



Desafio Backend

node.js

ESCOPO

Com o encerramento da pandemia, as escolas e universidades precisarão controlar a alocação de salas para seus professores e alunos.

Pensando nisso, decidimos ajudar desenvolvendo uma API em Node.js utilizando o AdonisJS (<https://adonisjs.com>). Levantamos os seguintes requisitos de sistema:

REQUISITOS FUNCIONAIS

- RF01: Permitir que aluno se cadastre na aplicação
- RF02: Permitir que aluno edite seus dados de cadastro
- RF03: Permitir que aluno exclua seus dados de cadastro
- RF04: Permitir que aluno consulte seus dados de cadastro
- RF05: Permitir que professor se cadastre na aplicação
- RF06: Permitir que professor edite seus dados de cadastro
- RF07: Permitir que professor exclua seus dados de cadastro
- RF08: Permitir que professor consulte seus dados de cadastro
- RF09: Permitir que professor cadastre uma nova sala
- RF10: Permitir que professor edite os dados de uma sala
- RF11: Permitir que professor exclua os dados de uma sala
- RF12: Permitir que professor consulte os dados de uma sala
- RF13: Permitir que professor aloque um aluno em uma sala



RF14: Permitir que professor remova o aluno de uma sala

RF15: Permitir que professor consulte todos os alunos de uma sala

RF16: Permitir que aluno consulte todas as salas que deverá comparecer

REGRAS DE NEGÓCIO

Para cada requisito funcional, teremos uma ou mais regras de negócio que deverão ser atendidas:

RN01 (RF01): Deve ser coletado do aluno os seguintes dados: Nome, e-mail, matrícula, data de nascimento.

RN02 (RF05): Deve ser coletado do professor os seguintes dados: Nome, e-mail, matrícula, data de nascimento.

RN03 (RF09): Deve ser coletado da sala: Número da sala, capacidade de alunos, disponibilidade (Se pode alocar aluno ou não).

RN03 (RF13): A sala não pode possuir o mesmo aluno mais de uma vez.

RN04 (RF13): A sala não pode exceder sua capacidade de alunos.

RN05 (RF13): O professor não poderá alocar um aluno para uma sala que não tenha sido criada por ele.

RN06 (RF16): Deverá ser retornado: Nome do aluno, array de objetos com nome do professor e o número da sala.

EXIGÊNCIAS

- Um arquivo exportado em json (do insomnia) com todas as rotas da aplicação.
- Desenvolva a API em Node.js, utilizando o AdonisJS (<https://adonisjs.com>)
- Nos envie um arquivo exportado em json (do insomnia) com todas as rotas da aplicação
- Entregue o link do github com o projeto commitado.



PRAZO DE ENTREGA

O desafio deve ser enviado com as exigências descritas neste documento, no prazo de 5 dias corridos a partir do recebimento.

A entrega deverá ser feita para o e-mail rh@clicksoft.com.br, com o título “DESAFIO NODE.JS”.

BOA SORTE!!!