

LEVANTAMENTO DE REQUISITOS E MODELAGEM DE DADOS - REDE SOCIAL

LABORATÓRIO DB

1. Introdução

Este documento tem como objetivo detalhar os requisitos e a modelagem de dados necessários para o desenvolvimento de uma rede social para a conclusão da matéria de Laboratório de Banco de Dados. Através deste levantamento, serão especificadas as funcionalidades essenciais da plataforma, as interações entre os usuários e a estrutura de dados que suportará as operações do sistema.

2. Requisitos Funcionais

Esta seção descreve as funcionalidades específicas que o sistema deve oferecer para garantir que as necessidades dos usuários sejam atendidas.

2.1. Cadastro de Usuários

- O sistema deve permitir que os usuários se registrem na plataforma com nome, e-mail, senha, data de nascimento e foto de perfil.
- O sistema deve validar o e-mail e a senha antes de criar o perfil.
- O cadastro deve gerar um ID único para cada usuário.

2.2. Postagens

- O usuário deve ser capaz de criar postagens contendo texto e mídia (imagens ou vídeos).
- O sistema deve permitir que outras pessoas curtam ou comentem nas postagens.
- Cada postagem terá um contador de interações (curtidas e comentários).

2.3. Interações

- O sistema deve permitir que um usuário curta ou comente nas postagens de outros usuários.
- O tipo de interação (curtida ou comentário) deve ser registrado e associado à postagem.
- A cada interação, o número de interações de uma postagem deve ser atualizado automaticamente.

2.4. Grupos e Comunidades

- O sistema deve permitir a criação de grupos temáticos com nome, descrição e data de criação.

- Cada grupo pode ter múltiplos membros (usuários).
- O sistema deve gerenciar a relação de pertencimento do usuário aos grupos, utilizando uma tabela de associação.

2.5. Mensagens Privadas

- O sistema deve permitir o envio de mensagens privadas entre usuários.
- As mensagens devem ser armazenadas no banco de dados, associadas aos respectivos remetente e destinatário.

2.6. Notificações

- O sistema deve gerar notificações para os usuários sempre que ocorrerem interações relevantes (ex: novas curtidas ou comentários em suas postagens).
- O sistema deve registrar o tipo de notificação (curtida, comentário, nova postagem, etc.).

3. Modelagem de Dados

Nesta seção, serão descritas as principais entidades, seus atributos e relacionamentos, fundamentais para a implementação da rede social.

3.1. Entidades e Atributos

Usuário

- **ID_Usuario** (INT, PK): Identificador único do usuário.
- **Nome_Usuario** (VARCHAR(100)): Nome completo do usuário.
- **Email** (VARCHAR(50)): Endereço de e-mail do usuário.
- **Senha** (VARCHAR(20)): Senha do usuário.
- **Data_Nascimento** (DATETIME): Data de nascimento.
- **Foto_Perfil** (LONGBLOB): Foto do perfil.
- **Num_Seguidores** (INT): Número de seguidores do usuário.
- **Num_Seguindo** (INT): Número de pessoas seguidas.

Postagem

- **ID_Postagem** (INT, PK): Identificador único da postagem.
- **ID_Usuario** (INT, FK): Identificador do usuário que fez a postagem.
- **Texto** (TEXT): Texto da postagem.
- **Midia** (LONGBLOB): Arquivo de mídia anexado à postagem.
- **Data_Hora_Postagem** (DATETIME): Data e hora da postagem.

- **Num_Interacoes** (INT): Número total de interações (curtidas e comentários).

Interação

- **ID_Interacao** (INT, PK): Identificador único da interação.
- **ID_Postagem** (INT, FK): Identificador da postagem interagida.
- **ID_Usuario** (INT, FK): Identificador do usuário que fez a interação.
- **Tipo_Interacao** (VARCHAR(45)): Tipo de interação (curtida ou comentário).
- **Data_Hora_Interacao** (DATETIME): Data e hora da interação.

