

# ESCOLA DE IDIOMAS

**PROJETO FINAL BACKEND - FIRJAN SENAI**

GRUPO 4

## O PROJETO

O PROJETO ESCOLA DE IDIOMAS REPRESENTA A CONCLUSÃO DAS DISCIPLINAS DO CURSO DE BACKEND - FIRJAN SENAI.

DURANTE A ELABORAÇÃO DO PROJETO, FORAM APLICADOS CONCEITOS DE LÓGICA DE PROGRAMAÇÃO, UML, JAVA, BANCO DE DADOS, SPRINGBOOT E API REST.

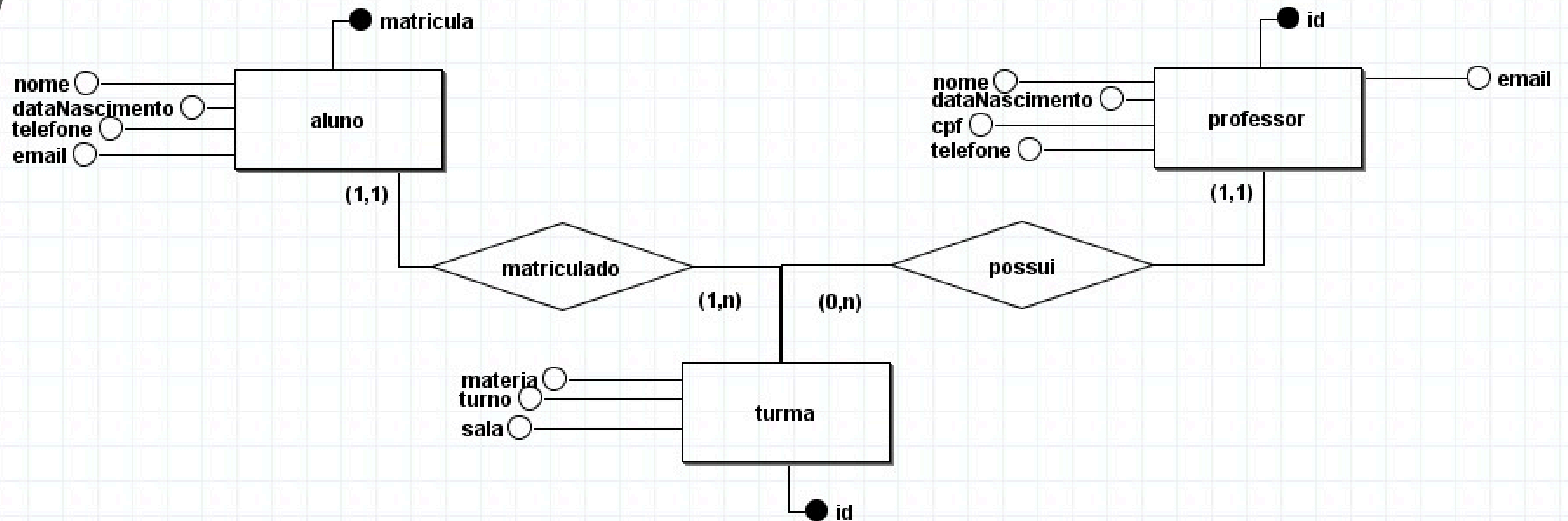
# ENTIDADES PRINCIPAIS E SUAS RELAÇÕES

UTILIZAMOS TRÊS TABELAS PRINCIPAIS:

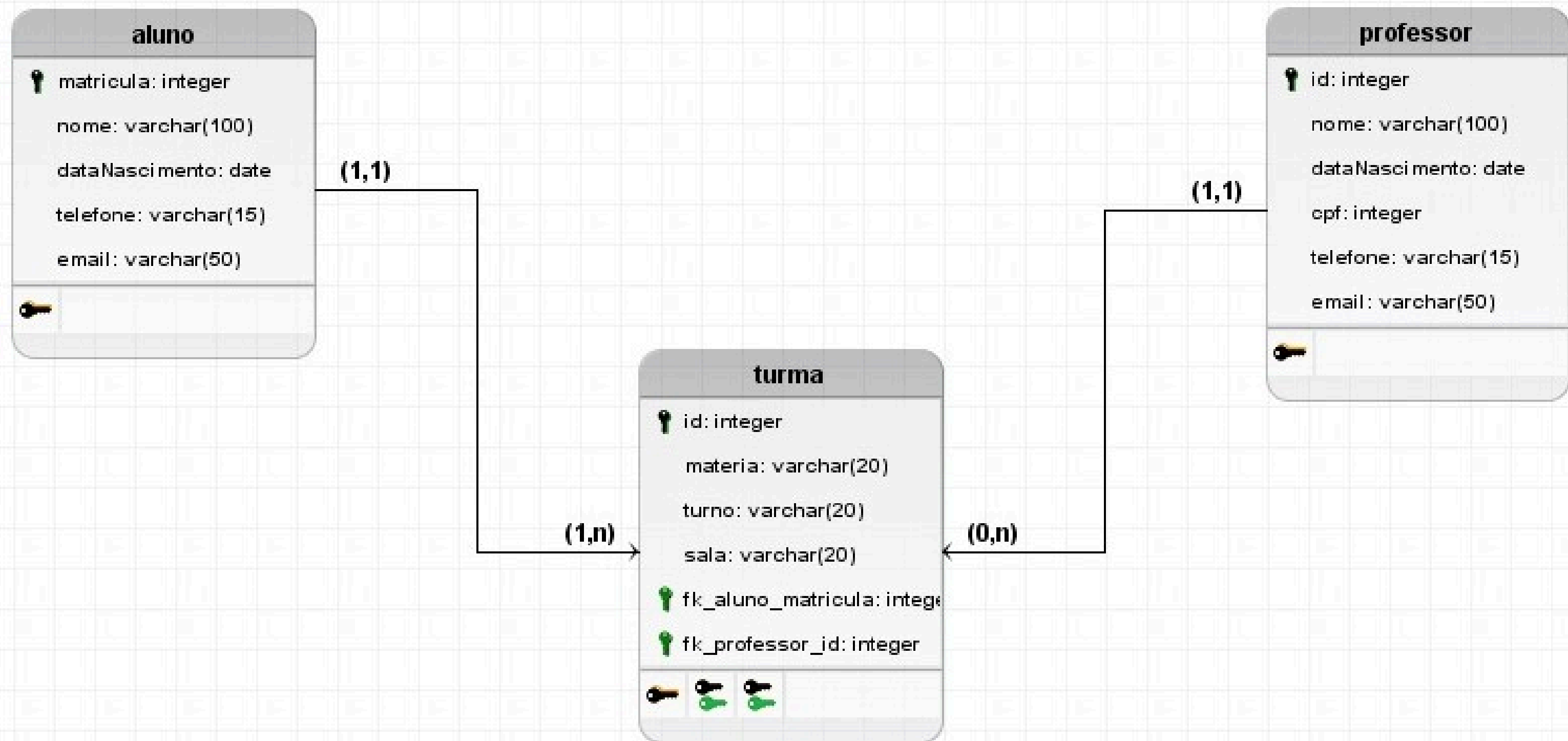
ALUNO;  
PROFESSOR;  
TURMA.

. ESSAS TABELAS COMPÕEM UM RELACIONAMENTO DE ENTIDADE ASSOCIATIVA.

A TABELA "ALUNO" ARMAZENA INFORMAÇÕES SOBRE CADA ESTUDANTE, COMO MATRÍCULA (SUA CHAVE PRIMÁRIA), NOME, DATA DE NASCIMENTO E CONTATO. A TABELA "PROFESSOR" CONTÉM DADOS SOBRE OS DOCENTES, INCLUINDO ID (SUA CHAVE PRIMÁRIA), SEUS DADOS PESSOAIS E DE CONTATO. JÁ A TABELA "TURMA" REGISTRA AS TURMAS DISPONÍVEIS, COM DETALHES SOBRE TURNO, DISCIPLINA, OS PROFESSORES RESPONSÁVEIS E OS ALUNOS MATRICULADOS. TAMBÉM É A TABELA "TURMA" QUE FAZ A LIGAÇÃO ENTRE ALUNOS E PROFESSORES.




modelo conceitual



modelo lógico




## professor

 id: integer  
nome: varchar(100)  
dataNascimento: date  
cpf: integer  
telefone: varchar(15)  
email: varchar(50)



- identificação professor

## turma

 id: integer  
matéria: varchar(20)  
turno: varchar(20)  
sala: varchar(20)  
 fk\_aluno\_matricula: integer  
 fk\_professor\_id: integer



- identificação turma
- chaves estrangeiras

## aluno

 matricula: integer  
nome: varchar(100)  
dataNascimento: date  
telefone: varchar(15)  
email: varchar(50)



- identificação aluno (matrícula)

Home Workspaces API Network Search Postman Invite Upgrade

My Workspace New Import Overview GET Post data

REST API basics: CRUD, test & variable / Post data Save Share

GET http://localhost:8080/professor Send

Params Authorization Headers (6) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Body Cookies Headers (5) Test Results (1/1) Status: 200 OK Time: 2.76 s Size: 1.06 KB Save as example

