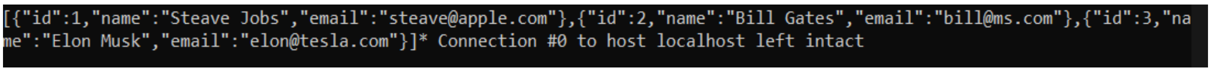
João Vitor Afonso Pereira - 11911BCC037

7)

Utilizando o comando:

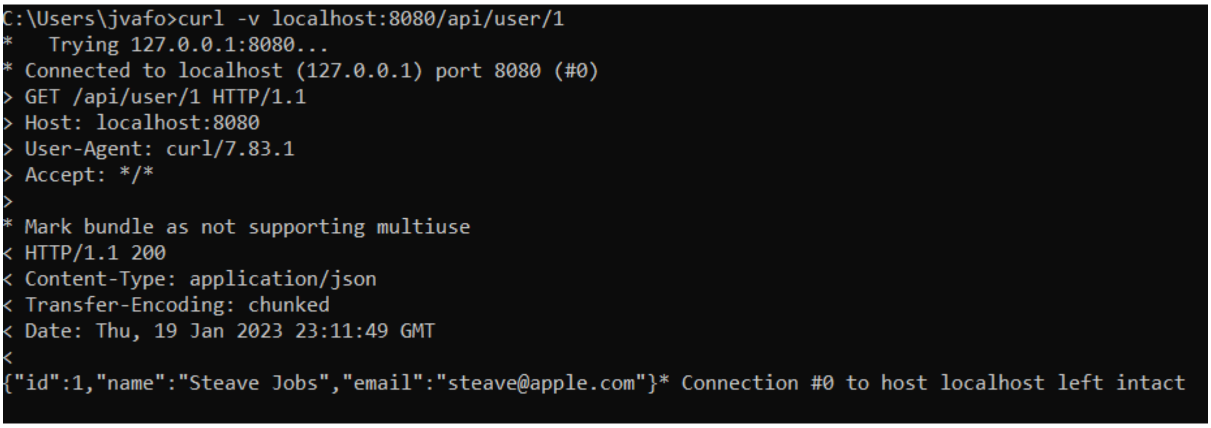
curl -v localhost:8080/api/user , tem-se a resposta



```
[{"id":1,"name":"Steave Jobs","email":"steave@apple.com"},{"id":2,"name":"Bill Gates","email":"bill@ms.com"},{"id":3,"na
me":"Elon Musk","email":"elon@tesla.com"}]* Connection #0 to host localhost left intact
```

Utilizando o comando:

