobim.com	Rio de Janeiro	Ipanema	22421-025
2	José Silva	Irmão	+55 82 11223-0220	silva@gmail.com	Maceió	Jaraguá	12310-121

A tabela Responsável tem uma chave primária (id), garantindo o pilar de chave primária da 1FN. Porém, quebra o princípio de valores atômicos, uma vez que o atributo contato é multivvalorado.

Para resolver isso, vamos criar uma nova tabela exclusiva para armazenar as informações dos contatos dos responsáveis.



Com essa correção, a tabela Responsável passa a ficar assim, garantindo a 1FN:

id	nome	parentesco	email	cidade	bairro	cep
1	Milton Jobim	Filho	jobim@jobim.com	Rio de Janeiro	Ipanema	22421-025
2	José Silva	Irmão	silva@gmail.com	Maceió	Jaraguá	12310-121

Observações: O atributo “email” não é considerado multivalorado, pois, em uma possível aplicação desse sistema, não será possível adicionar mais de um e-mail, já que, no cadastro do usuário responsável, esse e-mail será usado para realizar o login no sistema.

5.1.3 TABELA CUIDADOR

id	nome	cpf	registro_profissional	especialidade	disponibilidade	email	telefone	cidade	bairro	cep
1	Maria José	999.999.999-99	12328829-12	Enfermagem Nutrição	Seg, Qua, Sex - 08h às 12h	maria@maria.com	+55 82 999987-2212	Maceió	Tabuleiro	231232-767

A tabela Responsável tem uma chave primária (id), garantindo o pilar de chave primária da 1FN. No entanto, os campos especialidade, disponibilidade e telefone podem violar o princípio de valores atômicos, já que podem conter múltiplas informações em um único campo.

Para resolver isso, vamos criar uma nova tabela exclusiva para armazenar as informações dos seguintes atributos:

Especialidade_Cuidador

Especialidade_Cuidador		
id	id_cuidador	especialidade
1	1	Enfermagem
2	1	Nutrição

Especialidade_Cuidador

Especialidade_Cuidador		
id	id_cuidador	especialidade
1	1	Enfermagem
2	1	Nutrição

Disponibilidade_Cuidador

Disponibilidade_Cuidador				
id	id_cuidador	dia_semana	hora_inicio	hora_fim
1	1	Segunda	08:00	20:00
2	1	Quarta	08:00	20:00
3	1	Sexta	10:00	17:00

Disponibilidade_Cuidador

Disponibilidade_Cuidador				
id	id_cuidador	dia_semana	hora_inicio	hora_fim
1	1	Segunda	08:00	20:00
2	1	Quarta	08:00	20:00
3	1	Sexta	10:00	17:00

Contato_Cuidador

Contato_Cuidador		
id	id_cuidador	contato
1	1	+55 82 999987-2212

Contato_Cuidador

Contato_Cuidador		
id	id_cuidador	contato
1	1	+55 82 999987-2212

Com essa correção, a tabela Responsável passa a ficar assim, garantindo a 1FN:

id	nome	cpf	registro_profissional	email	cidade	bairro	cep
1	Maria José	999.999.999-99	12328829-12	maria@maria.com	Maceió	Tabuleiro	231232-767

5.1.4 TABELA AGENDAMENTO E TABELA AVALIAÇÃO

As duas tabelas estão na primeira forma normal, uma vez que possuem chaves primárias e valores atômicos.

AGENDAMENTO

id	data_hora_inicio	data_hora_fim	tipo	status	id_idoso	nome_idoso	id_cuidador
1	12/04/2024-12:00	12/05/2024-12:00	Acompanhamento diário	Pendente	1	Aldo Alves	1

AVALIAÇÃO

id	data	nota	comentario	id_responsavel	id_cuidador
1	31/12/2004	5	Ótimo profissional!	1	1

5.2 SEGUNDA FORMA NORMAL (2FN)

A 2FN garante que os atributos não-chave de uma tabela dependam totalmente da chave primária, eliminando dependências parciais [1].

