



Logo da Casa da Criança Batuíra



Logo da Faculdade Processus

Modelagem do Banco de Dados da Casa da Criança Batuíra

Brasília 2026

Resumo

O presente trabalho acadêmico apresenta a modelagem do banco de dados que dará suporte ao sistema de atendimento ao público das instituições Casa da Criança Baturia e Casa da Mãe Baturia, Organizações da Sociedade Civil (OSC) que promovem acolhimento a crianças, adolescentes e famílias em situação de vulnerabilidade. O sistema centraliza, em um único número de WhatsApp e em um painel web, o atendimento das duas organizações, roteando cada contato para o setor e atendente adequados. O trabalho concentra-se exclusivamente na fase de modelagem de dados, contemplando o Modelo Conceitual e o Modelo Lógico. Apresenta-se a fundamentação teórica do Modelo Entidade-Relacionamento (MER), a construção do Diagrama Entidade-Relacionamento (DER) em notação Chen, o Modelo Lógico Relacional resultante do mapeamento, a análise das dependências funcionais, a verificação das formas normais (1FN, 2FN e 3FN) e a definição das restrições de integridade. O modelo resultante é composto por nove entidades organizadas em quatro domínios funcionais e foi projetado para ser implementado em qualquer Sistema Gerenciador de Banco de Dados (SGBD) relacional.

Palavras-chave: *Modelagem de Dados; Modelo Conceitual; Modelo Lógico; Modelo Entidade-Relacionamento; Diagrama Entidade-Relacionamento; Normalização.*

Alunos:

Cleude Rodrigues Machado / (61) 99992-9820

Luís Felipe da Silva Barreto / (61) 98491-2580

Matheus Barbosa Trindade / (61) 99968-6170

Maria Eduarda Saraiva Cunha / (61) 98194-8569

Coordenadores:

Henderson Matsuura Sanches

Responsável da Casa da Criança Baturia:

Giovanna Queiroz, recebido no dia 26 de Março de 2026

Apresentação da Disciplina

Este trabalho acadêmico é o trabalho final para aprovação na disciplina Atividade de Extensão do curso Sistemas de Informação / Análise e Desenvolvimento de Sistemas da Faculdade Processus, sob orientação dos professores coordenadores. A disciplina possibilita o contato e a interferência direta do corpo acadêmico para além dos muros da faculdade, prestando consultoria tecnológica a uma organização da sociedade civil. O presente trabalho contempla a etapa de modelagem de dados, fase fundamental do ciclo de desenvolvimento de sistemas de informação, que precede e fundamenta qualquer implementação subsequente.

Cronograma de Atividades

Atividade	Data/Período
Reunião inicial e levantamento de requisitos de dados	17 de Março de 2026
Estudo da fundamentação teórica (MER, modelo relacional, normalização)	18 de Março de 2026
Identificação de entidades, atributos e chaves	20 de Março de 2026
Definição de relacionamentos e cardinalidades	20 de Março de 2026
Elaboração do Diagrama Entidade-Relacionamento (DER)	22 de Março de 2026
Conversão do MER para o modelo lógico relacional	22 de Março de 2026
Elaboração do dicionário de dados	23 de Março de 2026
Análise de dependências funcionais	25 de Março de 2026
Verificação das formas normais (1FN, 2FN, 3FN)	07 de Abril de 2026
Modelagem do ciclo de vida das conversas	14 de Abril de 2026
Redação do trabalho acadêmico	16 de Maio de 2026
Revisão e entrega do trabalho final	25 de Junho de 2026

Obs.: substituir as datas acima pelas datas reais do projeto.

Sumário

1. Introdução	5
2. Objetivo	5
2.1 Objetivo Geral	5
2.2 Objetivos Específicos	6
3. Fundamentação Teórica	6
3.1 Níveis de Abstração na Modelagem de Dados	6
3.2 Modelo Entidade-Relacionamento (MER)	6
3.3 Diagrama Entidade-Relacionamento (DER)	7
3.4 Modelo Lógico Relacional	7
3.5 Cardinalidades	8
3.6 Restrições de Integridade	8
3.7 Dependências Funcionais	9
3.8 Normalização	9
4. Modelagem Conceitual	9
4.1 Levantamento de Requisitos de Dados	9
4.2 Entidades Identificadas	10
4.3 Atributos das Entidades	10
4.4 Relacionamentos e Cardinalidades	11
4.5 Diagrama Entidade-Relacionamento (DER)	12
5. Modelagem Lógica	12
5.1 Mapeamento do MER para o Modelo Relacional	12
5.2 Esquema Relacional	13
5.3 Dicionário de Dados	14
5.4 Restrições de Integridade do Modelo	17
5.5 Análise de Dependências Funcionais	18
5.6 Verificação das Formas Normais	18
6. Modelagem do Ciclo de Vida das Conversas	19
7. Considerações sobre o SGBD-alvo	20

8. Considerações Finais	20
Referências	21

1. Introdução

O atendimento ao público de organizações do terceiro setor frequentemente ocorre por meio de aplicativos de mensagens, com destaque para o WhatsApp. Quando duas instituições — a Casa da Criança Batuíra e a Casa da Mãe Batuíra — compartilham o mesmo canal de contato, surge a necessidade de um sistema capaz de identificar a instituição desejada, direcionar a conversa ao setor competente e registrar todo o histórico de interações de forma persistente e auditável.