curl -v localhost:8080/api/user/1 , tem-se a resposta

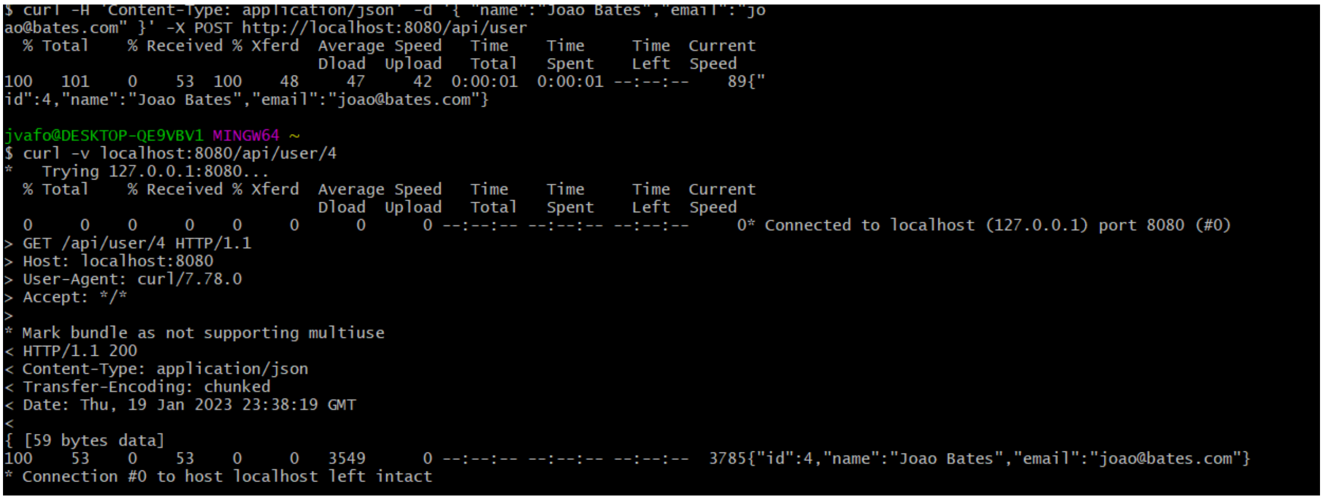


```
C:\Users\jvafo>curl -v localhost:8080/api/user/1
*   Trying 127.0.0.1:8080...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /api/user/1 HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.83.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200
< Content-Type: application/json
< Transfer-Encoding: chunked
< Date: Thu, 19 Jan 2023 23:11:49 GMT
<
{"id":1,"name":"Steave Jobs","email":"steave@apple.com"}* Connection #0 to host localhost left intact
```

8) Utilizando o comando de POST um puco alterado

curl -H 'Content-Type: application/json' -d '{ "name":"Joao Bates","email":"joao@bates.com" }' -X POST 'http://localhost:8080/api/user'

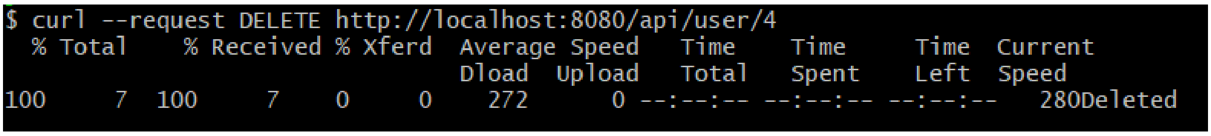
E o metodo Get



```
$ curl -H 'Content-Type: application/json' -d '{ "name":"Joao Bates","email":"jo
ao@bates.com" }' -X POST http://localhost:8080/api/user
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   101    0    53  100    48     47     42  0:00:01  0:00:01 --:--:--    89{"
id":4,"name":"Joao Bates","email":"joao@bates.com"}

jvafo@DESKTOP-QE9VBV1 MINGW64 ~
$ curl -v localhost:8080/api/user/4
*   Trying 127.0.0.1:8080...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /api/user/4 HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.78.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200
< Content-Type: application/json
< Transfer-Encoding: chunked
< Date: Thu, 19 Jan 2023 23:38:19 GMT
<
{ [59 bytes data]
100    53    0    53    0     0   3549      0 --:--:-- --:--:-- --:--:--  3785{"id":4,"name":"Joao Bates","email":"joao@bates.com"}
* Connection #0 to host localhost left intact
```

9) Utilizando o metodo Delete:
curl - -request DELETE http://localhost:8080/api/user/4