5.2.1 TABELA IDOSO

id	nome	data_nascimento	sexo	telefone_emergencia	id_responsavel	nome_responsavel	email_responsavel
1	Aldo Alves	19/01/1969	Masculino	+55 21 99999-0000	1	Milton Jobim	jobim@jobim.com
2	Marcio Lopes	29/12/1952	Masculino	+55 82 11223-0220	2	José Silva	silva@gmail.com

Como a nossa tabela utiliza apenas o campo “id” como chave primária, não existem dependências parciais, apenas dependências totais e transitivas. Isso significa que todos os atributos não-chave dependem totalmente/transitivamente do “id”, e não de parte de uma chave composta. Por isso, a tabela já atende aos requisitos da Segunda Forma Normal (2FN).

Dependências Funcionais:

id → nome, data_nascimento, sexo, telefone_emergencia, id_responsavel

id_responsavel → nome_responsavel, email_responsavel

5.2.2 TABELA RESPONSÁVEL

id	nome	parentesco	email	cidade	bairro	cep
1	Milton Jobim	Filho	jobim@jobim.com	Rio de Janeiro	Ipanema	22421-025
2	José Silva	Irmão	silva@gmail.com	Maceió	Jaraguá	12310-121

Não existe dependência parcial, pois a chave primária é simples. Portanto, atende aos requisitos da 2FN.

Dependências funcionais:

id → nome, parentesco, email, cep

cep → bairro, cidade

5.2.3 TABELA CUIDADOR

id	nome	cpf	registro_profissional	email	cidade	bairro	cep
1	Maria José	999.999.999-99	12328829-12	maria@maria.com	Maceió	Tabuleiro	231232-767

Não existe dependência parcial, pois a chave primária é simples. Portanto, atende aos requisitos da 2FN.

Dependências funcionais:

id → nome, cpf, registro_profissional, especialidade, disponibilidade, email, telefone, cep

cep → bairro, cidade

5.2.4 TABELA AGENDAMENTO E AVALIAÇÃO

AGENDAMENTO

id	data_hora_inicio	data_hora_fim	tipo	status	id_idoso	nome_idoso	id_cuidador
1	12/04/2024-12:00	12/05/2024-12:00	Acompanhamento diário	Pendente	1	Aldo Alves	1

AVALIAÇÃO

id	data	nota	comentario	id_responsavel	id_cuidador
1	31/12/2004	5	Ótimo profissional!	1	1

Tanto na tabela de agendamento quanto na de avaliação, não existem dependências parciais, pois a chave primária utilizada em cada uma delas é simples. Dessa forma, ambas atendem aos requisitos da Segunda Forma Normal (2FN).

5.3 TERCEIRA FORMA NORMAL (3FN)

A 3FN visa eliminar dependências transitivas entre os atributos não chave de uma tabela, garantindo que cada atributo dependa exclusivamente da chave primária [1].

5.3.1 TABELA IDOSO

id	nome	data_nascimento	sexo	telefone_emergencia	id_responsavel	nome_responsavel	email_responsavel
1	Aldo Alves	19/01/1969	Masculino	+55 21 99999-0000	1	Milton Jobim	jobim@jobim.com
2	Marcio Lopes	29/12/1952	Masculino	+55 82 11223-0220	2	José Silva	silva@gmail.com

Nossa tabela infringe um dos requisitos da Terceira Forma Normal (3FN), pois dois atributos apresentam dependência transitiva, ou seja, não dependem diretamente da chave primária (id), mas sim do id_responsavel. Para atender à 3FN, seria necessário separar essas informações em outra tabela. No entanto, como já existe uma tabela específica para o responsável, basta remover esses atributos da tabela atual e manter as informações relacionadas apenas na tabela do responsável.

Correção:

id	nome	data_nascimento	sexo	telefone_emergencia	id_responsavel
1	Aldo Alves	19/01/1969	Masculino	+55 21 99999-0000	1
2	Marcio Lopes	29/12/1952	Masculino	+55 82 11223-0220	2