Comentário

- **ID_Comentario** (INT, PK): Identificador único do comentário.
- **ID_Interacao** (INT, FK): Identificador da interação associada.
- **Texto_Comentario** (TEXT): Texto do comentário.
- **Data_Hora_Comentario** (DATETIME): Data e hora do comentário.

Grupo

- **ID_Grupo** (INT, PK): Identificador único do grupo.
- **Nome_Grupo** (VARCHAR(100)): Nome do grupo.
- **Descricao** (TEXT): Descrição do grupo.
- **Data_Criacao** (DATETIME): Data de criação do grupo.

Usuário_has_Grupo (Tabela de Associação)

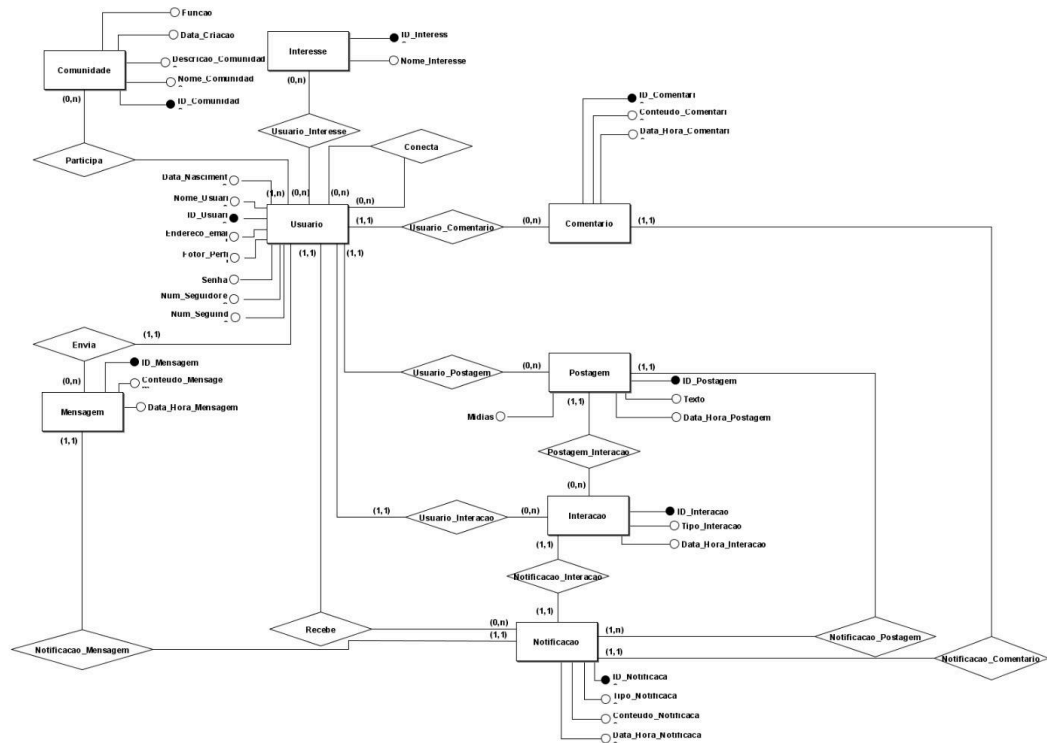
- **ID_Usuario** (INT, FK): Identificador do usuário.
- **ID_Grupo** (INT, FK): Identificador do grupo.