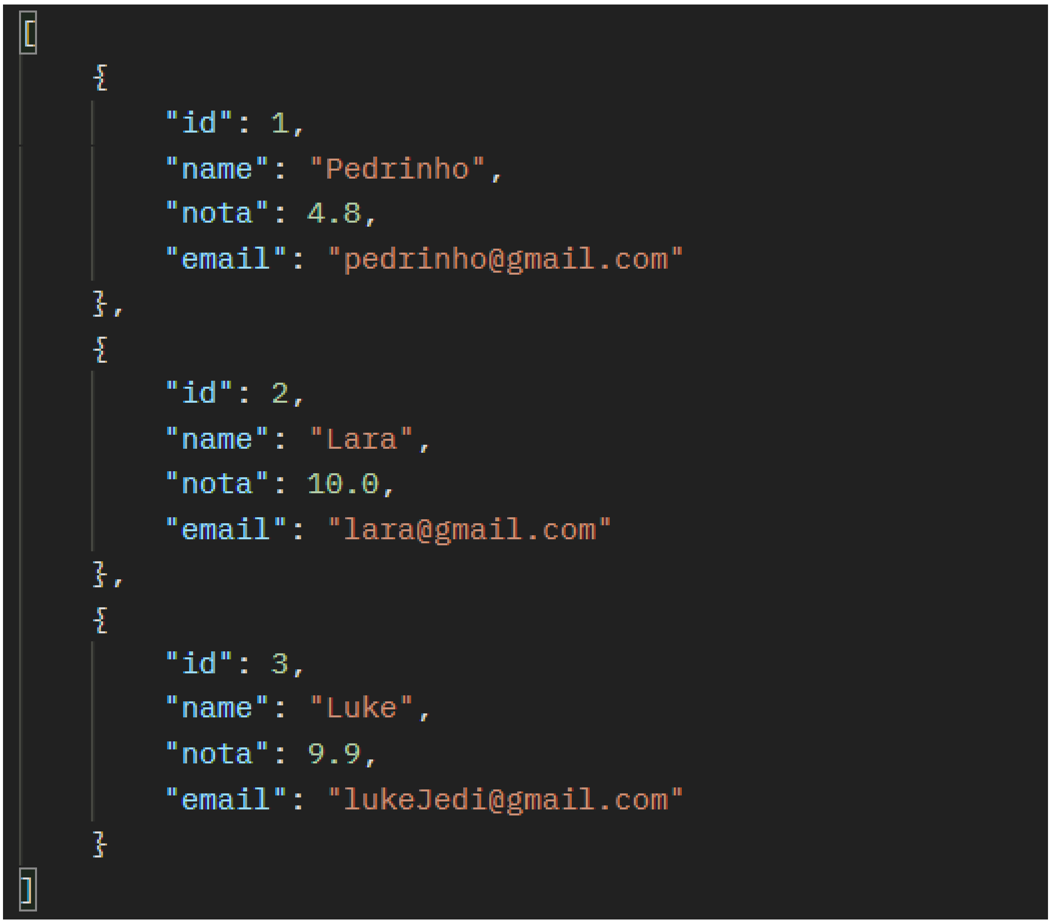


```
$ curl --request DELETE http://localhost:8080/api/user/4
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100     7  100     7    0     0    272      0 --:--:-- --:--:-- --:--:--   280Deleted
```

Mensagem Deleted ao fim do resultado no print

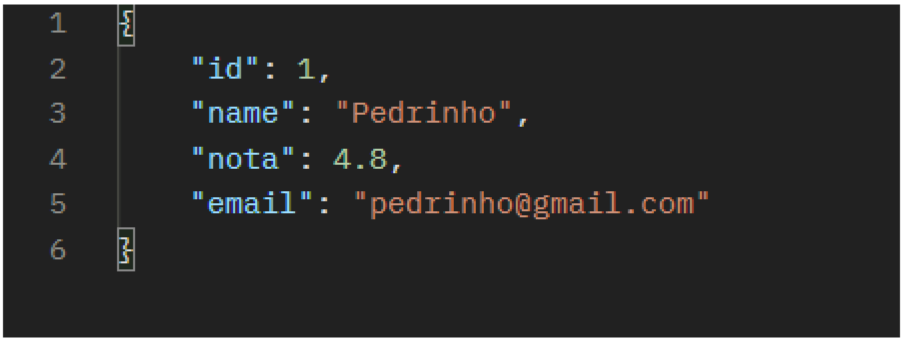
16) Foi criado a clase Student composta por nome, nota e email e criado rotas GET, POST, DELETE e PUT, abaixo esta o retorno de cada no Postman:

GET (http://localhost:8080/api/student):



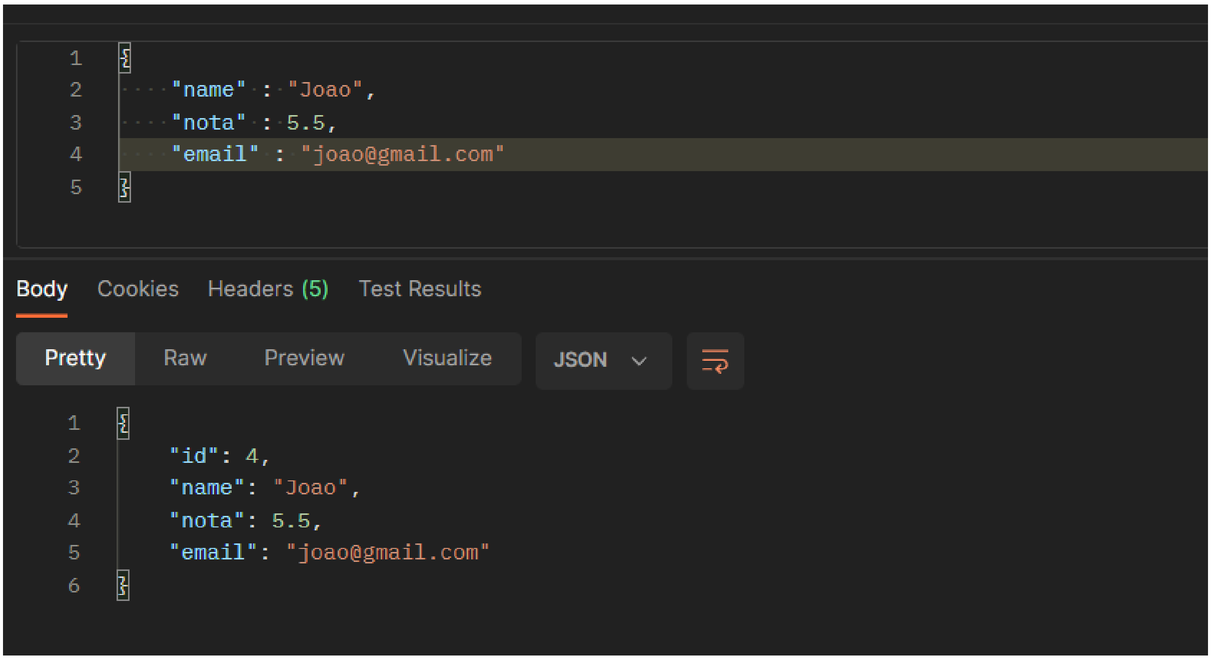
```
[
    {
        "id": 1,
        "name": "Pedrinho",
        "nota": 4.8,
        "email": "pedrinho@gmail.com"
    },
    {
        "id": 2,
        "name": "Lara",
        "nota": 10.0,
        "email": "lara@gmail.com"
    },
    {
        "id": 3,
        "name": "Luke",
        "nota": 9.9,
        "email": "lukeJedi@gmail.com"
    }
]
```

GET(http://localhost:8080/api/student/1):



```
{
    "id": 1,
    "name": "Pedrinho",
    "nota": 4.8,
    "email": "pedrinho@gmail.com"
}
```

