

Bem vindo ao **Sistema Gestão Escola Desafio**.

Esta ferramenta permite o cadastro de cursos e alunos, bem como a consulta, a edição e a deleção de ambos. Além disso, é possível vincular alunos a um curso, formando uma turma, através da realização de matrícula.

CONTEÚDO:

- 1- Endpoints do SGED;
- 2- Configurando Banco de Dados (Postgres ou H2);
- 3- Como usar o SGED.

ENDPOINTS DO SGED:

1. Endpoints de Alunos:
 - 1.1. Cadastrar aluno;
 - 1.2. Consultar todos os alunos cadastrados;
 - 1.3. Consultar aluno por número de cadastro;
 - 1.4. Atualizar todos os campos de um cadastro de aluno;
 - 1.5. Atualizar um campo específico de um cadastro de aluno;
 - 1.6. Apagar cadastro de aluno*.
2. Endpoints de Cursos
 - 2.1. Cadastrar curso;
 - 2.2. Consultar todos os cursos cadastrados;
 - 2.3. Consultar curso por número de cadastro;
 - 2.4. Atualizar todos os campos de um cadastro de curso;
 - 2.5. Atualizar um campo específico de um cadastro de curso;
 - 2.6. Apagar cadastro de curso*.
3. Endpoints de Matrículas
 - 3.1. Realizar matrícula;
 - 3.2. Consultar todas as matrículas realizadas;
 - 3.3. Apagar matrícula;
 - 3.4. Consultar alunos matriculados em um curso;
 - 3.5. Consultar os cursos em que um aluno foi matriculado;

***NOTA: na deleção de um aluno ou curso, todas as matrículas relacionadas são apagadas.**

BANCO DE DADOS:

Esta API está atualmente configurada para o banco de dados Postgres de nome db_TI360.

Contém o endereçamento (datasource) postgres com username postgres e password 123456 no arquivo application.yml.

```
spring:
  datasource:
    url: jdbc:postgresql://localhost:5432/db_TI360
    username: postgres
    password: 123456
    driver-class-name: org.postgresql.Driver
```

Contém a dependências Postgres no arquivo pom.xml:

```
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>42.3.1</version>
</dependency>
```

A título de testes, você pode usar H2 como banco de dados local, informando as seguintes linhas no arquivo application.yml:

```
spring:
  datasource:
    url: jdbc:h2:mem:testdb
    username: sa
    password:
  h2:
    console:
      enabled: true
      path: /h2-console
```

e a seguinte dependência no arquivo pom.xml:

```
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
```

COMO USAR:

Crie um ou mais alunos. Para isso, informe nome completo, idade (somente algarismos arábicos), e-mail (no formato nome @ domínio.extensão) e CEP (somente algarismos, sem traço):

The screenshot shows a REST client interface for a POST request to the endpoint `/alunos/novoAluno`. The header bar indicates the method is **POST** and the endpoint is `/alunos/novoAluno` with the description "Cadastra um novo aluno". Below the header, there are two tabs: **Parameters** and **Request body**. The **Parameters** tab is active, showing "No parameters". To the right of the tabs are **Cancel** and **Reset** buttons. Below the tabs, the **Request body** is set to `application/json` and is marked as **required**. The request body content is a JSON object:

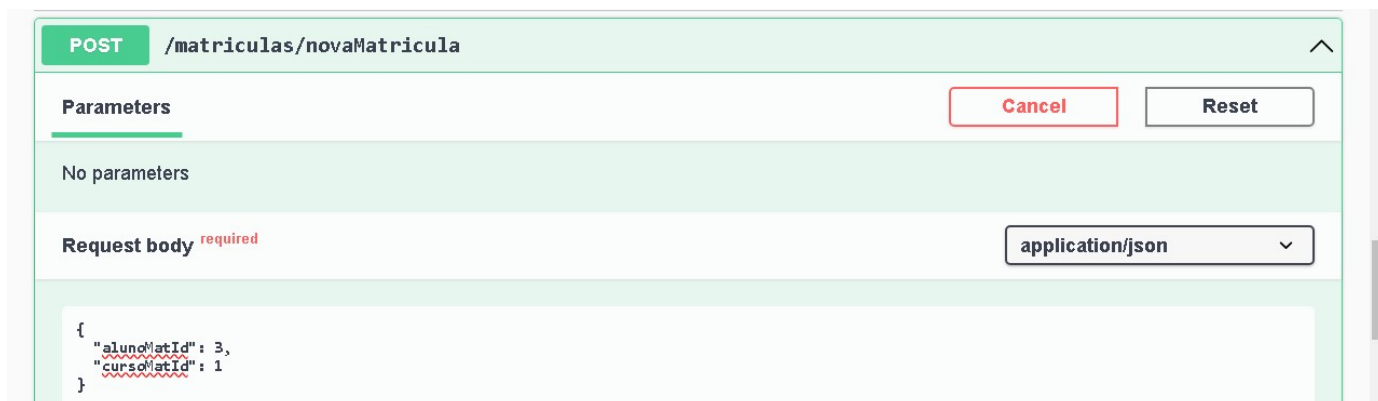
```
{  "nomeAluno": "Wesley Nazário Borges",  "idadeAluno": 35,  "emailAluno": "wfborges@ti360experience.com",  "cep": "09710240"}
```