As Organizações da Sociedade Civil (OSCs) têm histórica contribuição para as sociedades modernas, atuando em áreas como saúde, serviço social, educação e assistência. No Brasil, são mais de 780 mil OSCs, sendo quase 40 mil voltadas ao serviço social (IPEA, 2023). É nesse contexto que a Casa da Criança Batuíra se insere, promovendo acolhimento, desenvolvimento e apoio a crianças, adolescentes e famílias em situação de vulnerabilidade na região do Distrito Federal.

A modelagem de dados é a etapa fundamental para que esse sistema funcione de maneira consistente. É nela que se definem as entidades do domínio, suas propriedades, os relacionamentos entre elas e as regras que garantem a integridade das informações. Uma modelagem bem feita antecipa problemas, evita redundâncias, assegura a coerência dos dados e estabelece um esquema robusto e estável, independente de qualquer escolha tecnológica posterior. Em contrapartida, falhas de modelagem propagam-se por toda a aplicação e tornam-se onerosas de corrigir após a implementação.

Este trabalho acadêmico apresenta a modelagem completa do banco de dados que dará suporte ao sistema de atendimento das instituições Batuíra, percorrendo o Modelo Conceitual e o Modelo Lógico. Adota-se a notação Chen para o Diagrama Entidade-Relacionamento e o modelo relacional clássico, formalizado por Edgar F. Codd, como base para o modelo lógico.

2. Objetivo

O objetivo versa sobre qual horizonte se pretende alcançar com este trabalho acadêmico, definindo tanto o escopo geral quanto as etapas específicas para obter o resultado final.

2.1 Objetivo Geral

Este trabalho acadêmico busca elaborar e documentar a modelagem completa do banco de dados de apoio ao sistema de atendimento via WhatsApp da Casa da Criança Batuíra e da Casa da Mãe Batuíra, contemplando o Modelo Conceitual e o Modelo Lógico, fundamentados na teoria do Modelo Entidade-Relacionamento e do modelo relacional, e atendendo aos princípios de integridade e normalização.

2.2 Objetivos Específicos

- Apresentar a fundamentação teórica da modelagem de dados relacional;
- Construir o Modelo Entidade-Relacionamento (MER) do domínio, identificando entidades, atributos, chaves e relacionamentos;
- Elaborar o Diagrama Entidade-Relacionamento (DER) correspondente, em notação Chen;
- Mapear o Modelo Conceitual para o Modelo Lógico relacional;
- Documentar o dicionário de dados completo;
- Analisar as dependências funcionais identificadas;
- Verificar a conformidade do modelo com as formas normais (1FN, 2FN e 3FN);
- Especificar as restrições de integridade que asseguram a consistência dos dados.

3. Fundamentação Teórica

A modelagem de dados é o processo de criação de uma representação abstrata da realidade que será armazenada em um banco de dados. Ela é tradicionalmente conduzida em níveis de abstração distintos, sendo o conceitual e o lógico os abordados neste trabalho acadêmico. Esta seção apresenta os conceitos teóricos que fundamentam essa atividade.

3.1 Níveis de Abstração na Modelagem de Dados

Tradicionalmente, a modelagem de dados é organizada em três níveis principais de abstração:

- **Conceitual:** descreve, em alto nível, o que existe no domínio. Representa entidades, atributos e relacionamentos do mundo real, sem qualquer

compromisso com um SGBD específico. É a visão compartilhada com clientes e analistas. Ferramenta típica: o MER, representado graficamente pelo DER.

- **Lógico:** traduz o Modelo Conceitual em estruturas formais do modelo de dados adotado (no caso deste trabalho, o modelo relacional). Define tabelas, colunas, tipos, chaves primárias e estrangeiras, mas ainda independe do SGBD específico.
- **Físico:** trata da forma como os dados são efetivamente armazenados em meio físico (índices, organização de arquivos, blocos, etc.). Como o foco deste trabalho é a modelagem nos níveis conceitual e lógico, o nível físico é apenas brevemente referenciado na Seção 7.

3.2 Modelo Entidade-Relacionamento (MER)

O Modelo Entidade-Relacionamento, proposto por Peter Chen em 1976, é a abordagem mais difundida para a modelagem conceitual de dados. Sua abstração central é a entidade — um objeto do mundo real (concreto ou abstrato) com existência independente e sobre o qual se deseja armazenar informações. Cada entidade possui atributos, que são suas propriedades descritivas, e relaciona-se com outras entidades.

Os elementos fundamentais do MER são:

- **Entidade:** objeto distinguível sobre o qual se armazenam informações (ex.: Setor, Atendente, Conversa);
- **Atributo:** propriedade de uma entidade (ex.: nome, telefone, data de criação). Atributos podem ser simples ou compostos, monovalorados ou multivalorados, armazenados ou derivados;
- **Relacionamento:** associação significativa entre duas ou mais entidades (ex.: um Atendente pertence a um Setor);
- **Chave primária:** atributo (ou conjunto de atributos) que identifica unicamente cada ocorrência de uma entidade.

O MER é independente de qualquer SGBD ou tecnologia. Seu propósito é descrever o domínio do problema de forma compreensível por analistas, clientes e desenvolvedores.

3.3 Diagrama Entidade-Relacionamento (DER)

O Diagrama Entidade-Relacionamento é a representação gráfica do MER. Na notação original de Chen — adotada neste trabalho acadêmico — utilizam-se os seguintes símbolos:

- **Retângulo:** representa uma entidade;
- **Elipse:** representa um atributo;
- **Losango:** representa um relacionamento;
- **Linhas:** conectam entidades a relacionamentos e atributos às suas entidades;
- **Elipse com texto sublinhado:** indica o atributo identificador (chave primária);
- **Cardinalidades:** rotuladas sobre as linhas que ligam entidades aos losangos (1:1, 1:N ou N:M).