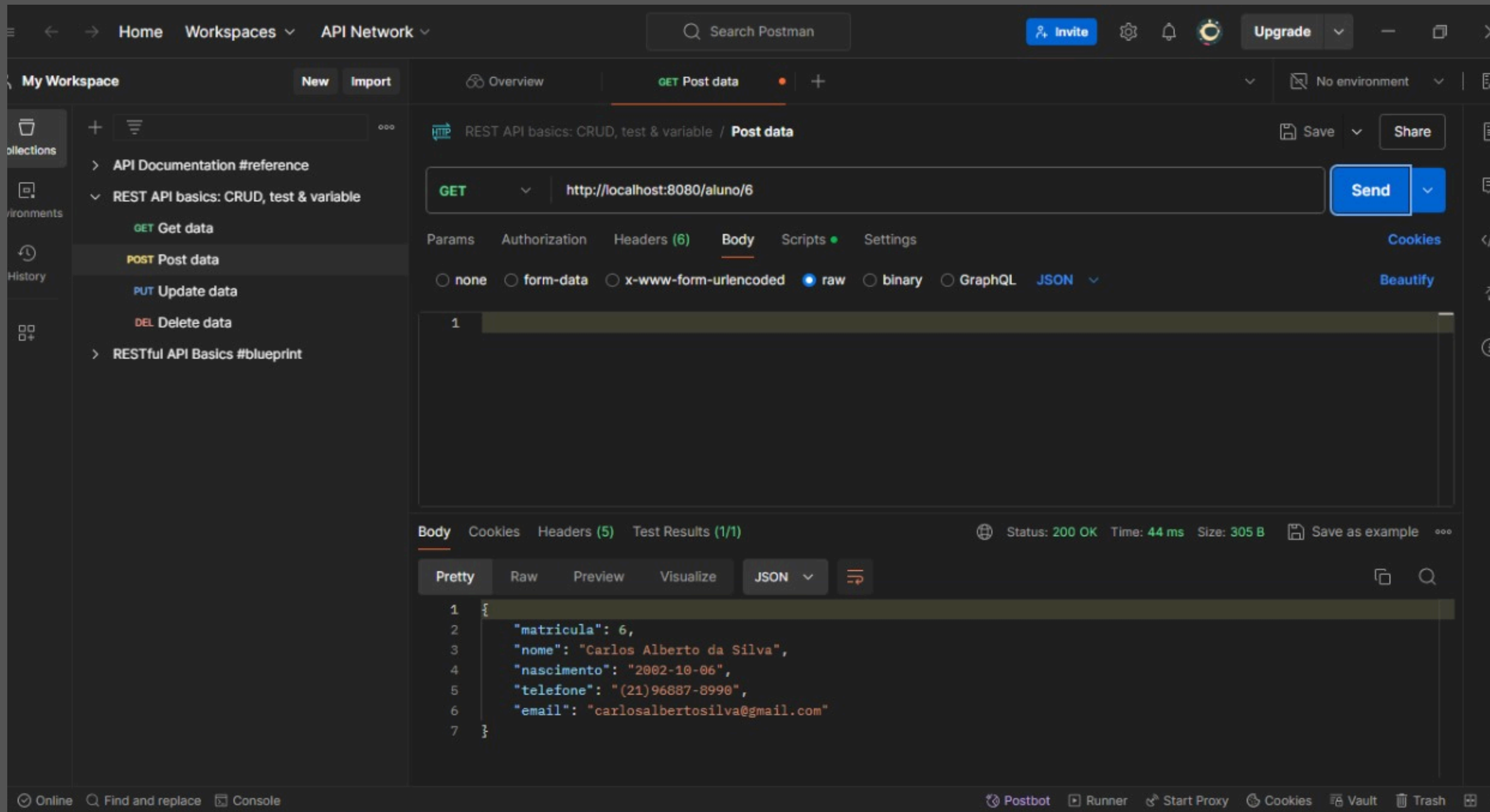
Pretty Raw Preview Visualize JSON

```
2 {
3   "id": 1,
4   "nome": "Mario Alcântara Soares",
5   "nascimento": "1978-10-02",
6   "cpf": "784152025-87",
7   "telefone": "(21)90212-8400",
8   "email": "marioalsoares@gmail.com"
9 },
10 {
11   "id": 2,
12   "nome": "Mario Alcântara Soares",
13   "nascimento": "1978-10-02",
14   "cpf": "784152025-87",
15   "telefone": "(21)90212-8400",
16   "email": "marioalsoares@gmail.com"
17 }
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

método GET (tabela professor)





método GET (tabela aluno): chamando aluno pelo ID.



Home Workspaces API Network Search Postman Invite Upgrade

My Workspace New Import Overview POST Post data No environment

Collections + API Documentation #reference REST API basics: CRUD, test & variable

Environments GET Get data POST Post data PUT Update data DEL Delete data RESTful API Basics #blueprint

History

REST API basics: CRUD, test & variable / Post data Save Share

POST http://localhost:8080/aluno Send

Params Authorization Headers (8) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "matricula": 64,
3   "nome": "Junior Andrade da Silva",
4   "nascimento": "2001-10-10",
5   "telefone": "(21)96787-7590",
6   "email": "juniorsilva@gmail.com"
7 }
8
```

