**Entidades e Relacionamentos**

**1. Aluno**

* **ID** (Primary Key): Identificador único do aluno.
* **Nome**: Nome do aluno.
* **Matrícula**: Número da matrícula do aluno.

**2. Ocorrência**

* **ID** (Primary Key): Identificador único da ocorrência.
* **Descrição**: Descrição da ocorrência.
* **Tipo**: Tipo da ocorrência (por exemplo: "Agressão Verbal", "Agressão Física", "Agressão Patrimonial").
* **Status**: Status da ocorrência ("andamento" ou "encerrado").
* **Alunos Envolvidos**: Lista de IDs de alunos envolvidos na ocorrência (relacionamento com a entidade Aluno).

**3. Acompanhamento**

* **ID** (Primary Key): Identificador único do acompanhamento.
* **OcorrênciaID** (Foreign Key): Relacionamento com a ocorrência à qual este acompanhamento pertence.
* **AlunoID** (Foreign Key): Relacionamento com o aluno associado a este acompanhamento.
* **Status**: Status do acompanhamento ("andamento" ou "encerrado").

**4. Atualização**

* **ID** (Primary Key): Identificador único da atualização.
* **AcompanhamentoID** (Foreign Key): Relacionamento com o acompanhamento ao qual esta atualização pertence.
* **DataHora**: Data e hora em que a atualização foi registrada.
* **Texto**: Texto da atualização.
* **Categoria**: Categoria da atualização ("Acompanhamento" ou "Encaminhamento").
* **Tipo**: Tipo de atualização. Se for "Acompanhamento", os tipos são "Agressão Verbal", "Agressão Física", etc. Se for "Encaminhamento", os tipos são "Interno" ou "Externo".

**Relacionamentos**

1. **Aluno - Ocorrência**:
   * Relacionamento **Muitos-para-Muitos (N**

**)**.

* + Um aluno pode estar envolvido em várias ocorrências e uma ocorrência pode envolver vários alunos.
  + Será necessário criar uma **tabela de junção** para representar essa relação (por exemplo, "AlunosOcorrencias").

1. **Ocorrência - Acompanhamento**:
   * Relacionamento **Um-para-Muitos (1**

**)**.

* + Uma ocorrência pode ter vários acompanhamentos, cada um para um aluno específico envolvido na ocorrência.

1. **Aluno - Acompanhamento**:
   * Relacionamento **Um-para-Muitos (1**

**)**.

* + Um aluno pode ter vários acompanhamentos, mas cada acompanhamento pertence a um aluno específico.

1. **Acompanhamento - Atualização**:
   * Relacionamento **Um-para-Muitos (1**

**)**.

* + Um acompanhamento pode ter várias atualizações associadas.

**Diagrama ER (Entidade-Relacionamento)**

Diagrama, Desenho técnico

Descrição gerada automaticamente

 **ALUNO** e **OCORRENCIA** têm uma relação **M**

por meio da tabela associativa **ALUNOSOCORRENCIAS**.

 **OCORRENCIA** está relacionada a vários **ACOMPANHAMENTOS** e cada acompanhamento está relacionado a um **ALUNO** específico.

 **ACOMPANHAMENTO** está relacionado a várias **ATUALIZACOES**.

**Detalhamento Visual:**

1. **Aluno**:
   * **Relacionamento Muitos-para-Muitos com Ocorrência**: Um aluno pode estar em várias ocorrências, e uma ocorrência pode ter vários alunos. Para isso, precisamos de uma entidade associativa chamada **AlunosOcorrencias** para resolver esse relacionamento muitos-para-muitos.
2. **Ocorrência**:
   * Relacionada a vários **Acompanhamentos** (1

), cada um relacionado a um aluno específico.

* + A **tabela AlunosOcorrencias** é usada para mapear os alunos envolvidos em cada ocorrência.

1. **Acompanhamento**:
   * Cada **Acompanhamento** refere-se a um aluno específico em uma ocorrência específica.
   * Um **Acompanhamento** pode ter várias **Atualizações**, cada uma registrada ao longo do tempo.
2. **Atualização**:
   * Cada atualização se relaciona com apenas um **Acompanhamento**.