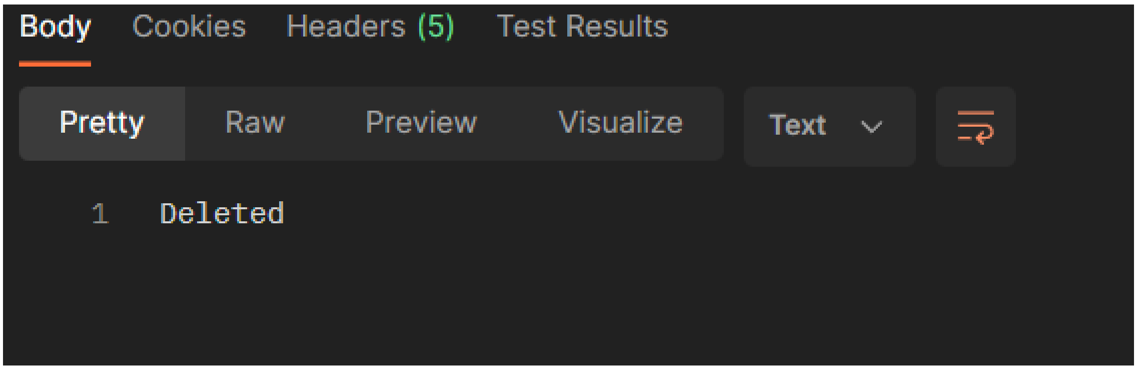
POST(http://localhost:8080/api/student):



```
{
    "name" : "Joao",
    "nota" : 5.5,
    "email" : "joao@gmail.com"
}
```

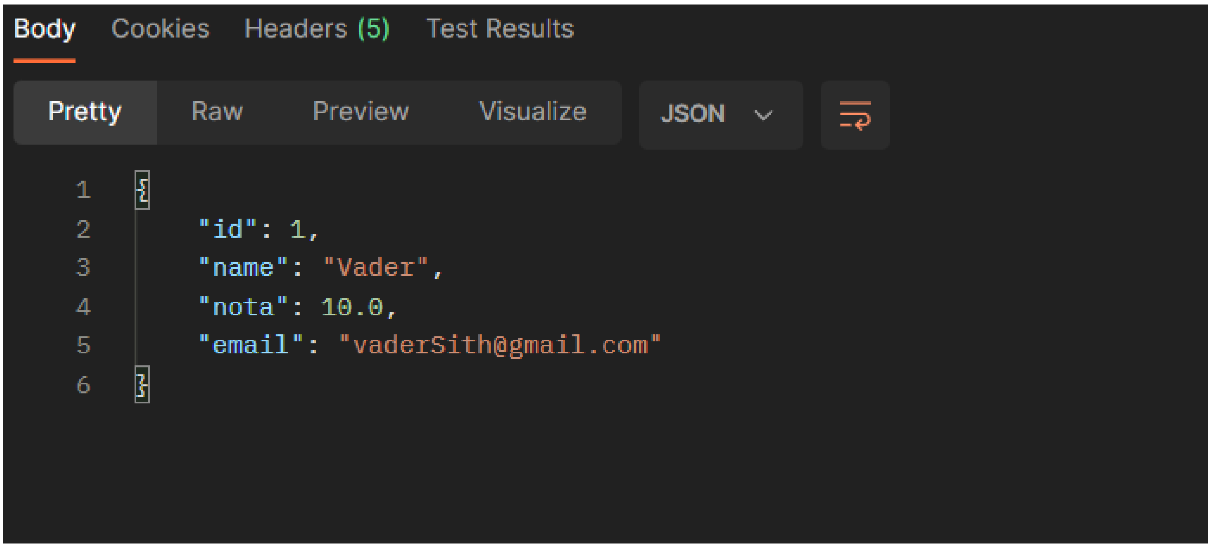
```
{
    "id": 4,
    "name": "Joao",
    "nota": 5.5,
    "email": "joao@gmail.com"
}
```

DELETE(http://localhost:8080/api/student/4):