3.2. Relacionamentos

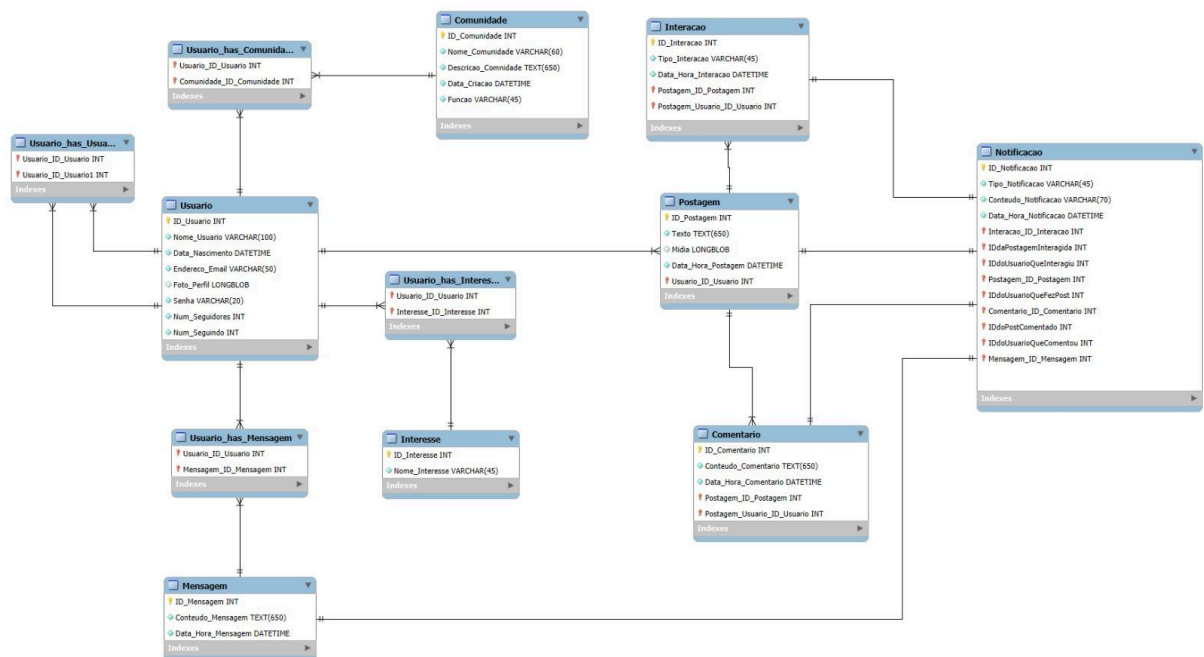
- **Usuário - Postagem**: Relação 1:1 (um para um). Cada postagem é associada a apenas um usuário.
- **Postagem - Interação**: Relação 1:1 (um para um). Cada postagem pode ter uma interação.
- **Interação - Comentário**: Relação 1:1 (um para um). Cada interação pode ter um comentário associado.
- **Usuário - Grupo**: Relação N:M (muitos para muitos). Um usuário pode pertencer a vários grupos e cada grupo pode ter múltiplos usuários.

4. Modelagem do Banco

4.1. Modelo Conceitual



4.2. Modelo Lógico



4.3. Modelo Físico

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE
,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-----
-- Schema mydb
-----

-----
-- Schema mydb
-----

CREATE SCHEMA IF NOT EXISTS mydb DEFAULT CHARACTER SET utf8 ;
USE mydb ;

-----
-- Table mydb.Usuario
-----
CREATE TABLE IF NOT EXISTS mydb.Usuario (
  ID_Usuario INT NOT NULL,
  Nome_Usuario VARCHAR(100) NOT NULL,
  Data_Nascimento DATETIME NOT NULL,
  Endereco_Email VARCHAR(50) NOT NULL,
  Foto_Perfil LONGBLOB NULL,
  Senha VARCHAR(20) NOT NULL,
  Num_Seguidores INT NOT NULL,
  Num_Seguindo INT NOT NULL,
  PRIMARY KEY (ID_Usuario),
  UNIQUE INDEX ID_Usuario_UNIQUE (ID_Usuario ASC) VISIBLE)
ENGINE = InnoDB;

-----
-- Table mydb.Comunidade
-----
CREATE TABLE IF NOT EXISTS mydb.Comunidade (
  ID_Comunidade INT NOT NULL,
  Nome_Comunidade VARCHAR(60) NOT NULL,
  Descricao_Comnidade TEXT(650) NOT NULL,
  Data_Criacao DATETIME NOT NULL,
  Funcao VARCHAR(45) NOT NULL,
  PRIMARY KEY (ID_Comunidade),
  UNIQUE INDEX ID_Comunidade_UNIQUE (ID_Comunidade ASC) VISIBLE)
ENGINE = InnoDB;