Em seguida, crie um ou mais cursos. Para isso, basta informar o nome do curso:

The screenshot shows a REST client interface for a POST request to the endpoint `/cursos/novoCurso`. The header bar indicates the method is **POST** and the endpoint is `/cursos/novoCurso` with the description "Cadastra um novo curso". Below the header, there are two tabs: **Parameters** and **Request body**. The **Parameters** tab is active, showing "No parameters". To the right of the tabs are **Cancel** and **Reset** buttons. Below the tabs, the **Request body** is set to `application/json` and is marked as **required**. The request body content is a JSON object:

```
{  "nomeCurso": "TI 360 - Start"}
```

Você fará a vinculação de um aluno com um curso através de uma Matrícula. Para isso, informe o código do aluno cadastrado e o código do curso:



The screenshot shows a REST client interface for a POST request to the endpoint `/matriculas/novaMatricula`. The request body is a JSON object with the following structure:

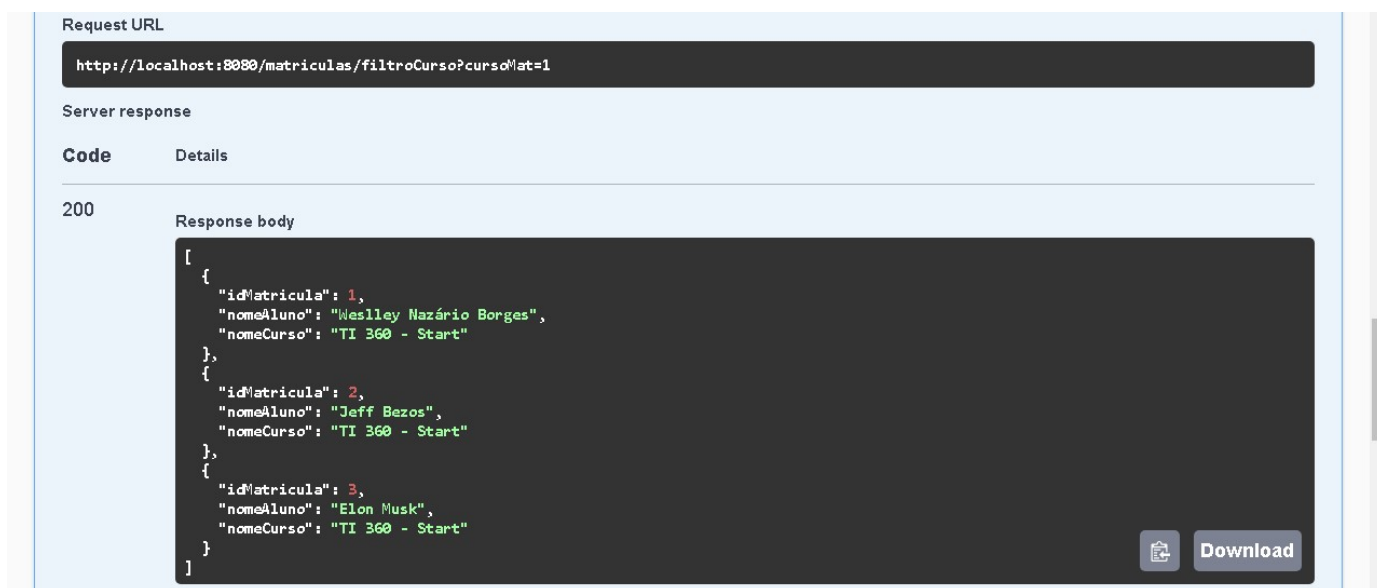
```
{  "alunoMatId": 3,  "cursoMatId": 1}
```

The interface includes a "Parameters" section with "No parameters", a "Request body" section with a dropdown menu set to "application/json", and "Cancel" and "Reset" buttons.

Caso tente realizar uma matrícula sem um aluno ou sem um curso previamente cadastrados, retornará um erro.

Uma vez cadastrados os alunos, os cursos e realizadas as matrículas, basta informar o código de cada objeto para consulta, edição e deleção.

Para a consulta de alunos em um determinado curso, basta informar o código do curso, como na URL da imagem:



The screenshot shows a REST client interface for a GET request to the endpoint `http://localhost:8080/matriculas/filtroCurso?cursoMat=1`. The server response is a 200 status code with a JSON response body containing an array of three objects:

```
[  {    "idMatricula": 1,    "nomeAluno": "Wesley Nazário Borges",    "nomeCurso": "TI 360 - Start"  },  {    "idMatricula": 2,    "nomeAluno": "Jeff Bezos",    "nomeCurso": "TI 360 - Start"  },  {    "idMatricula": 3,    "nomeAluno": "Elon Musk",    "nomeCurso": "TI 360 - Start"  }]
```

The interface includes a "Request URL" section, a "Server response" section, and a "Response body" section. There is also a "Download" button at the bottom right of the response body.

Para mais informações, envie um e-mail para: suporteSGED@acme.com

Agradecemos a sua confiança!