Existem notações alternativas — como Pé-de-Galinha (Crow's Foot) e UML — mas, para fins de modelagem conceitual e didática, a notação Chen é a mais explícita e amplamente utilizada em ambientes acadêmicos.

3.4 Modelo Lógico Relacional

O Modelo Lógico é a etapa subsequente à modelagem conceitual. Ele transforma as entidades e relacionamentos do MER em relações (tabelas, na terminologia comum), seguindo o modelo relacional formalizado por Edgar F. Codd em 1970.

Cada relação possui:

- Um conjunto de atributos (colunas), cada um com um domínio (tipo de dado);
- Tuplas (linhas) representando ocorrências da entidade;
- Uma chave primária que identifica unicamente cada tupla;
- Possíveis chaves estrangeiras que referenciam chaves primárias de outras relações, materializando os relacionamentos.

O Modelo Lógico permanece independente do SGBD específico: o mesmo esquema pode ser implementado em SQLite, PostgreSQL, MySQL, Oracle, entre outros, sem alterações estruturais.

3.5 Cardinalidades

A cardinalidade de um relacionamento expressa quantas ocorrências de uma entidade podem estar associadas a quantas ocorrências de outra. As três cardinalidades clássicas são:

- **1:1 (um para um):** cada ocorrência de A se relaciona com no máximo uma de B, e vice-versa. Exemplo: cada pessoa possui um único CPF, e cada CPF pertence a uma única pessoa.
- **1:N (um para muitos):** uma ocorrência de A pode se relacionar com muitas de B, mas cada B se relaciona com no máximo uma A. Exemplo: um setor possui vários atendentes; cada atendente pertence a um único setor.
- **N:M (muitos para muitos):** uma ocorrência de A pode se relacionar com muitas de B, e vice-versa. Exemplo: alunos matriculam-se em várias disciplinas, e cada disciplina tem vários alunos.

Além da cardinalidade máxima (1 ou N), define-se também a cardinalidade mínima, que indica se a participação da entidade no relacionamento é obrigatória (mínimo 1) ou opcional (mínimo 0). A notação completa expressa o par (mínimo, máximo) — por exemplo, (1,N) ou (0,1).

3.6 Restrições de Integridade

As restrições de integridade são regras que asseguram que apenas dados consistentes sejam armazenados no banco. As principais categorias são:

- **Integridade de entidade:** toda relação deve ter uma chave primária declarada e nenhum atributo dessa chave pode assumir valor nulo;
- **Integridade referencial:** uma chave estrangeira deve referenciar um valor existente na chave primária da tabela referenciada (ou ser nula, se a coluna assim o permitir);
- **Integridade de domínio:** cada atributo deve assumir valores compatíveis com seu domínio (tipo, faixa, formato, conjunto de valores permitidos);
- **Integridade definida pelo usuário:** regras de negócio específicas do domínio, expressas por meio de restrições CHECK, asserções ou gatilhos.

3.7 Dependências Funcionais

Uma dependência funcional $X \rightarrow Y$ indica que, dado o valor dos atributos em X, o valor dos atributos em Y é unicamente determinado. Em outras palavras, em

todas as tuplas de uma relação em que os valores de X sejam iguais, os valores de Y também o serão.

Exemplo: na relação `Atendente(id, nome, sector_id, email)`, tem-se $id \rightarrow nome$, $id \rightarrow email$, $id \rightarrow sector_id$, pois conhecendo-se o id do atendente conhecem-se todos os seus demais atributos. A análise das dependências funcionais é a base do processo de normalização.

3.8 Normalização

A normalização é um processo sistemático que decompõe relações com problemas estruturais (redundância, anomalias de inserção, atualização e remoção) em relações menores, bem formadas. As três primeiras formas normais, em ordem crescente de rigor, são as mais aplicadas em projetos práticos:

- **1ª Forma Normal (1FN):** todos os atributos da relação contêm valores atômicos (não multivalorados e não compostos), e não há grupos repetidos;
- **2ª Forma Normal (2FN):** a relação está em 1FN e todo atributo não-chave depende funcionalmente da chave primária inteira (eliminação de dependências parciais; restrição relevante apenas quando há chave primária composta);
- **3ª Forma Normal (3FN):** a relação está em 2FN e nenhum atributo não-chave depende transitivamente da chave primária (eliminação de dependências transitivas entre atributos não-chave).

Um modelo normalizado até a 3FN é, em geral, considerado adequado para aplicações relacionais convencionais, equilibrando integridade, eliminação de redundâncias e desempenho razoável de consultas.

4. Modelagem Conceitual

Esta seção apresenta o Modelo Entidade-Relacionamento (MER) construído a partir do levantamento de requisitos junto às instituições Baturia, culminando no Diagrama Entidade-Relacionamento (DER) em notação Chen.