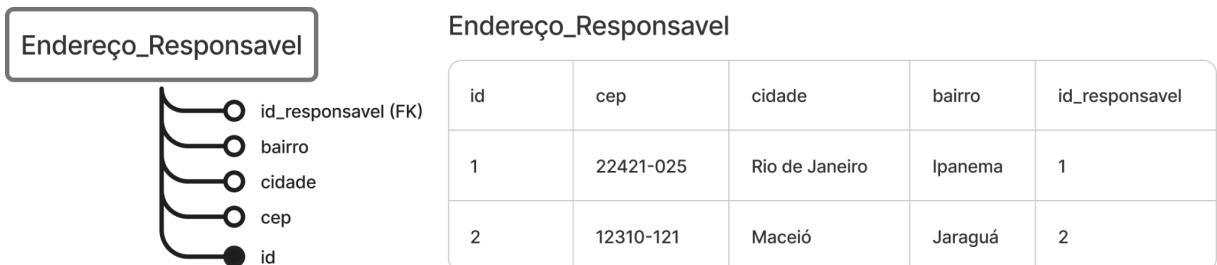
Dependências Funcionais:

id → nome, data_nascimento, sexo, telefone_emergencia, id_responsavel

5.3.2 TABELA RESPONSÁVEL

id	nome	parentesco	email	cidade	bairro	cep
1	Milton Jobim	Filho	jobim@jobim.com	Rio de Janeiro	Ipanema	22421-025
2	José Silva	Irmão	silva@gmail.com	Maceió	Jaraguá	12310-121

A tabela de responsável também não atende a 3FN, pois os atributos cidade e bairro dependem de cep e não diretamente do id - uma dependência transitiva. Nesse caso, é necessário criar uma tabela para armazenar as informações de endereço dos responsáveis.



Correção:

id	nome	parentesco	email
1	Milton Jobim	Filho	jobim@jobim.com
2	José Silva	Irmão	silva@gmail.com

Dependências funcionais:

id → nome, parentesco, email

5.3.3 TABELA CUIDADOR

id	nome	cpf	registro_profissional	email	cidade	bairro	cep
1	Maria José	999.999.999-99	12328829-12	maria@maria.com	Maceió	Tabuleiro	231232-767

A tabela de cuidador também apresenta o mesmo problema da tabela de responsável, incluindo atributos com dependência transitiva, como bairro e cidade. Assim como foi feito para a tabela de responsável, será necessário criar uma tabela separada para armazenar o endereço do cuidador.



Correção:

id	nome	cpf	registro_profissional	email
1	Maria José	999.999.999-99	12328829-12	maria@maria.com

Dependências funcionais:

id → nome, cpf, registro_profissional, email, telefone

5.3.4 TABELA AGENDAMENTO

id	data_hora_inicio	data_hora_fim	tipo	status	id_idoso	nome_idoso	id_cuidador
1	12/04/2024-12:00	12/05/2024-12:00	Acompanhamento diário	Pendente	1	Aldo Alves	1

A tabela de agendamento possui apenas um atributo que fere as restrições da 3FN, que é o nome_idoso. Para evitar a criação de uma nova tabela, basta remover esse atributo e manter essa informação apenas na tabela de idoso. Caso seja necessário consultar o nome do idoso em um agendamento, basta realizar um join utilizando o id_idoso para obter o nome desejado.

Correção:

id	data_hora_inicio	data_hora_fim	tipo	status	id_idoso	id_cuidador
1	12/04/2024-12:00	12/05/2024-12:00	Acompanhamento diário	Pendente	1	1

Dependências funcionais:

id → data_hora_inicio, data_hora_fim, tipo, status, id_idoso, id_cuidador

5.3.4 TABELA AVALIAÇÃO

id	data	nota	comentario	id_responsavel	id_cuidador
1	31/12/2004	5	Ótimo profissional!	1	1

A tabela de avaliação não possui dependências transitivas, atendendo assim aos requisitos da 3FN.

Dependências funcionais:

id → data, nota, comentario, id_responsavel, id_cuidador

5.4 FORMA NORMAL DE BOYCE-CODD (BCNF)

A BCNF visa eliminar redundâncias e inconsistências nos dados [1].

Todas as tabelas já foram normalizadas até a Terceira Forma Normal (3FN) e atendem aos requisitos da BCNF. No entanto, como solicitado no trabalho, serão apresentados alguns cenários relativos em que a aplicação da BCNF pode ser exemplificada.