**Modelo Entidade-Relacionamento Explicado**

* **Aluno** e **Ocorrência** compartilham uma relação **N**

, o que significa que a ocorrência pode envolver vários alunos e cada aluno pode estar em várias ocorrências. Isso é resolvido por uma **tabela associativa** chamada **AlunosOcorrencias**.

* A **Ocorrência** tem uma relação **1**

com **Acompanhamento**. Para cada ocorrência, há vários acompanhamentos (um para cada aluno envolvido). Um acompanhamento se refere a um aluno específico e uma ocorrência específica.

* O **Acompanhamento** está relacionado com a **Atualização** por um relacionamento **1**

, pois um acompanhamento pode ter várias atualizações ao longo do tempo.

**Descrição dos Atributos:**

* Todos os IDs (ID do aluno, ID da ocorrência, etc.) são **chaves primárias** e servem para identificar de forma única cada entidade.
* **AlunosOcorrencias**: é uma tabela de junção contendo **AlunoID** e **OcorrênciaID**, que juntos formam a **chave primária composta** para relacionar alunos e ocorrências.

**Sugestão de Implementação para Banco de Dados**

Aqui está um exemplo de como essas entidades poderiam ser representadas em um banco de dados relacional (como MySQL ou PostgreSQL):

sql

Copiar código

CREATE TABLE Aluno (

ID SERIAL PRIMARY KEY,

Nome VARCHAR(100),

Matricula INT

);

CREATE TABLE Ocorrencia (

ID SERIAL PRIMARY KEY,

Descricao TEXT,

Tipo VARCHAR(50),

Status VARCHAR(20)

);

CREATE TABLE AlunosOcorrencias (

AlunoID INT,

OcorrenciaID INT,

PRIMARY KEY (AlunoID, OcorrenciaID),

FOREIGN KEY (AlunoID) REFERENCES Aluno(ID),

FOREIGN KEY (OcorrenciaID) REFERENCES Ocorrencia(ID)

);

CREATE TABLE Acompanhamento (

ID SERIAL PRIMARY KEY,

OcorrenciaID INT,

AlunoID INT,

Status VARCHAR(20),

FOREIGN KEY (OcorrenciaID) REFERENCES Ocorrencia(ID),

FOREIGN KEY (AlunoID) REFERENCES Aluno(ID)

);

CREATE TABLE Atualizacao (

ID SERIAL PRIMARY KEY,

AcompanhamentoID INT,

DataHora TIMESTAMP,

Texto TEXT,

Categoria VARCHAR(20),

Tipo VARCHAR(50),

FOREIGN KEY (AcompanhamentoID) REFERENCES Acompanhamento(ID)

);

**Resumo**

* **Aluno**, **Ocorrência**, **Acompanhamento**, e **Atualização** são as entidades principais.
* A relação entre **Aluno** e **Ocorrência** é **M**

e exige uma tabela de junção (**AlunosOcorrencias**).

* **Ocorrência** está relacionada a vários **Acompanhamentos** e cada **Acompanhamento** está relacionado a um **Aluno** específico.
* **Acompanhamento** pode ter várias **Atualizações**, que são registradas ao longo do tempo.

MERMAID LIVE EDITOR

**erDiagram**

    ALUNO **{**

        int ID

        string Nome

        int Matricula

**}**

    OCORRENCIA **{**

        int ID

        string Descricao

        string Tipo

        string Status

**}**

    ALUNOSOCORRENCIAS **{**

        int AlunoID

        int OcorrenciaID

**}**

    ACOMPANHAMENTO **{**

        int ID

        int OcorrenciaID

        int AlunoID

        string Status

**}**

    ATUALIZACAO **{**

        int ID

        int AcompanhamentoID

        datetime DataHora

        string Texto

        string Categoria

        string Tipo

**}**

    ALUNO **||--o{** ALUNOSOCORRENCIAS **:** "tem"

    OCORRENCIA **||--o{** ALUNOSOCORRENCIAS **:** "tem"

    OCORRENCIA **||--o|** ACOMPANHAMENTO **:** "tem"

    ALUNO **||--o|** ACOMPANHAMENTO **:** "relacionado a"