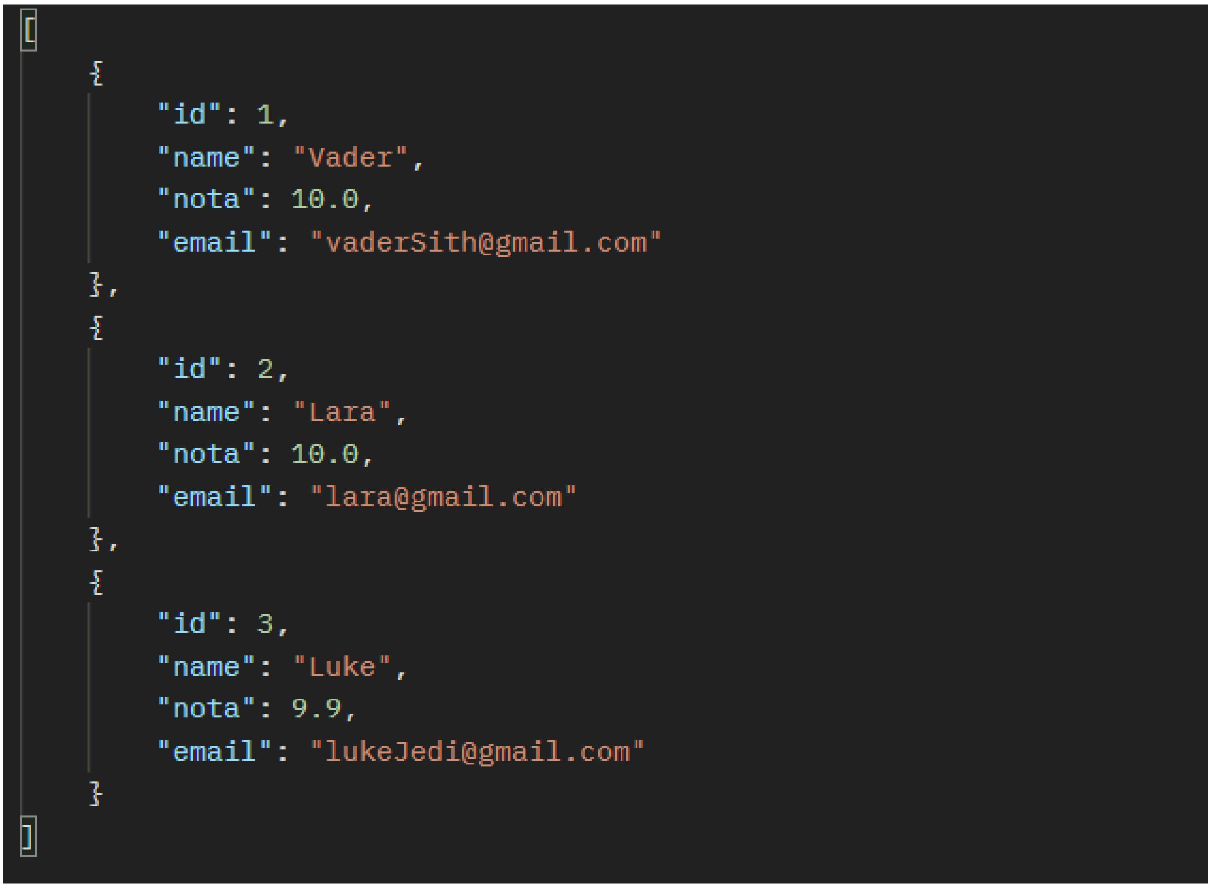
```
Deleted
```

PUT(http://localhost:8080/api/student/1):



```
{
    "id": 1,
    "name": "Vader",
    "nota": 10.0,
    "email": "vaderSith@gmail.com"
}
```

E um ultimo get para mostrar todos os dados no banco apos todos os comandos:
GET (http://localhost:8080/api/student):



```
[
    {
        "id": 1,
        "name": "Vader",
        "nota": 10.0,
        "email": "vaderSith@gmail.com"
    },
    {
        "id": 2,
        "name": "Lara",
        "nota": 10.0,
        "email": "lara@gmail.com"
    },
    {
        "id": 3,
        "name": "Luke",
        "nota": 9.9,
        "email": "lukeJedi@gmail.com"
    }
]
```