4.1 Levantamento de Requisitos de Dados

Da análise do domínio e das conversas mantidas com a equipe das instituições, identificaram-se as seguintes necessidades de armazenamento de informação:

- Registrar os setores de atendimento de cada uma das duas instituições, com nome, descrição, ordem de exibição no menu e indicador de atividade;
- Cadastrar os atendentes vinculados a cada setor, com seus dados de identificação e perfil;
- Manter o estado e o histórico de cada conversa atendida, identificando o contato, a instituição, o setor e o atendente envolvidos;
- Persistir todas as mensagens trocadas em cada conversa, distinguindo o sentido (recebida do contato ou enviada pela instituição);
- Rastrear as transferências de conversas entre setores, preservando setor de origem, setor de destino e atendente responsável pelo encaminhamento;
- Armazenar informações complementares sobre cada contato, como apelido e anotações livres mantidos pelos atendentes;
- Manter um catálogo de respostas rápidas (mensagens-modelo) reutilizáveis pelos atendentes;
- Gerenciar chaves de acesso para integração programática com a API do sistema;
- Persistir fluxos conversacionais configuráveis utilizados pelo bot.

4.2 Entidades Identificadas

A partir do levantamento, foram identificadas nove entidades, agrupadas em quatro domínios funcionais coerentes, conforme o Quadro 1.

Entidade	Domínio	Finalidade
Setor	Organizacional	Setores de atendimento de cada instituição.
Atendente	Organizacional	Atendentes vinculados a um setor.
Conversa	Atendimento	Estado e metadados de cada atendimento.
Mensagem	Atendimento	Mensagens trocadas em cada conversa.
Transferência	Atendimento	Histórico de transferências entre setores.
Contato	Apoio	Apelido e anotações sobre cada número de contato.
RespostaRápida	Apoio	Mensagens-modelo reutilizáveis pelos atendentes.
ChaveAPI	Integração	Chaves de acesso programático à API.
Fluxo	Integração	Fluxos conversacionais do bot.

Quadro 1 — Entidades do modelo e seus domínios funcionais. Fonte: Próprio autor.

4.3 Atributos das Entidades

Para cada entidade, foram identificados os atributos apresentados no Quadro 2. O atributo identificador (chave primária) é indicado entre parênteses.

Entidade	Atributos
Setor	id (chave), nome, descrição, emoji, ordem_menu, ativo, instituição.
Atendente	id (chave), nome, número_whatsapp, e-mail, ativo, cargo, url_avatar, bio, credencial_autenticação.
Conversa	id (chave), telefone_contato, nome_contato, status, instituição, avaliação_csat, criada_em, atualizada_em.
Mensagem	id (chave), conteúdo, direção, criada_em.
Transferência	id (chave), criada_em.
Contato	telefone (chave natural), apelido, anotações, atualizado_em.
RespostaRápida	id (chave), título, conteúdo, atalho, ativa, criada_em.
ChaveAPI	id (chave), nome, e-mail_responsável, prefixo, ativa, criada_em, último_uso_em.
Fluxo	id (chave), nome, nós, arestas, ativo, criado_em, atualizado_em.

Quadro 2 — Atributos identificados por entidade. Fonte: Próprio autor.

4.4 Relacionamentos e Cardinalidades

Os relacionamentos identificados entre as entidades, com suas cardinalidades máximas, são apresentados no Quadro 3.

Relacionamento	Cardinalidade	Descrição
Setor possui Atendente	1:N	Um setor possui muitos atendentes; cada atendente pertence a um único setor.
Setor atende Conversa	1:N	Um setor pode estar associado a várias conversas; cada conversa pertence a no máximo um setor.
Atendente conduz Conversa	1:N	Um atendente pode conduzir várias conversas; cada conversa tem no máximo um atendente responsável.
Conversa contém Mensagem	1:N	Uma conversa contém várias mensagens; cada mensagem pertence a uma única conversa.
Conversa sofre Transferência	1:N	Uma conversa pode ter várias transferências; cada transferência pertence a uma única conversa.
Setor é_origem_de Transferência	1:N	Um setor pode aparecer como origem em várias transferências.
Setor é_destino_de Transferência	1:N	Um setor pode aparecer como destino em várias transferências.
Atendente origina Transferência	1:N	Um atendente pode ter originado várias transferências.
Contato participa de Conversa	1:N	Um contato (identificado pelo telefone) pode ter várias conversas associadas.

Quadro 3 — Relacionamentos do modelo e suas cardinalidades máximas. Fonte: Próprio autor.

Considerando também as cardinalidades mínimas, destacam-se:

- Atendente → Setor: (1,1) — todo atendente deve, obrigatoriamente, pertencer a um setor;
- Conversa → Setor: (0,1) — uma conversa pode ainda não ter setor atribuído (durante o menu inicial de escolha);
- Conversa → Atendente: (0,1) — uma conversa pode estar em fila, ainda sem atendente designado;
- Mensagem → Conversa: (1,1) — toda mensagem está obrigatoriamente vinculada a uma conversa;
- Transferência → Conversa: (1,1) — toda transferência refere-se a uma conversa existente.

Observa-se que todos os relacionamentos identificados são do tipo 1:N. Não foram identificados relacionamentos N:M no domínio, o que dispensa, no mapeamento lógico, a criação de tabelas associativas.

4.5 Diagrama Entidade-Relacionamento (DER)

A Figura 1 apresenta o Diagrama Entidade-Relacionamento em notação Chen, sintetizando graficamente as entidades, atributos e relacionamentos descritos.

[ESPAÇO RESERVADO PARA IMAGEM]

Figura 1 — Diagrama Entidade-Relacionamento (DER) em notação Chen. Fonte: Próprio autor.