5.4.1 TABELA IDOSO

id	nome	data_nascimento	sexo	telefone_emergencia	id_responsavel
1	Aldo Alves	19/01/1969	Masculino	+55 21 99999-0000	1
2	Marcio Lopes	29/12/1952	Masculino	+55 82 11223-0220	2

Em um contexto relativo, que não será adotado nesse projeto (já que todas as tabelas já atendem à BCNF), podemos considerar o atributo `id_responsavel` como uma chave candidata, assumindo que cada responsável estaria vinculado a apenas um idoso. Apesar de esse cenário não refletir a lógica real do sistema, ele é utilizado aqui apenas para os requisitos do trabalho proposto, com o objetivo de exemplificar a aplicação da BCNF.

Dependências funcionais:

(id, id_responsavel) → nome, data_nascimento, sexo, telefone_emergencia

5.4.2 TABELA RESPONSÁVEL

id	nome	parentesco	email
1	Milton Jobim	Filho	jobim@jobim.com
2	José Silva	Irmão	silva@gmail.com

Em um contexto relativo, podemos considerar o atributo email como uma chave candidata, em um cenário onde o e-mail seja único para login. Com isso, também é possível utilizá-lo junto ao id como uma superchave composta. A seguir, serão apresentadas as dependências funcionais, considerando tanto a superchave (id, email) quanto a chave candidata email.

Dependências funcionais:

(id, email) → nome, parentesco

email → nome, parentesco

5.4.3 TABELA CUIDADOR

id	nome	cpf	registro_profissional	email
1	Maria José	999.999.999-99	12328829-12	maria@maria.com

Em um contexto relativo, podemos considerar os atributos cpf e/ou registro_profissional como chaves candidatas, já que são identificadores únicos e cada cuidador possui os seus. Dessa forma, é possível tratá-los como superchaves, seja individualmente ou em combinação com o id.

Dependências funcionais:

cpf → nome, registro_profissional, email

registro_profissional → nome, cpf, email

(id, cpf, registro_profissional) → nome, email

5.4.4 TABELA AGENDAMENTO

id	data_hora_inicio	data_hora_fim	tipo	status	id_idoso	id_cuidador
1	12/04/2024-12:00	12/05/2024-12:00	Acompanhamento diário	Pendente	1	1

Em um contexto relativo, podemos considerar data_hora_inicio e id_cuidador como chaves candidatas, levando em consideração que cada cuidador só pode realizar um atendimento por vez, ou seja, não pode ter dois agendamentos no mesmo horário.

Dependências funcionais:

$$(\text{id_cuidador}, \text{data_hora_inicio}) \rightarrow \text{tipo, status, data_hora_fim, id_idoso}$$

5.4.5 TABELA AVALIAÇÃO

id	data	nota	comentario	id_responsavel	id_cuidador
1	31/12/2004	5	Ótimo profissional!	1	1

Em um contexto relativo, podemos considerar id_responsavel, id_cuidador e data como chaves candidatas, levando em consideração que cada responsável só pode avaliar um cuidador por dia.

Dependências funcionais:

$$(\text{id_responsavel}, \text{id_cuidador}, \text{data}) \rightarrow \text{nota, comentario}$$

