

**Faculdade Processus**

**Curso de ADS e SI / 2º 2025**

**Disciplina: Atividade de Extensão Modelagem e Diagnóstico de Negócio**

**Professor: Antônio Carlos Guedes de Moraes**

Projeto do Site da Igreja Presbiteriana

**Equipe: Jhonata Cassio Batista de Moura**

# ÍNDICE

## Sumário

ÍNDICE .....	1
INTRODUÇÃO .....	3
Modelagem Conceitual do Banco de Dados .....	4
Entidades e Atributos .....	4
Modelagem Lógica do Banco de Dados .....	7
Script SQL de Criação do Banco de Dados .....	7
Regras de Integridade .....	9
1. Integridade de Domínio .....	9
2. Integridade de Entidade .....	9
3. Integridade Referencial .....	9
4. Integridade de Negócio .....	9
Normalização .....	10
1ª Forma Normal (1FN) .....	10
2ª Forma Normal (2FN) .....	10
• Todas as tabelas com chave primária simples possuem dependência total da PK (ex.: Membros, Departamentos, Ofertas, Eventos, Visitantes) .....	10
• A única tabela com chave composta é Membros_Departamentos, e nela: .....	10
○ Não há dependência parcial entre os atributos. ....	10
○ Todos os atributos dependem da combinação das chaves id_membro e id_departamento. ....	10
• Não existe atributo que dependa apenas de parte da chave composta .....	10
3ª Forma Normal (3FN) .....	10
• Nenhum atributo depende de outro atributo não-chave. ....	10
• Exemplos aplicados: .....	10
○ O campo valor da oferta depende apenas da PK id_oferta, e não de outro atributo. ....	10
○ O campo convidado_por da tabela Visitantes depende exclusivamente da relação com a PK da tabela Membros. ....	10
○ O campo funcao em Membros_Departamentos depende da combinação das chaves e não de outro campo não-chave. ....	10
• Não foram incluídos atributos derivados ou informações duplicadas entre tabelas. ....	10
Relacionamentos do Banco de Dados .....	10
1. Membro – Ofertas (1:N) .....	10
• Um membro pode realizar várias ofertas (dízimos ou ofertas voluntárias) .....	10
• Cada oferta é registrada para um único membro. ....	10
• FK implementada: Ofertas.id_membro → Membros.id_membro .....	11
2. Membro – Visitantes (1:N) .....	11
• Um membro pode participar de vários departamentos. ....	11
• Um departamento pode ter vários membros. ....	11
• A relação é definida pela tabela associativa Membros_Departamentos. ....	11
• FKs implementadas: .....	11
○ Membros_Departamentos.id_membro → Membros.id_membro .....	11
○ Membros_Departamentos.id_departamento → Departamentos.id_departamento .....	11

Departamentos – Membros_Departamentos (1:N) .....	11
• Um departamento contém vários membros vinculados a ele por funções e datas de entrada. ....	11
• FK implementada: Membros_Departamentos.id_departamento → Departamentos.id_departamento .....	11
Membros – Membros_Departamentos (1:N) .....	11
• Cada membro pode estar associado a vários departamentos. ....	11
• FK implementada:.....	11
• Membros_Departamentos.id_membro → Membros.id_membro .....	11
Eventos (sem FK obrigatória na estrutura atual) .....	11
• A tabela <i>Eventos</i> possui o campo <i>responsavel</i> , mas ele é apenas texto.....	11
• Não estabelece relacionamento formal com Membros ou Usuários. ....	11
• Caso deseje, posso transformar “responsável” em chave estrangeira. ....	11
<b>Conclusão .....</b>	<b>12</b>

## INTRODUÇÃO

Este documento apresenta a continuação do projeto de extensão desenvolvido junto à Igreja Presbiteriana, A Igreja Presbiteriana é reconhecida pelo Compromisso com o ensino bíblico, Trabalho com famílias e jovens, Cultos reverentes e organizados e Ações sociais na comunidade, A tradição presbiteriana valoriza profundamente o estudo da Bíblia, a pregação expositiva e a formação espiritual contínua dos membros.. Com base nos princípios bíblicos de comunhão, ensino e oração, a igreja mantém atividades como cultos, visitas missionárias, ações solidárias e campanhas.

No semestre anterior, o diagnóstico identificou que a igreja ainda realiza grande parte de sua organização de forma manual, o que gera perda de informações e dificulta o registro e a divulgação de suas ações. Por isso, neste semestre, o foco do trabalho foi avançar na estruturação do banco de dados que servirá de base para um futuro sistema digital capaz de apoiar a gestão e fortalecer a comunicação da instituição.