-----
-- Table mydb.Mensagem
```

```

-----
CREATE TABLE IF NOT EXISTS mydb.Mensagem (
  ID_Mensagem INT NOT NULL,
  Conteudo_Mensagem TEXT(650) NOT NULL,
  Data_Hora_Mensagem DATETIME NOT NULL,
  PRIMARY KEY (ID_Mensagem),
  UNIQUE INDEX ID_Mensagem_UNIQUE (ID_Mensagem ASC) VISIBLE)
ENGINE = InnoDB;

```

```

-----
-- Table mydb.Interesse

```

```

-----
CREATE TABLE IF NOT EXISTS mydb.Interesse (
  ID_Interesse INT NOT NULL,
  Nome_Interesse VARCHAR(45) NOT NULL,
  PRIMARY KEY (ID_Interesse),
  UNIQUE INDEX ID_Interesse_UNIQUE (ID_Interesse ASC) VISIBLE)
ENGINE = InnoDB;

```

```

-----
-- Table mydb.Postagem

```

```

-----
CREATE TABLE IF NOT EXISTS mydb.Postagem (
  ID_Postagem INT NOT NULL,
  Texto TEXT(650) NOT NULL,
  Midia LONGBLOB NULL,
  Data_Hora_Postagem DATETIME NOT NULL,
  Usuario_ID_Usuario INT NOT NULL,
  PRIMARY KEY (ID_Postagem, Usuario_ID_Usuario),
  UNIQUE INDEX ID_Postagem_UNIQUE (ID_Postagem ASC) VISIBLE,
  INDEX fk_Postagem_Ususario1_idx (Usuario_ID_Usuario ASC) VISIBLE,
  CONSTRAINT fk_Postagem_Ususario1
    FOREIGN KEY (Usuario_ID_Usuario)
      REFERENCES mydb.Usuario (ID_Usuario)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table mydb.Interacao

```

```

-----
CREATE TABLE IF NOT EXISTS mydb.Interacao (
  ID_Interacao INT NOT NULL,
  Tipo_Interacao VARCHAR(45) NOT NULL,
  Data_Hora_Interacao DATETIME NOT NULL,
  Postagem_ID_Postagem INT NOT NULL,
  Postagem_Usuario_ID_Usuario INT NOT NULL,
  PRIMARY KEY (ID_Interacao, Postagem_ID_Postagem,
  Postagem_Usuario_ID_Usuario),
  UNIQUE INDEX ID_Interacao_UNIQUE (ID_Interacao ASC) VISIBLE,
  INDEX fk_Interacao_Postagem1_idx (Postagem_ID_Postagem ASC,
  Postagem_Usuario_ID_Usuario ASC) VISIBLE,
  CONSTRAINT fk_Interacao_Postagem1
    FOREIGN KEY (Postagem_ID_Postagem , Postagem_Usuario_ID_Usuario)

```

```
REFERENCES mydb.Postagem (ID_Postagem , Usuario_ID_Usuario)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table mydb.Comentario
-- -----
```

```
CREATE TABLE IF NOT EXISTS mydb.Comentario (
  ID_Comentario INT NOT NULL,
  Conteudo_Comentario TEXT(650) NOT NULL,
  Data_Hora_Comentario DATETIME NOT NULL,
  Postagem_ID_Postagem INT NOT NULL,
  Postagem_Usuario_ID_Usuario INT NOT NULL,
  PRIMARY KEY (ID_Comentario, Postagem_ID_Postagem,
Postagem_Usuario_ID_Usuario),
  UNIQUE INDEX ID_Comentario_UNIQUE (ID_Comentario ASC) VISIBLE,
  INDEX fk_Comentario_Postagem1_idx (Postagem_ID_Postagem ASC,
Postagem_Usuario_ID_Usuario ASC) VISIBLE,
  CONSTRAINT fk_Comentario_Postagem1
  FOREIGN KEY (Postagem_ID_Postagem , Postagem_Usuario_ID_Usuario)
  REFERENCES mydb.Postagem (ID_Postagem , Usuario_ID_Usuario)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table mydb.Notificacao
-- -----
```