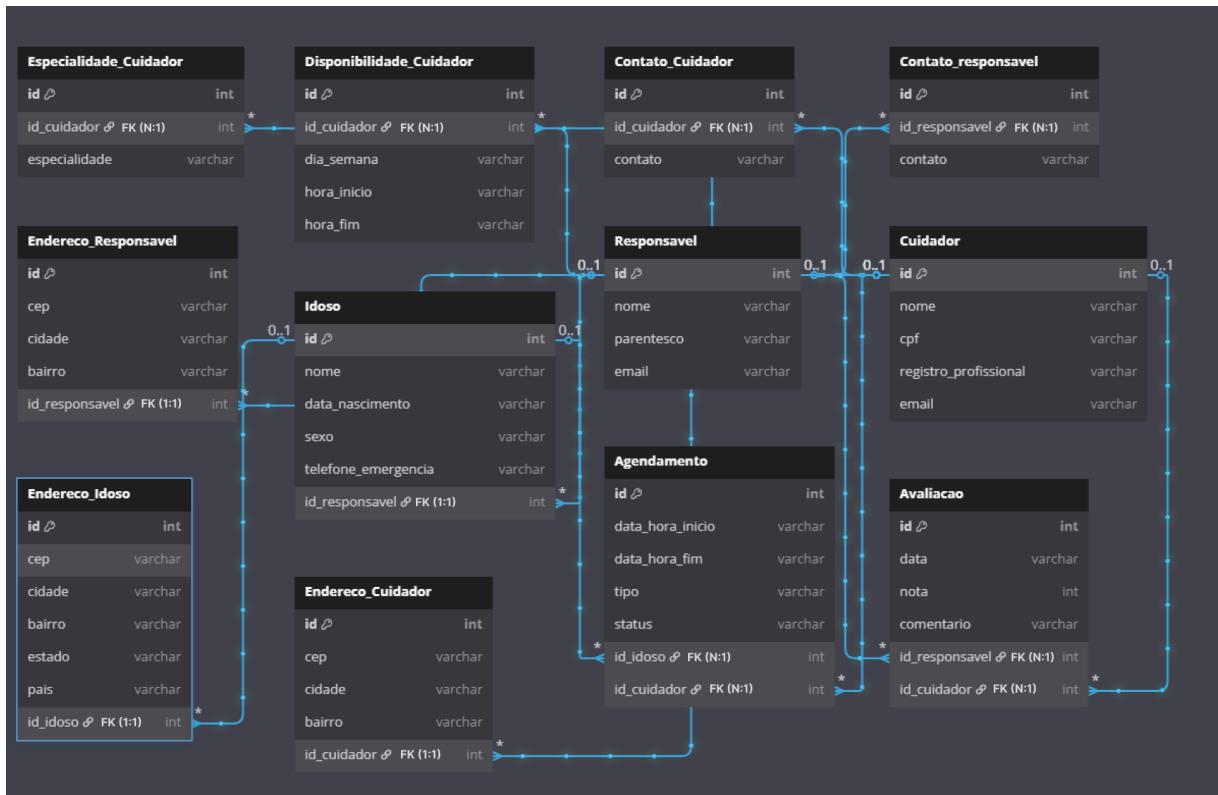
5.5 RESULTADO DA NORMALIZAÇÃO

Existem algumas tabelas repetidas, como as de endereço e contato. Isso aconteceu, pois cada uma delas está ligada a um tipo específico (idoso, cuidador ou responsável), usando um ID diferente para manter as relações corretas.

No futuro, dá pra simplificar e transformar em uma tabela única de endereços e outra de contatos, usando um campo para indicar a quem pertence aquele registro (se é do idoso, cuidador ou responsável). Assim, o modelo fica mais prático e fácil de manter caso precise expandir o sistema.

6. MODELO LÓGICO RELACIONAL

Abaixo está o modelo lógico do projeto, apresentado em formato de tabelas relacionais, feito no site dbDiagram. O modelo mostra os nomes das entidades, atributos, tipos de dados, chaves primárias, chaves estrangeiras e suas cardinalidades. Também segue o link para visualizar o modelo diretamente no site.



<https://dbdiagram.io/d/682741d81227bdcb4ea5f554>

7. CONSULTAS SQL

Após transformar nosso modelo lógico em SQL, usando o MySql, e criar todas as tabelas, vou realizar algumas operações para demonstrar as principais funcionalidades do sistema.

Todos os códigos estarão no dump e no repositório do projeto.