As atividades realizadas incluíram:

- **Modelagem Conceitual**, com a identificação das principais informações da igreja e seus relacionamentos, representados por meio do DER.
- **Modelagem Lógica**, convertendo o modelo conceitual em tabelas relacionais com chaves primárias e estrangeiras.
- **Normalização**, garantindo que os dados fiquem organizados e sem redundâncias.
- **Regras de Integridade**, assegurando que as informações armazenadas sejam consistentes e confiáveis.

Com essas etapas, o projeto avança para uma base sólida, contribuindo para que a Igreja tenha um sistema mais organizado, moderno e alinhado às suas necessidades sociais, administrativas e missionárias.

## Modelagem Conceitual do Banco de Dados

A modelagem conceitual foi desenvolvida com base nos requisitos, casos de uso e regras de negócio levantados no semestre anterior. O objetivo desta etapa é identificar de forma clara **quais informações precisam ser armazenadas e como elas se relacionam**, preparando a estrutura inicial do banco de dados que dará suporte ao sistema da Igreja Presbiteriana

A partir da análise das funcionalidades foram identificadas as principais **entidades**, seus **atributos essenciais** e os **relacionamentos** entre elas.

### Entidades e Atributos

Na modelagem conceitual:

- **Entidades** são os objetos principais que precisam ser armazenados no banco de dados (ex.: Membro, Evento, Doação).
- **Atributos** são as informações que descrevem cada entidade (ex.: Nome, CPF, Data, Valor).

As entidades representam “o que” será guardado e os atributos mostram “quais dados” caracterizam cada entidade, são eles:

#### • Membro

Representa todas as pessoas cadastradas na igreja, incluindo membros ativos, inativos ou recém-chegados.

Contém informações essenciais para identificação e acompanhamento.

#### Atributos:

- **ID\_Membro (PK)**
- Nome
- Data\_Nascimento
- Sexo
- Telefone
- Email
- Endereço
- Data\_Batismo
- Status (ativo, inativo)

#### • Departamento

Representa os setores e ministérios da igreja, como louvor, ensino, recepção, infantil, etc.

**Atributos:**

- **ID\_Departamento (PK)**
- Nome
- Descricao

**• Membros\_Departamentos**

Tabela associativa que registra quais membros pertencem a quais departamentos, incluindo a função desempenhada.

**Atributos:**

- **ID\_Membro (FK)**
- **ID\_Departamento (FK)**
- Funcao
- Data\_Entrada

**• Evento**

Inclui cultos, reuniões, campanhas, ensaios e eventos especiais organizados pela igreja.

**Atributos:**

- **ID\_Evento (PK)**
- Titulo
- Data
- Local
- Tipo (Culto, Reunião, Evento Especial)
- Responsavel (*campo textual — não é FK na estrutura atual*)

**• Oferta**

Registra as contribuições financeiras feitas pelos membros, como dízimos e ofertas.

**Atributos:**

- **ID\_Oferta (PK)**
- Tipo (Dízimo, Oferta)
- Valor
- Data
- ID\_Membro (FK)

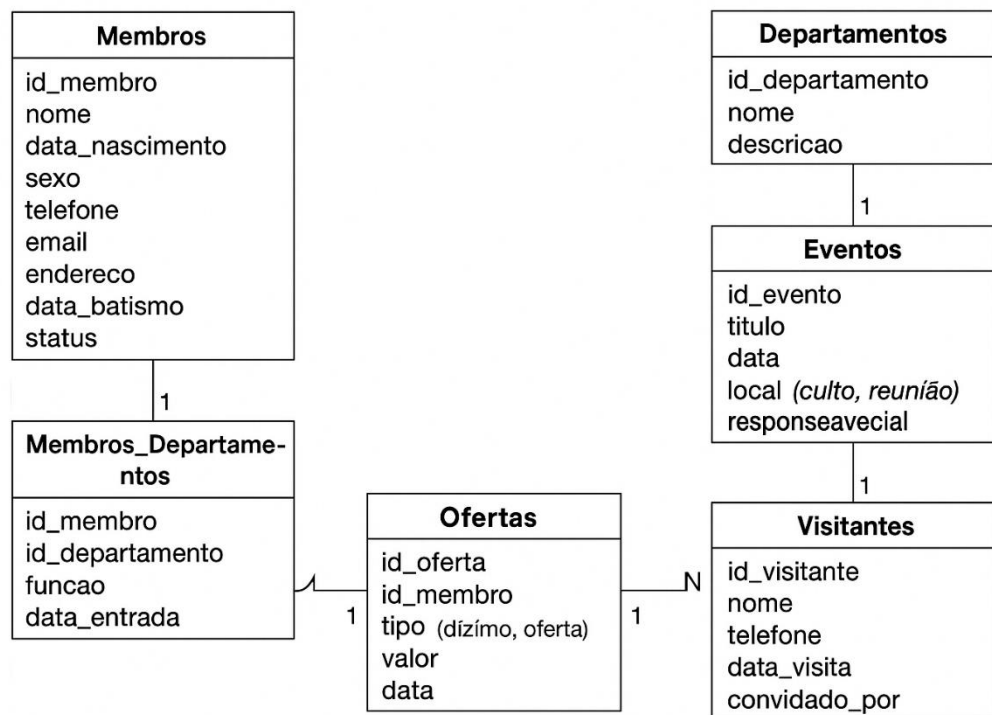
• **Visitante**

Representa pessoas que visitaram a igreja e foram convidadas por um membro.