```
CREATE TABLE IF NOT EXISTS mydb.Notificacao (
  ID_Notificacao INT NOT NULL,
  Tipo_Notificacao VARCHAR(45) NOT NULL,
  Conteudo_Notificacao VARCHAR(70) NOT NULL,
  Data_Hora_Notificacao DATETIME NOT NULL,
  Interacao_ID_Interacao INT NOT NULL,
  IDdaPostagemInteragida INT NOT NULL,
  IDdoUsuarioQueInteragiu INT NOT NULL,
  Postagem_ID_Postagem INT NOT NULL,
  IDdoUsuarioQueFezPost INT NOT NULL,
  Comentario_ID_Comentario INT NOT NULL,
  IDdoPostComentado INT NOT NULL,
  IDdoUsuarioQueComentou INT NOT NULL,
  Mensagem_ID_Mensagem INT NOT NULL,
  PRIMARY KEY (ID_Notificacao, Interacao_ID_Interacao, IDdaPostagemInteragida,
IDdoUsuarioQueInteragiu, Postagem_ID_Postagem, IDdoUsuarioQueFezPost,
Comentario_ID_Comentario, IDdoPostComentado, IDdoUsuarioQueComentou,
Mensagem_ID_Mensagem),
  UNIQUE INDEX ID_Notificacao_UNIQUE (ID_Notificacao ASC) VISIBLE,
  INDEX fk_Notificacao_Interacao1_idx (Interacao_ID_Interacao ASC,
IDdaPostagemInteragida ASC, IDdoUsuarioQueInteragiu ASC) VISIBLE,
  INDEX fk_Notificacao_Postagem1_idx (Postagem_ID_Postagem ASC,
IDdoUsuarioQueFezPost ASC) VISIBLE,
  INDEX fk_Notificacao_Comentario1_idx (Comentario_ID_Comentario ASC,
IDdoPostComentado ASC, IDdoUsuarioQueComentou ASC) VISIBLE,
```

```

INDEX fk_Notificacao_Mensagem1_idx (Mensagem_ID_Mensagem ASC) VISIBLE,
CONSTRAINT fk_Notificacao_Interacao1
  FOREIGN KEY (Interacao_ID_Interacao , IDdaPostagemInteragida ,
IDdoUsuarioQueInteragiu)
  REFERENCES mydb.Interacao (ID_Interacao , Postagem_ID_Postagem ,
Postagem_Usuario_ID_Usuario)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT fk_Notificacao_Postagem1
  FOREIGN KEY (Postagem_ID_Postagem , IDdoUsuarioQueFezPost)
  REFERENCES mydb.Postagem (ID_Postagem , Usuario_ID_Usuario)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT fk_Notificacao_Comentario1
  FOREIGN KEY (Comentario_ID_Comentario , IDdoPostComentado ,
IDdoUsuarioQueComentou)
  REFERENCES mydb.Comentario (ID_Comentario , Postagem_ID_Postagem ,
Postagem_Usuario_ID_Usuario)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT fk_Notificacao_Mensagem1
  FOREIGN KEY (Mensagem_ID_Mensagem)
  REFERENCES mydb.Mensagem (ID_Mensagem)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table mydb.Usuario_has_Comunidade
-----

```