Como apoio à leitura, segue uma representação textual simplificada dos relacionamentos identificados (notação: «entidade (mín,máx) — relacionamento — (mín,máx) entidade»):

SETOR	(0,N)	—possui—	(1,1)	ATENDENTE
SETOR	(0,N)	—atende—	(0,1)	CONVERSA
ATENDENTE	(0,N)	—conduz—	(0,1)	CONVERSA
CONVERSA	(1,1)	—contém—	(1,N)	MENSAGEM
CONVERSA	(1,1)	—sofre—	(0,N)	TRANSFERÊNCIA
SETOR	(0,N)	—origem—	(0,1)	TRANSFERÊNCIA
SETOR	(0,N)	—destino—	(1,1)	TRANSFERÊNCIA
ATENDENTE	(0,N)	—originou—	(0,1)	TRANSFERÊNCIA
CONTATO	(1,1)	—possui—	(0,N)	CONVERSA

CONTATO, RESPOSTA_RÁPIDA, CHAVE_API, FLUXO → entidades independentes

5. Modelagem Lógica

Esta seção apresenta o Modelo Lógico relacional resultante do mapeamento do MER, seguindo as regras formais de conversão. Adicionalmente, são apresentadas as restrições de integridade aplicadas, a análise de dependências funcionais e a verificação das formas normais.

5.1 Mapeamento do MER para o Modelo Relacional

As seguintes regras de conversão foram aplicadas na transformação do MER em esquema relacional:

- Cada entidade do MER torna-se uma relação (tabela), com seus atributos mapeados em colunas;
- A chave primária da entidade torna-se a chave primária da tabela. Em geral foi adotada uma chave artificial (surrogate key) do tipo INTEGER autoincrementado; exceção feita à entidade Contato, que utiliza o próprio telefone como chave natural;
- Relacionamentos do tipo 1:N são materializados por uma chave estrangeira na tabela do lado N, referenciando a chave primária da tabela do lado 1;
- Atributos opcionais são mapeados como colunas com permissão de valor nulo (NULL) ou com valor padrão (DEFAULT) apropriado;
- Atributos derivados (por exemplo, contadores de mensagens não lidas) não são armazenados — serão computados em tempo de consulta;
- Como todos os relacionamentos identificados são 1:N, não foi necessária a criação de tabelas associativas.

5.2 Esquema Relacional

O esquema relacional resultante é composto por nove relações. Na notação a seguir, os atributos sublinhados representam a chave primária e os atributos seguidos de FK→ indicam chaves estrangeiras.

```
sectors ( id PK, name, description, emoji, menu_order, active, institution )

attendants ( id PK, name, sector_id FK→sectors.id, whatsapp_number, email, active, role, avatar_url, bio, credencial )

conversations ( id PK, contact_phone, contact_name, sector_id FK→sectors.id, attendant_id FK→attendants.id, status, institution, csat_rating, created_at, updated_at )

messages ( id PK, conversation_id FK→conversations.id, content, direction, created_at )
```

```

conversation_transfers ( id PK, conversation_id FK→conversations.id,
                        from_sector_id FK→sectors.id,
                        to_sector_id   FK→sectors.id,
                        from_attendant_id FK→attendants.id,
                        created_at )

contacts ( phone PK, name_override, notes, updated_at )

quick_replies ( id PK, title, content, shortcut, active, created_at )

api_keys ( id PK, name, user_email, key_prefix, active,
           created_at, last_used_at )

flows ( id PK, name, nodes, edges, active, created_at, updated_at )

```

5.3 Dicionário de Dados

Esta subseção detalha, tabela a tabela, todos os atributos do esquema, seus tipos genéricos, restrições e valores padrão. A convenção de tipos é independente de SGBD: INTEGER para inteiros e chaves; TEXT para cadeias de caracteres; valores booleanos representados como INTEGER (0 ou 1).

5.3.1 Tabela sectors

Armazena os setores de atendimento. Cada instituição possui seu próprio conjunto de setores, ordenados pelo atributo menu_order.

Atributo	Tipo	Restrições / Padrão	Descrição
id	INTEGER	PK, autoincremento	Identificador único do setor.
name	TEXT	NOT NULL	Nome do setor (ex.: Financeiro).
description	TEXT	DEFAULT "	Descrição exibida ao usuário.
emoji	TEXT	DEFAULT "	Emoji ilustrativo no menu.
menu_order	INTEGER	DEFAULT 0	Ordem de exibição no menu.
active	INTEGER	DEFAULT 1	Setor ativo (1) ou inativo (0).
institution	TEXT	DEFAULT 'crianca'	Instituição: 'crianca' ou 'mae'.

5.3.2 Tabela attendants

Cadastro dos atendentes habilitados a operar o painel. Cada atendente pertence a um setor.

Atributo	Tipo	Restrições / Padrão	Descrição
id	INTEGER	PK, autoincremento	Identificador único do atendente.
name	TEXT	NOT NULL	Nome do atendente.
sector_id	INTEGER	NOT NULL, FK	Setor ao qual pertence.
whatsapp_number	TEXT	DEFAULT "	Número para notificações.
email	TEXT	DEFAULT "	E-mail de identificação.

Atributo	Tipo	Restrições / Padrão	Descrição
active	INTEGER	DEFAULT 1	Indicador de atendente ativo.
role	TEXT	DEFAULT "	Cargo/função (campo descritivo).
avatar_url	TEXT	DEFAULT "	URL da foto de perfil.
bio	TEXT	DEFAULT "	Breve descrição do atendente.
credencial	TEXT	DEFAULT "	Credencial de autenticação do atendente.

Chave estrangeira: sector_id → sectors(id).