    ACOMPANHAMENTO **||--o|** ATUALIZACAO **:** "tem"

PROMPT PARA IA

**Prompt para Restabelecer a Conversa:**

**Prompt para Restabelecer Conversa:**

"Olá, estou desenvolvendo um projeto em **Node.js** com **Vite**, que se trata de um sistema de **cadastro de ocorrências escolares** envolvendo alunos. Até agora, fiz as seguintes implementações e melhorias com a ajuda do ChatGPT:

1. **Backend com Express.js**:
   * **Endpoints** para gerenciar ocorrências, alunos, acompanhamentos, e atualizações.
   * **Relação entre entidades**:
     + **Alunos**: identificados por nome e matrícula.
     + **Ocorrências**: registram situações envolvendo alunos, com uma descrição, tipo e status.
     + **Acompanhamento**: cada ocorrência tem acompanhamentos específicos por aluno.
     + **Atualizações**: um acompanhamento pode ter várias atualizações, categorizadas como "Acompanhamento" ou "Encaminhamento".
2. **Frontend com React (usando MUI - Material UI)**:
   * **Formulário de Cadastro de Ocorrências**: permite selecionar alunos e cadastrar ocorrências com tipo e descrição.
   * **Painel de Acompanhamento por Aluno**: para cada aluno, é possível visualizar as ocorrências nas quais ele está envolvido.
   * **Funcionalidade de Atualizações**: cada acompanhamento pode ter atualizações adicionais que podem ser categorizadas como "Acompanhamento" ou "Encaminhamento", sendo:
     + **Encaminhamento** dividido em **Interno** ou **Externo**.
     + **Acompanhamento** com tipos como "Agressão Verbal", "Agressão Física", etc.
   * Foi implementada uma lógica onde, ao selecionar uma categoria de atualização, o campo "Tipo" é exibido de forma condicional para o usuário.
3. **Entidade Relacionamento (ER)**:
   * Construímos um **Modelo Entidade-Relacionamento** para mapear o sistema:
     + Entidades principais: **Aluno**, **Ocorrência**, **Acompanhamento**, e **Atualização**.
     + Relacionamentos:
       - **Aluno** e **Ocorrência**: Muitos para muitos.
       - **Ocorrência** e **Acompanhamento**: Um para muitos.
       - **Acompanhamento** e **Atualização**: Um para muitos.
     + Foi feita uma representação do **diagrama ER** em formato **Mermaid**, que detalha como essas entidades se relacionam.
4. **Requisições HTTP**:
   * O sistema atual faz chamadas **fetch** para gerenciar atualizações, mudanças de status dos acompanhamentos e ocorrências.
   * O fluxo de cadastro de atualizações foi detalhado, onde o botão "Salvar Atualização" só é liberado se todos os campos obrigatórios forem preenchidos corretamente.
5. **Diagrama Mermaid**:
   * Criamos um diagrama para o modelo ER usando Mermaid, que ajuda a visualizar o relacionamento entre as entidades do projeto. Aqui está o diagrama:

mermaid

Copiar código

erDiagram

ALUNO {

int ID

string Nome

int Matricula

}

OCORRENCIA {

int ID

string Descricao

string Tipo

string Status

}

ALUNOSOCORRENCIAS {

int AlunoID

int OcorrenciaID

}

ACOMPANHAMENTO {

int ID

int OcorrenciaID

int AlunoID

string Status

}

ATUALIZACAO {

int ID

int AcompanhamentoID

datetime DataHora

string Texto

string Categoria

string Tipo

}

ALUNO ||--o{ ALUNOSOCORRENCIAS : "tem"

OCORRENCIA ||--o{ ALUNOSOCORRENCIAS : "tem"

OCORRENCIA ||--o| ACOMPANHAMENTO : "tem"

ALUNO ||--o| ACOMPANHAMENTO : "relacionado a"

ACOMPANHAMENTO ||--o| ATUALIZACAO : "tem"

Gostaria de continuar com o desenvolvimento a partir daqui. Precisamos de melhorias no painel de acompanhamento dos alunos, ajustes no backend ou qualquer outra melhoria que possa ser sugerida. Por favor, considere os pontos acima e me ajude a continuar o desenvolvimento do sistema."