```

CREATE TABLE IF NOT EXISTS mydb.Usuario_has_Comunidade (
  Usuario_ID_Usuario INT NOT NULL,
  Comunidade_ID_Comunidade INT NOT NULL,
  PRIMARY KEY (Usuario_ID_Usuario, Comunidade_ID_Comunidade),
  INDEX fk_Usuario_has_Comunidade_Comunidade1_idx
(Comunidade_ID_Comunidade ASC) VISIBLE,
  INDEX fk_Usuario_has_Comunidade_Usuario_idx (Usuario_ID_Usuario ASC)
VISIBLE,
  CONSTRAINT fk_Usuario_has_Comunidade_Usuario
  FOREIGN KEY (Usuario_ID_Usuario)
  REFERENCES mydb.Usuario (ID_Usuario)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT fk_Usuario_has_Comunidade_Comunidade1
  FOREIGN KEY (Comunidade_ID_Comunidade)
  REFERENCES mydb.Comunidade (ID_Comunidade)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table mydb.Usuario_has_Mensagem
-----

```

```

CREATE TABLE IF NOT EXISTS mydb.Usuario_has_Mensagem (

```



```

    Usuario_ID_Usuario INT NOT NULL,
    Mensagem_ID_Mensagem INT NOT NULL,
    PRIMARY KEY (Usuario_ID_Usuario, Mensagem_ID_Mensagem),
    INDEX fk_Ususario_has_Mensagem_Mensagem1_idx (Mensagem_ID_Mensagem ASC)
    VISIBLE,
    INDEX fk_Ususario_has_Mensagem_Ususario1_idx (Usuario_ID_Usuario ASC)
    VISIBLE,
    CONSTRAINT fk_Ususario_has_Mensagem_Ususario1
    FOREIGN KEY (Usuario_ID_Usuario)
    REFERENCES mydb.Usuario (ID_Usuario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT fk_Ususario_has_Mensagem_Mensagem1
    FOREIGN KEY (Mensagem_ID_Mensagem)
    REFERENCES mydb.Mensagem (ID_Mensagem)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table mydb.Usuario_has_Interesse
-----

```

```

CREATE TABLE IF NOT EXISTS mydb.Usuario_has_Interesse (
    Usuario_ID_Usuario INT NOT NULL,
    Interesse_ID_Interesse INT NOT NULL,
    PRIMARY KEY (Usuario_ID_Usuario, Interesse_ID_Interesse),
    INDEX fk_Ususario_has_Interesse_Interesse1_idx (Interesse_ID_Interesse ASC)
    VISIBLE,
    INDEX fk_Ususario_has_Interesse_Ususario1_idx (Usuario_ID_Usuario ASC) VISIBLE,
    CONSTRAINT fk_Ususario_has_Interesse_Ususario1
    FOREIGN KEY (Usuario_ID_Usuario)
    REFERENCES mydb.Usuario (ID_Usuario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT fk_Ususario_has_Interesse_Interesse1
    FOREIGN KEY (Interesse_ID_Interesse)
    REFERENCES mydb.Interesse (ID_Interesse)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table mydb.Usuario_has_Usuario
-----

```

```

CREATE TABLE IF NOT EXISTS mydb.Usuario_has_Usuario (
    Usuario_ID_Usuario INT NOT NULL,
    Usuario_ID_Usuario1 INT NOT NULL,
    PRIMARY KEY (Usuario_ID_Usuario, Usuario_ID_Usuario1),
    INDEX fk_Usuario_has_Usuario_Usuario2_idx (Usuario_ID_Usuario1 ASC) VISIBLE,
    INDEX fk_Usuario_has_Usuario_Usuario1_idx (Usuario_ID_Usuario ASC) VISIBLE,
    CONSTRAINT fk_Usuario_has_Usuario_Usuario1
    FOREIGN KEY (Usuario_ID_Usuario)
    REFERENCES mydb.Usuario (ID_Usuario)
    ON DELETE NO ACTION

```

```
ON UPDATE NO ACTION,  
CONSTRAINT fk_Usuario_has_Usuario_Usuario2  
FOREIGN KEY (Usuario_ID_Usuario1)  
REFERENCES mydb.Usuario (ID_Usuario)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```