5.3.3 Tabela conversations

Representa cada atendimento individual. O atributo status modela o ciclo de vida da conversa por meio de uma máquina de estados (ver Seção 6).

Atributo	Tipo	Restrições / Padrão	Descrição
id	INTEGER	PK, autoincremento	Identificador único da conversa.
contact_phone	TEXT	NOT NULL	Telefone do contato.
contact_name	TEXT	DEFAULT "	Nome informado pelo WhatsApp.
sector_id	INTEGER	FK, anulável	Setor atribuído à conversa.
attendant_id	INTEGER	FK, anulável	Atendente responsável.
status	TEXT	DEFAULT 'pending_menu'	Estado atual da conversa.
institution	TEXT	DEFAULT "	Instituição escolhida pelo contato.
csat_rating	TEXT	DEFAULT NULL	Avaliação de satisfação (1 a 4).
created_at	TEXT	DEFAULT datetime()	Data/hora de criação.
updated_at	TEXT	DEFAULT datetime()	Data/hora da última atualização.

Chaves estrangeiras: sector_id → sectors(id); attendant_id → attendants(id).

5.3.4 Tabela messages

Registra cada mensagem trocada em uma conversa. O atributo direction distingue mensagens recebidas do contato ('in') das enviadas pelo sistema ('out').

Atributo	Tipo	Restrições / Padrão	Descrição
id	INTEGER	PK, autoincremento	Identificador único da mensagem.
conversation_id	INTEGER	NOT NULL, FK	Conversa à qual pertence.
content	TEXT	NOT NULL	Conteúdo textual da mensagem.
direction	TEXT	DEFAULT 'in'	Sentido: 'in' (recebida) ou 'out' (enviada).
created_at	TEXT	DEFAULT datetime()	Data/hora do registro.

Chave estrangeira: conversation_id → conversations(id).

5.3.5 Tabela conversation_transfers

Mantém o histórico de transferências de uma conversa entre setores, preservando a rastreabilidade do encaminhamento.

Atributo	Tipo	Restrições / Padrão	Descrição
id	INTEGER	PK, autoincremento	Identificador da transferência.
conversation_id	INTEGER	NOT NULL, FK	Conversa transferida.
from_sector_id	INTEGER	FK, anulável	Setor de origem.
to_sector_id	INTEGER	NOT NULL, FK	Setor de destino.
from_attendant_id	INTEGER	FK, anulável	Atendente que originou a transferência.
created_at	TEXT	DEFAULT datetime()	Data/hora da transferência.

Chaves estrangeiras: conversation_id → conversations(id); from_sector_id e to_sector_id → sectors(id); from_attendant_id → attendants(id).

5.3.6 Tabela contacts

Armazena informações complementares sobre cada número de contato. A chave primária é o próprio número de telefone (chave natural).

Atributo	Tipo	Restrições / Padrão	Descrição
phone	TEXT	PK	Número de telefone (chave primária natural).
name_override	TEXT	DEFAULT "	Apelido com precedência de exibição.
notes	TEXT	DEFAULT "	Anotações livres sobre o contato.
updated_at	TEXT	DEFAULT datetime()	Data/hora da última edição.

5.3.7 Tabela quick_replies

Cadastro de respostas rápidas: mensagens-modelo reutilizáveis que os atendentes inserem com um atalho, agilizando o atendimento.

Atributo	Tipo	Restrições / Padrão	Descrição
id	INTEGER	PK, autoincremento	Identificador da resposta.
title	TEXT	NOT NULL	Título descritivo.
content	TEXT	NOT NULL	Texto que será inserido.
shortcut	TEXT	DEFAULT "	Atalho de invocação (ex.: /ola).
active	INTEGER	DEFAULT 1	Indicador de resposta ativa.
created_at	TEXT	DEFAULT datetime()	Data/hora de criação.

5.3.8 Tabela api_keys

Chaves de acesso para integração programática com a API. Persiste-se apenas um prefixo identificador e a representação protegida da chave.

Atributo	Tipo	Restrições / Padrão	Descrição
id	INTEGER	PK, autoincremento	Identificador da chave.
name	TEXT	NOT NULL	Nome descritivo.
user_email	TEXT	NOT NULL	E-mail do responsável.
key_prefix	TEXT	NOT NULL	Prefixo visível (ex.: btr_cd6f...).
active	INTEGER	DEFAULT 1	Indicador de chave ativa.
created_at	TEXT	DEFAULT datetime()	Data/hora de criação.
last_used_at	TEXT	anulável	Data/hora do último uso.

5.3.9 Tabela flows

Persiste os fluxos conversacionais do bot. Os atributos nodes e edges armazenam estruturas serializadas em formato JSON.

Atributo	Tipo	Restrições / Padrão	Descrição
id	INTEGER	PK, autoincremento	Identificador do fluxo.
name	TEXT	NOT NULL	Nome do fluxo.
nodes	TEXT	NOT NULL, DEFAULT '[]'	Nós do fluxo (JSON serializado).
edges	TEXT	NOT NULL, DEFAULT '[]'	Arestas do fluxo (JSON serializado).
active	INTEGER	DEFAULT 1	Indicador de fluxo ativo.
created_at	TEXT	DEFAULT datetime()	Data/hora de criação.
updated_at	TEXT	DEFAULT datetime()	Data/hora da última atualização.

5.4 Restrições de Integridade do Modelo

O modelo aplica as quatro categorias de restrições discutidas na fundamentação teórica.