7.1 INSERT, UPDATE, DELETE

```
● ● ●
1  -- 1.1 INSERÇÃO
2  INSERT INTO Responsavel (nome,parentesco,email)
3    VALUES ('João Marques', 'Filho', 'joao@joao.com');
4
5  INSERT INTO Contato_Responsavel (telefone,id_responsavel)
6    VALUES ('82999993231', 1);
7
8  INSERT INTO Cuidador (nome,cpf,registro_profissional,email)
9    VALUES ('Matheus Cabral', '000.000.000-00', '1231422-12', 'matheus@gmail.com');
10
11 INSERT INTO Endereco_Cuidador (cep,cidade,bairro,id_cuidador)
12   VALUES ('19864-821', 'Murici', 'Ouro Branco', 1);
13
14 INSERT INTO Especialidade_Cuidador (especialidade,id_cuidador)
15   VALUES ('Enfermagem Dermatológica', 1);
16
17 INSERT INTO Disponibilidade_Cuidador (dia_semana,hora_inicio,hora_fim,id_cuidador)
18   VALUES ('Segunda-feira','07:30:00','20:30:00', 1);
19
20 INSERT INTO Disponibilidade_Cuidador (dia_semana,hora_inicio,hora_fim,id_cuidador)
21   VALUES ('Quarta-feira','07:30:00','20:30:00', 1);
22
23 INSERT INTO Idoso (nome,data_nASCimento,sexo,telefone_emergencia,id_responsavel)
24   VALUES ('Aldo Alvez', '1969-12-02', 'MASCULINO' , '82998213221', 1);
25
26 INSERT INTO Agendamento (data_hora_inicio,data_hora_fim,tipo,status_consulta,id_cuidador,id_idoso)
27   VALUES ('2025-04-28 09:00:00','2025-04-29 09:00:00','Diária Completa','Concluido',1,5);
28
29 INSERT INTO Avaliacao (data,nota,comentario,id_cuidador,id_responsavel)
30   VALUES ('2025-04-18 09:00:00', 5, 'Ótima profissional' , 1, 3);
31
32 -- 1.2 REMOÇÃO
33 DELETE FROM Agendamento WHERE id = 4;
34 DELETE FROM Endereco_Idoso WHERE id = 4;
35 DELETE FROM Idoso WHERE id = 6;
36
37 -- 1.3 ALTERAÇÃO
38 UPDATE Cuidador SET email = 'elina@elianam.com' WHERE id = 3;
39 UPDATE Responsavel SET parentesco = 'Sobrinha' WHERE id = 7;
```

7.2 SELECT E JOIN

```
● ● ●
1 -- 1. Listar todas as informações dos idosos
2 SELECT * FROM idoso;
3
4 -- 2. Listar todos os idosos e seus responsáveis
5 SELECT
6   idoso.nome AS nome_idoso
7 , responsavel.nome AS nome_responsavel,
8 responsavel.parentesco AS parentesco_responsavel
9 FROM idoso
10 INNER JOIN responsavel ON idoso.id_responsavel = responsavel.id;
11
12 -- 3. Listar todos os cuidadores e seus contatos
13 SELECT
14   cuidador.nome AS nome_cuidador,
15 contato_cuidador.telefone AS contato_cuidador
16 FROM cuidador
17 LEFT JOIN contato_cuidador ON cuidador.id = contato_cuidador.id_cuidador;
18
19 -- 4. Listar apenas os agendamentos que já foram concluídos
20 SELECT
21   agendamento.data_hora_fim AS horario_finalizacao_consulta,
22   idoso.nome AS nome_idoso, cuidador.nome AS nome_cuidador
23 FROM agendamento
24   INNER JOIN idoso ON agendamento.id_idoso = idoso.id
25   INNER JOIN cuidador ON agendamento.id_cuidador = cuidador.id
26 WHERE agendamento.status_consulta = 'Concluido';
27
28 -- 5. Listar cuidadores e suas especializações
29 SELECT
30   cuidador.nome AS nome_cuidador,
31 especialidade_cuidador.especialidade AS especializacao
32 FROM cuidador
33   INNER JOIN especialidade_cuidador ON cuidador.id = especialidade_cuidador.id_cuidador;
```

```
● ● ●
1 -- 6. Listar todos os idosos e agendamentos
2 SELECT
3   idoso.nome AS nome_idoso,
4   agendamento.data_hora_inicio, agendamento.status_consulta
5 FROM idoso
6   LEFT JOIN agendamento ON idoso.id = agendamento.id_idoso;
7
8 -- 7 Listar responsáveis e endereços
9 SELECT
10   responsavel.nome AS nome_responsavel,
11   endereco_responsavel.cidade, endereco_responsavel.bairro
12 FROM endereco_responsavel
13   RIGHT JOIN responsavel ON endereco_responsavel.id_responsavel = responsavel.id;
```

```

1 -- 8 Listar cuidadores e endereços
2 SELECT
3     responsavel.nome AS nome_responsavel,
4     endereco_responsavel.cidade, endereco_responsavel.bairro
5     FROM endereco_responsavel
6         RIGHT JOIN responsavel ON endereco_responsavel.id_responsavel = responsavel.id;
7
8 -- 9 Listar avaliações que possuem nota acima de 3.0
9 SELECT *
10    FROM avaliacao
11   WHERE nota >= 3;
12
13 -- 10. Listar todas as avaliações e as informações dos idoso e responsaveis
14 SELECT
15     avaliacao.id, avaliacao.nota,
16     cuidador.nome AS nome_cuidador,
17     cuidador.registro_profissional AS registro_cuidador,
18     responsavel.nome AS nome_responsavel
19     FROM avaliacao
20        INNER JOIN cuidador ON avaliacao.id_cuidador = cuidador.id
21        INNER JOIN responsavel ON avaliacao.id_responsavel = responsavel.id;