**Atributos:**

- **ID\_Visitante (PK)**
- Nome
- Telefone
- Data\_Visita
- Convidado\_Por (FK – Membro)

**DER CONCEITUAL**



## Modelagem Lógica do Banco de Dados

A modelagem lógica foi desenvolvida a partir do modelo conceitual e representada no MySQL Workbench por meio de um Diagrama Entidade-Relacionamento Estendido (EER). Nesta etapa, cada entidade identificada no modelo conceitual foi convertida em uma tabela, com seus respectivos atributos, tipos de dados, chaves primárias e chaves estrangeiras.

A modelagem lógica busca traduzir as necessidades da Igreja Presbiteriana para estruturas compatíveis com o SGBD MySQL, garantindo clareza, precisão e coerência na implementação física posterior.

### Principais elementos definidos na modelagem lógica:

- **Tabelas:**  
Todas as entidades foram transformadas em tabelas individuais:  
Membro, Departamento, Evento, Oferta, Visitante
- **Tipos de dados:**  
Cada atributo recebeu um tipo compatível com o MySQL, como:  
INT, VARCHAR, CHAR, DATE, DATETIME, DECIMAL, LONGTEXT, ENUM.
- **Chaves Primárias (PK):**  
Definidas como INT AUTO\_INCREMENT, garantindo unicidade e facilidade de identificação.
- **Chaves Estrangeiras (FK):**  
Implementadas conforme os relacionamentos definidos no DER, com ações estabelecidas para exclusão e atualização (CASCADE ou NO ACTION).
- **Índices:**  
Criados para agilizar consultas e manter eficiência na busca de registros.

A modelagem lógica tornou possível definir todas as regras do banco de dados antes de gerar o script SQL final.

## Script SQL de Criação do Banco de Dados

Após a elaboração da modelagem conceitual e lógica, foi realizada a etapa de implementação física no MySQL Workbench. O código SQL abaixo foi gerado através da ferramenta do MySQL, sendo responsável pela criação das tabelas, relacionamentos, restrições de integridade e índices que compõem o banco de dados da Igreja Presbiteriana.

Esse script representa uma parte da estrutura final do banco, pronta para ser utilizada no desenvolvimento do sistema.

-- MySQL Workbench Forward Engineering

```

CREATE TABLE Membros (
  id_membro INT AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(100),
  data_nascimento DATE,
  sexo CHAR(1),
  telefone VARCHAR(20),
  email VARCHAR(100),
  endereco VARCHAR(200),
  data_batismo DATE,
  status ENUM('ativo', 'inativo') DEFAULT 'ativo'
);

```

```

CREATE TABLE Departamentos (
  id_departamento INT AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(100),
  descricao TEXT
);

```

```

CREATE TABLE Membros_Departamentos (
  id_membro INT,
  id_departamento INT,
  funcao VARCHAR(100),
  data_entrada DATE,
  PRIMARY KEY (id_membro, id_departamento),
  FOREIGN KEY (id_membro) REFERENCES Membros(id_membro),
  FOREIGN KEY (id_departamento) REFERENCES Departamentos(id_departamento)
);

```

```

CREATE TABLE Eventos (
  id_evento INT AUTO_INCREMENT PRIMARY KEY,
  titulo VARCHAR(100),
  data DATETIME,
  local VARCHAR(150),
  tipo ENUM('culto', 'reunião', 'evento especial'),
  responsavel VARCHAR(100)
);

```

```

CREATE TABLE Ofertas (
  id_oferta INT AUTO_INCREMENT PRIMARY KEY,
  id_membro INT,
  tipo ENUM('dízimo', 'oferta'),
  valor DECIMAL(10,2),
  data DATE,
  FOREIGN KEY (id_membro) REFERENCES Membros(id_membro)
);

```

```
CREATE TABLE Visitantes (  
  id_visitante INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(100),  
  telefone VARCHAR(20),  
  data_visita DATE,  
  convidado_por INT,  
  FOREIGN KEY (convidado_por) REFERENCES Membros(id_membro)  
);
```

## Regras de Integridade

As regras de integridade asseguram que os dados sejam válidos, completos e consistentes dentro do banco. No projeto, foram aplicadas quatro categorias:

### 1. Integridade de Domínio

Garante que cada coluna aceite apenas valores válidos.

### 2. Integridade de Entidade

Assegura que cada registro tenha uma identificação única.