5.4.1 Integridade de Entidade

- Toda relação possui uma chave primária declarada;
- Todas as tabelas com chave artificial utilizam um atributo INTEGER autoincrementado;
- Na relação contacts, a chave primária natural é phone (TEXT), assegurando unicidade do contato.

5.4.2 Integridade Referencial

As seguintes chaves estrangeiras compõem o modelo:

- attendants.sector_id → sectors(id) — obrigatória (NOT NULL);

- `conversations.sector_id` → `sectors(id)` — opcional (anulável);
- `conversations.attendant_id` → `attendants(id)` — opcional (anulável);
- `messages.conversation_id` → `conversations(id)` — obrigatória (NOT NULL);
- `conversation_transfers.conversation_id` → `conversations(id)` — obrigatória;
- `conversation_transfers.from_sector_id` → `sectors(id)` — opcional;
- `conversation_transfers.to_sector_id` → `sectors(id)` — obrigatória;
- `conversation_transfers.from_attendant_id` → `attendants(id)` — opcional.

5.4.3 Integridade de Domínio

- Atributos inteiros (`active`, `menu_order`, `csat_rating`) restritos a valores numéricos;
- Atributos textuais com domínios enumerados implícitos: `direction` ∈ {'in', 'out'}; `status` ∈ {'pending_institution', 'pending_menu', 'waiting', 'active', 'closed'}; `institution` ∈ {'crianca', 'mae'};
- Datas armazenadas em formato ISO 8601 (TEXT).

5.4.4 Integridade Definida pelo Usuário

- O atributo `direction` de `messages` deve assumir apenas os valores 'in' ou 'out';
- O atributo `status` de `conversations` deve respeitar a máquina de estados definida na Seção 6;
- Em `conversation_transfers`, deve valer `from_sector_id` ≠ `to_sector_id`;
- Em `api_keys`, deve haver unicidade de `key_prefix`.

5.5 Análise de Dependências Funcionais

As principais dependências funcionais identificadas em cada relação do modelo são apresentadas no Quadro 4.

Relação	Dependências Funcionais
<i>sectors</i>	<code>id</code> → <code>name</code> , <code>description</code> , <code>emoji</code> , <code>menu_order</code> , <code>active</code> , <code>institution</code> (<code>institution</code> , <code>menu_order</code>) → <code>id</code> (chave alternada)
<i>attendants</i>	<code>id</code> → <code>name</code> , <code>sector_id</code> , <code>whatsapp_number</code> , <code>email</code> , <code>active</code> , <code>role</code> , <code>avatar_url</code> , <code>bio</code> , <code>credencial</code> <code>email</code> → <code>id</code> (chave alternada, assumindo unicidade)
<i>conversations</i>	<code>id</code> → <code>contact_phone</code> , <code>contact_name</code> , <code>sector_id</code> , <code>attendant_id</code> , <code>status</code> , <code>institution</code> , <code>csat_rating</code> , <code>created_at</code> , <code>updated_at</code>
<i>messages</i>	<code>id</code> → <code>conversation_id</code> , <code>content</code> , <code>direction</code> , <code>created_at</code>

Relação	Dependências Funcionais
<i>conversation_transfers</i>	$id \rightarrow conversation_id, from_sector_id, to_sector_id, from_attendant_id, created_at$
<i>contacts</i>	$phone \rightarrow name_override, notes, updated_at$
<i>quick_replies</i>	$id \rightarrow title, content, shortcut, active, created_at$ $shortcut \rightarrow id$ (chave alternada)
<i>api_keys</i>	$id \rightarrow name, user_email, key_prefix, active, created_at, last_used_at$ $key_prefix \rightarrow id$ (chave alternada)
<i>flows</i>	$id \rightarrow name, nodes, edges, active, created_at, updated_at$

Quadro 4 — Dependências funcionais por relação. Fonte: Próprio autor.

Em todas as relações, a chave primária determina funcionalmente todos os demais atributos — condição necessária para a 2ª Forma Normal. Nas relações em que se reconhece outro atributo capaz de determinar a chave primária, este é classificado como chave alternada (candidate key).

5.6 Verificação das Formas Normais

A análise tabela a tabela demonstra que o modelo está em 3ª Forma Normal, conforme detalhado a seguir.

5.6.1 Primeira Forma Normal (1FN)

- Todos os atributos do esquema são atômicos sob o ponto de vista relacional;
- Não há atributos multivalorados nem grupos repetidos;
- Cada célula da relação contém exatamente um valor do domínio declarado.

Observação: os atributos *nodes* e *edges* da relação *flows* armazenam estruturas serializadas em formato JSON. Sob a interpretação estrita do modelo relacional, esses valores poderiam ser vistos como uma violação da atomicidade. Adota-se, contudo, a interpretação pragmática moderna, em que o JSON é tratado como um valor único e indivisível pela aplicação. Uma alternativa estritamente normalizada exigiria a criação de duas novas entidades — *FluxoNo* e *FluxoAresta* —, decisão que pode ser considerada em evolução futura.

5.6.2 Segunda Forma Normal (2FN)

- O modelo atende trivialmente à 2FN: todas as nove relações possuem chave primária simples (composta por um único atributo);
- Como não existem chaves primárias compostas no esquema, não há possibilidade de dependências parciais a eliminar.