```

7.3 AGREGAÇÕES

```

1 -- 1. Total de cuidadores cadastrados
2     SELECT COUNT(*) AS total_cuidadores FROM cuidador;
3 -- 2. Total de idosos cadastrados
4     SELECT COUNT(*) AS total_idosos FROM idoso;
5 -- 3. Média das notas de avaliações
6     SELECT AVG(nota) AS media_nota FROM avaliacao;
7 -- 4. Média das idades dos idosos
8     SELECT AVG(YEAR(CURDATE()) - YEAR(data_nascimento)) AS media_idade FROM idoso;
9 -- 5. Total de agendamentos pendentes
10    SELECT COUNT(*) AS total_pendentes FROM agendamento WHERE status_consulta = 'Pendente';
11 -- 6. Maior e menor nota das avaliações
12    SELECT MAX(nota) AS nota_maxima, MIN(nota) AS nota_minima FROM avaliacao;
13 -- 7. Total de avaliações
14    SELECT COUNT(*) AS total_avaliacoes FROM avaliacao;
15 -- 8. Total de contatos de responsáveis cadastrados
16    SELECT COUNT(*) FROM contato_responsavel;
17 -- 9. Total de responsáveis
18    SELECT COUNT(*) FROM responsavel;
19 -- 10. total de endereços de idosos cadastrados
20    SELECT COUNT(*) FROM endereco_idoso;

```

7.4 ORDENAÇÕES

```
● ● ●
1 -- 1. Quantidade de avaliações por cuidador
2 SELECT id_cuidador, count(*) as total_avaliacoes
3   FROM avaliacao
4     GROUP BY id_cuidador;
5 -- 2. Média de notas por cuidador
6 SELECT id_cuidador, avg(notas) as media_nota
7   FROM avaliacao
8     GROUP BY id_cuidador;
9 -- 3. Quantidade de agendamentos por status --> ordenando do maior para o menor
10 SELECT status_consulta, count(*) as total_agendamentos
11   FROM agendamento
12     GROUP BY status_consulta
13       ORDER BY total_agendamentos DESC;
14 -- 4. Cuidadores ordenados por nome
15 SELECT * FROM cuidador
16   ORDER BY nome ASC;
17 -- 5. Idosos ordenados por idade
18 SELECT * FROM idoso
19   ORDER BY data_nASCimento ASC;
20 -- 6. Cuidadores que possuem mais de uma especialização
21 SELECT id_cuidador, count(*) as total_especialidades
22   FROM especialidade_cuidador
23     GROUP BY id_cuidador
24       HAVING count(*) > 1;
25 -- 7. 5 idosos mais novos
26 SELECT * FROM idoso
27   ORDER BY data_nASCimento DESC
28   LIMIT 5;
29 -- 8. Média das notas ordenadas
30 SELECT id_cuidador, avg(notas) as media_nota
31   FROM avaliacao
32     GROUP BY id_cuidador
33       ORDER BY media_nota DESC;
34 -- 9. Total de contatos cadastrados dos cuidadores ordenados
35 SELECT id_cuidador, count(*) as total_contatos
36   FROM contato_cuidador
37     GROUP BY id_cuidador
38       ORDER BY total_contatos DESC;
39 -- 10. Agendamentos ordenados por data de inicio
40 SELECT * FROM agendamento
41   ORDER BY data_hora_inicio ASC;
```