Implementação:

- Todas as tabelas possuem chave primária `ID_* AUTO_INCREMENT`.
- Campo CPF do membro possui índice único (`CPF_UNIQUE`) para evitar duplicidade.

### 3. Integridade Referencial

Garantida por meio das chaves estrangeiras, controlando dependência entre tabelas.

### 4. Integridade de Negócio

Regras específicas da instituição foram incorporadas:

- Membro tem status padrão “ativo”.
- Movimentações financeiras devem ter obrigatoriamente um usuário responsável.
- Benefício só pode ser registrado para pessoas cadastradas como beneficiário.

## Normalização

Durante o desenvolvimento, o banco de dados foi normalizado para eliminar redundâncias e evitar anomalias em inserções, atualizações e exclusões.

### 1ª Forma Normal (1FN)

- Todos os atributos são **atômicos**, ou seja, não existem campos que armazenem múltiplos valores em um único atributo.
- Não foram utilizados campos multivalorados nem estruturas repetitivas.
- Exemplos aplicados:
  - Campos como *telefone*, email, tipo de oferta e status armazenam apenas um valor.
  - Na tabela *Membros\_Departamentos*, cada registro representa apenas uma função em um departamento..

### 2ª Forma Normal (2FN)

- Todas as tabelas com chave primária simples possuem dependência total da PK (ex.: *Membros*, *Departamentos*, *Ofertas*, *Eventos*, *Visitantes*).
- A única tabela com chave composta é *Membros\_Departamentos*, e nela:
  - Não há dependência parcial entre os atributos.
  - Todos os atributos dependem da combinação das chaves *id\_membro* e *id\_departamento*.
- Não existe atributo que dependa apenas de parte da chave composta.

### 3ª Forma Normal (3FN)

- Nenhum atributo depende de outro atributo não-chave.
- Exemplos aplicados:
  - O campo valor da oferta depende apenas da PK *id\_oferta*, e não de outro atributo.
  - O campo *convidado\_por* da tabela *Visitantes* depende exclusivamente da relação com a PK da tabela *Membros*.
  - O campo *funcao* em *Membros\_Departamentos* depende da combinação das chaves e não de outro campo não-chave.
- Não foram incluídos atributos derivados ou informações duplicadas entre tabelas.

## Relacionamentos do Banco de Dados

Todos os relacionamentos definidos no DER foram implementados no modelo lógico e físico:

### 1. Membro – Ofertas (1:N)

- Um membro pode realizar várias ofertas (dízimos ou ofertas voluntárias).
- Cada oferta é registrada para um único membro.

- FK implementada:  
Ofertas.id\_membro → Membros.id\_membro
- Membro – Visitantes (1:N)
- Um membro pode convidar vários visitantes.
- Cada visitante é vinculado ao membro que o convidou.
- FK implementada:  
Visitantes.convidado\_por → Membros.id\_membro

#### **.Membro – Departamentos (N:N através de tabela associativa)**

- Um membro pode participar de vários departamentos.
- Um departamento pode ter vários membros.
- A relação é definida pela tabela associativa Membros\_Departamentos.
- FKs implementadas:
  - Membros\_Departamentos.id\_membro → Membros.id\_membro
  - Membros\_Departamentos.id\_departamento → Departamentos.id\_departamento

#### **Departamentos – Membros\_Departamentos (1:N)**

- Um departamento contém vários membros vinculados a ele por funções e datas de entrada.
- FK implementada:  
Membros\_Departamentos.id\_departamento → Departamentos.id\_departamento

#### **Membros – Membros\_Departamentos (1:N)**

- Cada membro pode estar associado a vários departamentos.
- FK implementada:  
Membros\_Departamentos.id\_membro → Membros.id\_membro

#### **Eventos (sem FK obrigatória na estrutura atual)**

- A tabela *Eventos* possui o campo responsável, mas ele é apenas texto.
- Não estabelece relacionamento formal com Membros ou Usuários.
- Caso deseje, posso transformar “responsável” em chave estrangeira.

## Conclusão

A continuidade do projeto no segundo semestre permitiu transformar o estudo conceitual anterior em uma base sólida e estruturada no MySQL. A criação da modelagem lógica, o estabelecimento das regras de integridade, a normalização das tabelas e a definição dos relacionamentos garantem um banco de dados robusto, flexível e preparado para sustentar o sistema da Igreja Presbiteriana.

Com a aplicação dessas técnicas, o sistema futuro poderá oferecer:

- Maior organização das informações
- Segurança estrutural
- Rastreabilidade de ações
- Apoio à gestão administrativa, social e missionária
- Facilidade na geração de relatórios e histórico

O banco de dados está em desenvolvimento para sua implementação física completa, bem como para a futura construção da interface do sistema, atendendo às necessidades reais da instituição.