

# Sistema de Gerenciamento de Cursos (V4 COMPANY)

Esse Documento contém regras do sistema, schema do Banco de Dados e considerações importantes sobre o sistema .

## Descrição do Sistema

O sistema foi construído para gerenciar uma plataforma de cursos, o usuário pode criar uma conta e fazer login, gerando assim um JWT(JsonWebToken) que sera responsável por autenticar o usuário.

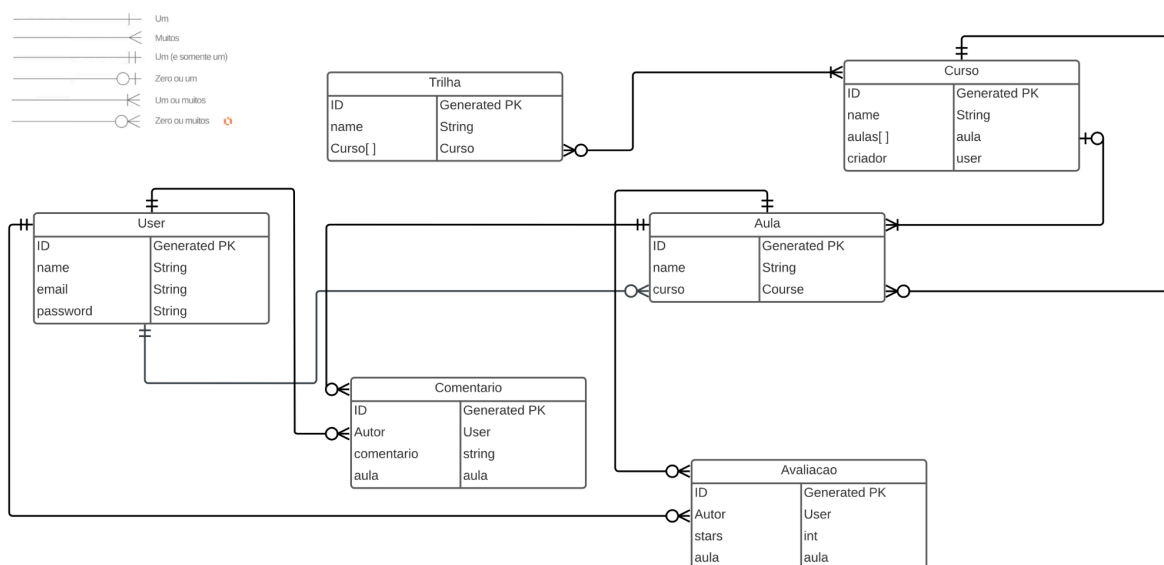
O usuário pode criar trilhas de ensino, essas trilhas podem conter inúmeros cursos, cada curso contém aulas, cada aula por sua vez, possui ou não, comentários e avaliações(de 0 a 5 estrelas).

Cada Aula obrigatoriamente está associada a um curso, por isso não é possível criar aulas sem criar nenhum curso.

Cada Aula possui um link único.

O mesmo curso pode estar em diferentes trilhas.

## Schema do Banco de Dados



## Como rodar

Instalar dependências e popular users.

```
npm i
npm run populatewUsers
```

Agora que você já tem usuários, pegue o JWT de algum deles (usando a rota de login que devolve o token do usuário logado) e preencha o .ENV

### Criar .ENV

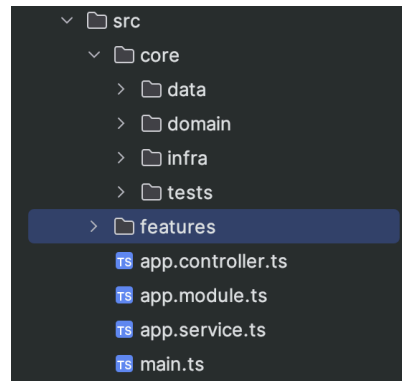
```
JWT_SECRET= SEGREDO JWT (preferencialmente string randômica)

DB_USER=
DB_PASSWORD=
DB_NAME=
PORT=
HOSTNAME=
JWT_TOKEN=SEU TOKEN
```

### Popular Cursos e Aulas (opcional)

```
npm run populatewCursos
npm run populatewAulas
```

# Arquitetura do Sistema



O sistema é construído com uma arquitetura MVC, ou seja, cada módulo possui controller e service.

## CORE

> **domain:** Possui a declaração da classe Custom Response, essa classe é sempre instanciada nos controllers, possui a estrutura da resposta padrão.

> **infra:** Contém os middlewares, nesse caso, contém apenas o middleware de autenticação do token JWT.

> **data:** Contém a declaração das entidades e seus respectivos providers.

> **test:** Contém arquivos que populam o banco de dados automaticamente. Simulando requisições.

## FEATURES

> **TODOS OS MÓDULOS DA APLICAÇÃO.**

**O controller** recebe as requisições, referenciam DTO's (Data Transfer Object), cada DTO é validado por um PIPE, para conferir se todas as informações da requisição estão no formato correto.

**O service** é a única camada que se comunica com o Banco de Dados.

**O module** é a camada onde se faz declarações aos providers das entidades, indica controllers e services desse módulo

.