

Modelagem da Entidade Notificação

1. Entidade Usuário (Entidade Forte)

Primeiro, precisamos garantir que a entidade **Usuário** esteja bem definida, pois **Notificação** depende dela. A entidade **Usuário** teria uma **chave primária** que será referenciada pela entidade **Notificação**. Por exemplo:

Atributo	Tipo de Dado	Restrições
idUsuario	INT	PK , NOT NULL, AUTO_INCREMENT
nome	VARCHAR(255)	NOT NULL
email	VARCHAR(255)	UNIQUE, NOT NULL
<i>(outros atributos do usuário)</i>		

2. Entidade Notificação (Entidade Fraca)

A entidade **Notificação** será modelada da seguinte forma, incorporando a **chave estrangeira** que aponta para **Usuário**:

Atributo	Tipo de Dado	Restrições	Observações
idNotificacao	INT	PK , NOT NULL, AUTO_INCREMENT	Chave primária própria da notificação.
matriculaDestino	VARCHAR(50)	NOT NULL	Corresponde à matrícula do usuário que receberá a notificação. Será uma chave estrangeira para Usuário.matricula se essa for a PK ou atributo com constraint UNIQUE no usuário.
titulo	VARCHAR(255)	NOT NULL	Título da notificação.
mensagem	TEXT	NOT NULL	Conteúdo detalhado da notificação.

Atributo	Tipo de Dado	Restrições	Observações
tipo	VARCHAR(50)	NOT NULL	Categoria ou tipo da notificação (ex: “Alerta”, “Info”, “Promoção”).
idRelacionado	INT	NULLABLE	ID de alguma outra entidade relacionada (ex: idPedido, idChamado). É nulo se a notificação não se relaciona a uma entidade específica.
dataCriacao	DATETIME	NOT NULL, DEFAULT CUR-RENT_TIMES- TAMP	Data e hora em que a notificação foi criada.
lida	BOOLEAN	NOT NULL, DEFAULT FALSE	Indica se a notificação foi lida pelo usuário.

Chave Estrangeira e Dependência

A **dependência** de Notificação em relação a Usuário será estabelecida através da chave estrangeira. Existem algumas maneiras de implementar isso:

Opção A: Usando idUsuario como Chave Estrangeira (Recomendado)

Se a chave primária da tabela Usuário for idUsuario, e matriculaDestino na Notificação for na verdade uma referência ao idUsuario, a modelagem ficaria assim:

- matriculaDestino na tabela Notificação seria renomeado para idUsuario e se tornaria uma chave estrangeira (FK) para Usuário.idUsuario.

```
CREATE TABLE Usuario (
  idUsuario INT PRIMARY KEY AUTO_INCREMENT,
  matricula VARCHAR(50) UNIQUE NOT NULL, -- Matrícula do usuário, se for um atributo impor
```

```

    nome VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL
    -- Outros atributos do usuário
);

CREATE TABLE Notificacao (
    idNotificacao INT PRIMARY KEY AUTO_INCREMENT,
    idUsuario INT NOT NULL, -- Chave estrangeira para Usuario
    titulo VARCHAR(255) NOT NULL,
    mensagem TEXT NOT NULL,
    tipo VARCHAR(50) NOT NULL,
    idRelacionado INT,
    dataCriacao DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL,
    lida BOOLEAN DEFAULT FALSE NOT NULL,
    FOREIGN KEY (idUsuario) REFERENCES Usuario(idUsuario)
        ON DELETE CASCADE -- Se o usuário for deletado, suas notificações também serão.
        ON UPDATE CASCADE -- Se o id do usuário mudar, a referência na notificação também m
);

```

Vantagens: Clareza, performance em junções. **Consideração:** matriculaDestino do requisito original seria na verdade o idUsuario.

Opção B: Usando matricula como Chave Estrangeira (Se matricula for a PK ou UNIQUE em Usuário) Se a matricula for a chave primária de Usuário, ou se for um atributo **UNIQUE** com um índice, então poderíamos usá-la como chave estrangeira diretamente:

```

CREATE TABLE Usuario (
    matricula VARCHAR(50) PRIMARY KEY, -- Matrícula como PK
    idUsuario INT UNIQUE NOT NULL AUTO_INCREMENT, -- Adiciona um ID interno se necessário
    nome VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL
    -- Outros atributos do usuário
);

CREATE TABLE Notificacao (
    idNotificacao INT PRIMARY KEY AUTO_INCREMENT,
    matriculaDestino VARCHAR(50) NOT NULL, -- Chave estrangeira para Usuario.matricula
    titulo VARCHAR(255) NOT NULL,
    mensagem TEXT NOT NULL,
    tipo VARCHAR(50) NOT NULL,
    idRelacionado INT,
    dataCriacao DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL,
    lida BOOLEAN DEFAULT FALSE NOT NULL,
    FOREIGN KEY (matriculaDestino) REFERENCES Usuario(matricula)
        ON DELETE CASCADE
);

```

```

ON UPDATE CASCADE
);

```

Vantagens: Direta correspondência com o nome do atributo do requisito.
Considerações: A *matricula* precisa ser a chave primária ou ter uma restrição UNIQUE na tabela *Usuario* para ser usada como FK.

Escolha e Justificativa

Eu **optaria pela Opção A** (usando *idUsuario* como chave estrangeira na tabela *Notificação* e renomeando *matriculaDestino* para *idUsuario*).

Motivos:

1. **Consistência de Chaves Primárias:** É uma prática comum e recomendada ter um ID inteiro (AUTO_INCREMENT) como chave primária para a maioria das tabelas. Isso simplifica a gestão de relacionamentos e, em muitos casos, pode ser mais eficiente do que usar strings como chaves primárias.
2. **Manutenção e Flexibilidade:** Se o formato ou valor da matrícula do usuário precisar mudar no futuro, um *idUsuario* interno permanecerá estável, evitando a necessidade de atualizar todas as notificações.
3. **Clareza do Relacionamento:** Fica explícito que a notificação pertence a um *idUsuario* específico.

Se a *matricula* for um conceito de negócio muito forte e sempre único para o usuário, e a equipe de desenvolvimento preferir usá-la como a principal forma de identificar um usuário em todo o sistema, a **Opção B** também é válida, desde que *matricula* seja PRIMARY KEY ou UNIQUE na tabela *Usuario*.

Diagrama Simplificado (Modelo Relacional)