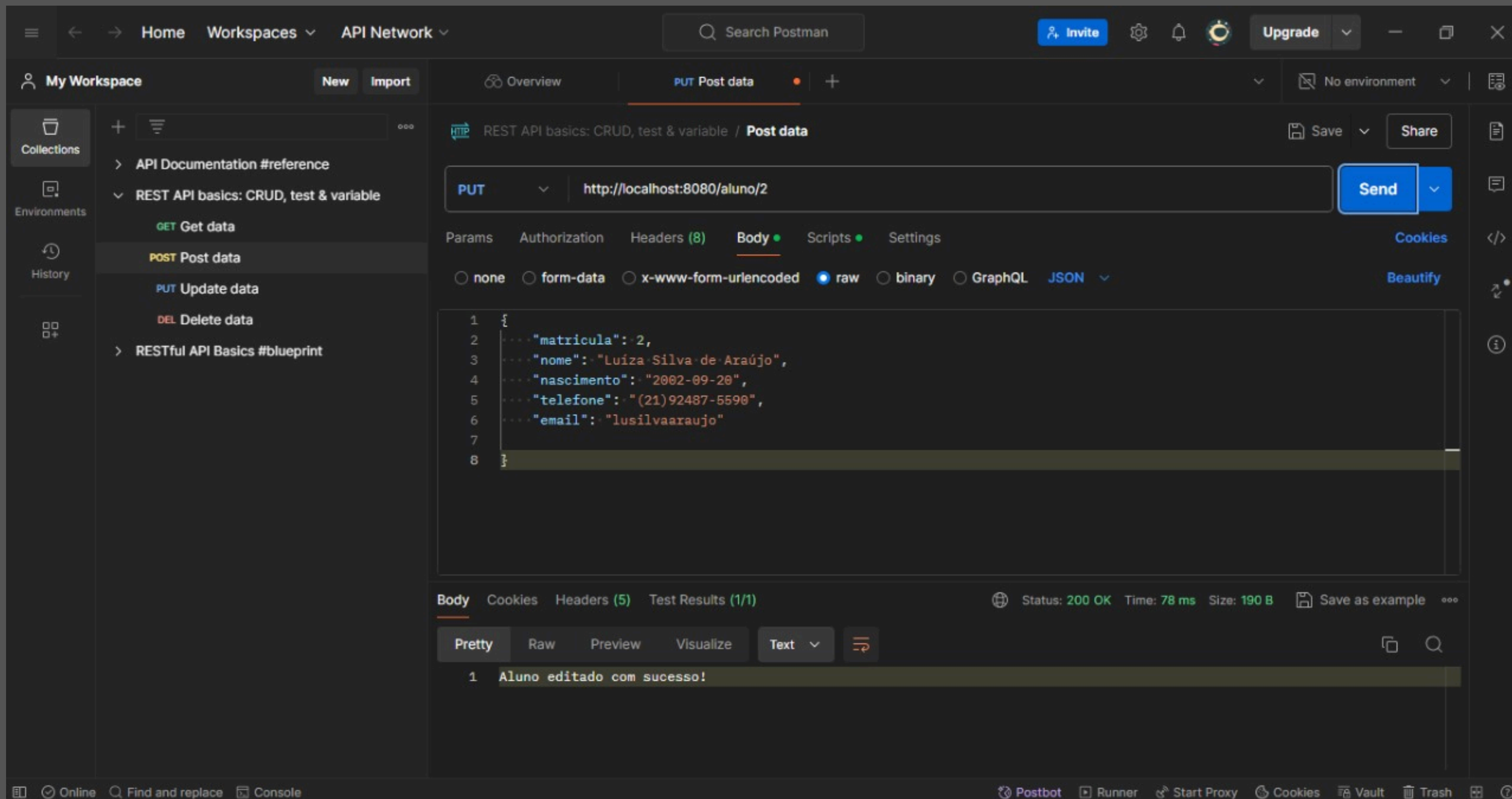
Body Cookies Headers (5) Test Results (1/1) Status: 200 OK Time: 289 ms Size: 193 B Save as example

Pretty Raw Preview Visualize Text

```
1 Aluno adicionado com Sucesso!
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

método POST: adicionando aluno



método PUT: editando aluno

# GRUPO 4

## INTEGRANTES

ERICK DE ANDRADE NINA

JOÃO PAULO DE LIMA BALBINO DOS SANTOS

LETÍCIA ARAUJO MONTEIRO DE BARROS

PAULO SÉRGIO CARNEIRO DE ASSIS NETO