5.6.3 Terceira Forma Normal (3FN)

- Não foram identificadas dependências transitivas entre atributos não-chave em nenhuma das nove relações;
- Na relação *conversations*, os atributos *contact_name* e *contact_phone* são propriedades da própria conversa (o nome de exibição é capturado no momento da conversa e pode variar entre conversas distintas de um mesmo número). A informação consolidada do contato é mantida separadamente na relação *contacts*;
- Em *attendants*, *sector_id* é uma chave estrangeira; o nome do setor (*name*) reside exclusivamente em *sectors*, evitando duplicação;
- Em *conversations*, o nome do setor e o nome do atendente não são armazenados redundantemente — apenas as chaves estrangeiras.

Conclui-se que o modelo está em 3ª Forma Normal (3FN), condição que minimiza a redundância e elimina as principais anomalias de inserção, atualização e remoção típicas de modelos não-normalizados.

6. Modelagem do Ciclo de Vida das Conversas

Além da modelagem estrutural — entidades, atributos e relacionamentos — o domínio exige a modelagem do comportamento das conversas ao longo do tempo. O atributo *status* da entidade *Conversa* representa o estado atual do atendimento e segue uma máquina de estados finitos, cujos estados e transições são apresentados no Quadro 5.

Estado	Significado
<i>pending_institution</i>	Aguardando o contato escolher a instituição (Casa da Criança ou Casa da Mãe).
<i>pending_menu</i>	Aguardando o contato escolher o setor desejado.
<i>waiting</i>	Na fila do setor, aguardando um atendente assumir o atendimento.
<i>active</i>	Em atendimento ativo por um atendente.
<i>closed</i>	Atendimento encerrado.

Quadro 5 — Estados do ciclo de vida de uma conversa. Fonte: Próprio autor.

As transições válidas entre estados são:

- *pending_institution* → *pending_menu* (após a escolha da instituição);
- *pending_menu* → *waiting* (após a escolha do setor);
- *waiting* → *active* (quando um atendente assume o atendimento);
- *active* → *closed* (encerramento manual do atendimento);

- waiting → closed (encerramento automático por expiração na fila);
- active → closed (encerramento automático por inatividade).
- A modelagem dessa máquina de estados é parte integrante da modelagem de dados, pois define o domínio do atributo status e estabelece quais transformações de estado são consideradas válidas, restringindo as atualizações permitidas sobre a relação conversations.

7. Considerações sobre o SGBD-alvo

Embora o foco deste trabalho acadêmico seja a modelagem de dados nos níveis conceitual e lógico, registra-se brevemente, a título de contexto, que o Modelo Lógico apresentado será materializado em um SGBD relacional embarcado — o SQLite 3 — adequado ao porte da aplicação e aos recursos disponíveis na instituição parceira. Trata-se de uma escolha de nível físico, externa ao escopo deste trabalho.

Essa decisão não afeta a modelagem apresentada, uma vez que o Modelo Lógico relacional, por definição, é independente do SGBD. O mesmo esquema poderia ser implementado, sem alterações estruturais, em PostgreSQL, MySQL, Oracle, SQL Server ou qualquer outro SGBD relacional compatível com o padrão SQL. As decisões relativas à organização física do armazenamento, índices e configuração do SGBD pertencem à etapa subsequente de projeto físico e não fazem parte deste trabalho.

8. Considerações Finais

Este trabalho acadêmico apresentou a modelagem completa do banco de dados que sustentará o sistema de atendimento das instituições Casa da Criança Baturia e Casa da Mãe Baturia. O trabalho percorreu o Modelo Conceitual e o Modelo Lógico de abstração, ancorado na fundamentação teórica do Modelo Entidade-Relacionamento, do modelo relacional e da teoria de normalização.

A modelagem conceitual identificou nove entidades, organizadas em quatro domínios funcionais coerentes (organizacional, atendimento, apoio e integração), com relacionamentos predominantemente do tipo 1:N. O Diagrama Entidade-Relacionamento, em notação Chen, sintetiza graficamente o modelo.

O mapeamento para o Modelo Lógico produziu nove relações, com chaves primárias e estrangeiras explícitas, restrições de integridade documentadas nas

quatro categorias (entidade, referencial, domínio e definidas pelo usuário) e dicionário de dados completo. A análise das dependências funcionais e a verificação das formas normais confirmam que o modelo está em 3ª Forma Normal, atendendo às boas práticas consagradas no projeto de bancos de dados relacionais.

Adicionalmente, modelou-se o comportamento dinâmico das conversas por meio de uma máquina de estados finitos, complementando a modelagem estrutural com uma representação comportamental do ciclo de vida do atendimento.

Por fim, conclui-se o trabalho com a entrega de uma modelagem coesa, normalizada, livre das principais anomalias relacionais e pronta para ser implementada em qualquer SGBD relacional, constituindo uma base sólida para a etapa subsequente de implementação do sistema.

Referências

CHEN, P. P. The Entity-Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems, v. 1, n. 1, p. 9–36, 1976.

CODD, E. F. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, v. 13, n. 6, p. 377–387, 1970.

DATE, C. J. Introdução a Sistemas de Bancos de Dados. 8. ed. Rio de Janeiro: Elsevier.

ELMASRI, R.; NAVATHE, S. B. Sistemas de Banco de Dados. 7. ed. São Paulo: Pearson.

HEUSER, C. A. Projeto de Banco de Dados. 6. ed. Porto Alegre: Bookman, 2009.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados. 7. ed. Rio de Janeiro: Elsevier.

IPEA — Instituto de Pesquisa Econômica Aplicada. Mapa das Organizações da Sociedade Civil. Disponível em <<https://mapaosc.ipea.gov.br/indicadores>>. Acesso em maio de 2026.