7.5 SUBCONSULTAS

```
● ● ●

1 -- 1. Listar cuidadores que possuem avaliações
2 SELECT * FROM cuidador
3   WHERE id IN (SELECT id_cuidador FROM avaliacao);
4
5 -- 2. Listar responsáveis que possuem mais de um idoso
6 SELECT * FROM responsavel
7   WHERE id IN (
8     SELECT id_responsavel FROM idoso
9     GROUP BY id_responsavel
10    HAVING count(*) > 1
11  );
12
13 -- 3. Listar idosos que não têm agendamento
14 SELECT * FROM idoso i
15   WHERE NOT EXISTS (
16     SELECT 1 FROM agendamento a
17     WHERE a.id_idoso = i.id
18  );
19
20 -- 4. Listar cuidadores que nunca receberam avaliação
21 SELECT * FROM cuidador
22   WHERE id NOT IN (SELECT id_cuidador FROM avaliacao);
23
24 -- 5. Listar agendamentos que receberam nota 5
25 SELECT * FROM agendamento
26   WHERE id_idoso IN (
27     SELECT id_idoso FROM avaliacao
28     WHERE nota = (SELECT max(nota) FROM avaliacao)
29  );
30
31 -- 6. Listar responsáveis que já avaliaram algum cuidador
32 SELECT distinct r.*
33   FROM responsavel r
34   WHERE EXISTS (
35     SELECT 1 FROM avaliacao a
36     WHERE a.id_responsavel = r.id AND a.id_cuidador = 1
37  );
38
39 -- 7. Listar cuidadores que tem mais de uma especialidade
40 SELECT * FROM cuidador
41   WHERE id IN (
42     SELECT id_cuidador FROM especialidade_cuidador
43     GROUP BY id_cuidador
44    HAVING count(*) > 1
45  );
```

```
● ● ●
1 -- 8. Listar todos os idosos que moram nos bairros dos cuidadores
2 SELECT * FROM idoso
3   WHERE id IN (
4     SELECT e.id_idoso FROM endereco_idoso e
5       WHERE e.bairro IN (
6         SELECT ec.bairro FROM endereco_cuidador ec
7       )
8   );
9
10 -- 9. Listar cuidadores que tem mais de um agendamento concluído
11 SELECT * FROM cuidador c
12   WHERE EXISTS (
13     SELECT 1 FROM agendamento a
14       WHERE a.id_cuidador = c.id AND a.status_consulta = 'concluido'
15   );
16
17 -- 10. Listar responsáveis que nunca cadastraram contato
18 SELECT * FROM responsavel r
19   WHERE NOT EXISTS (
20     SELECT 1 FROM contato_responsavel cr
21       WHERE cr.id_responsavel = r.id
22   );
```

8. CONCLUSÃO

Segue junto ao PDF o arquivo SQL com todas as operações demonstradas, além do dump completo do banco de dados. Chegando ao final desse projeto, deixo aqui algumas conclusões e observações.

O sistema, apesar de básico e ainda não preparado para grande escalabilidade, está bem estruturado, documentado e totalmente normalizado. Meu principal objetivo foi criar uma base sólida, que possa servir de ponto de partida para projetos futuros, onde pretendo aperfeiçoar os conceitos, implementar uma interface e construir um backend robusto. Quem sabe, no futuro, transformar essa ideia em uma aplicação completa!

Quero agradecer à professora pelos ensinamentos que me deram base e confiança para chegar até aqui, ao monitor pelo apoio e pelas correções ao longo do desenvolvimento, e também a todos os colegas que estiveram juntos comigo durante esse período na disciplina de banco de dados.

Repositório: <https://github.com/VictorBrasileiroo/Projeto-BD-AB2>

9. REFERÊNCIAS

- [1] **MICROSOFT.** Descrição da normalização do banco de dados: processo de normalização e suas formas normais. Disponível em:
<https://learn.microsoft.com/pt-br/office/troubleshoot/access/database-normalization-description>. Acesso em: 18 maio 2025.
- [2] **NBR 6023.** *Informação e documentação – Referências – Elaboração.* Rio de Janeiro: ABNT, 2018.
- [3] **HEUSER,** Carlos Alberto. *Projeto de Banco de Dados.* 6. ed. Porto Alegre: Bookman, 2009.