



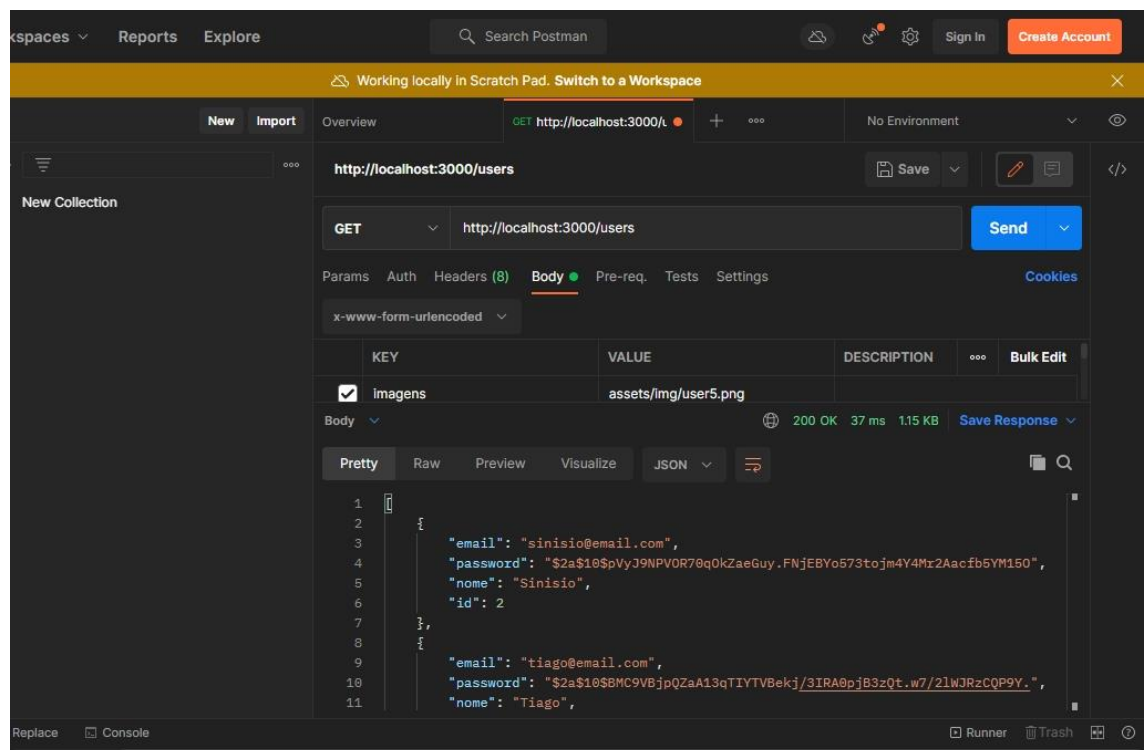
Relatório das Atividades

UC10 – Interação com APIs

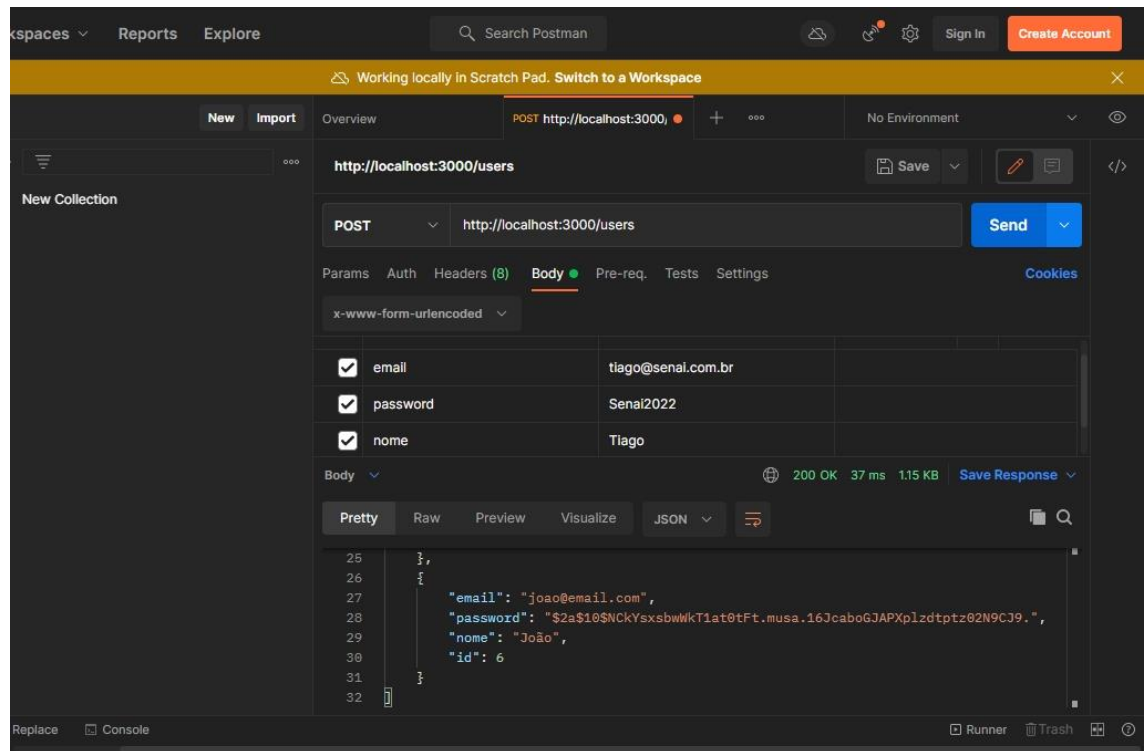
Tiago de Almeida

1. Atividade Online 1 - Tela de Login (Testar os métodos GET, POST, PUT e DELETE ->Print do postman e código)

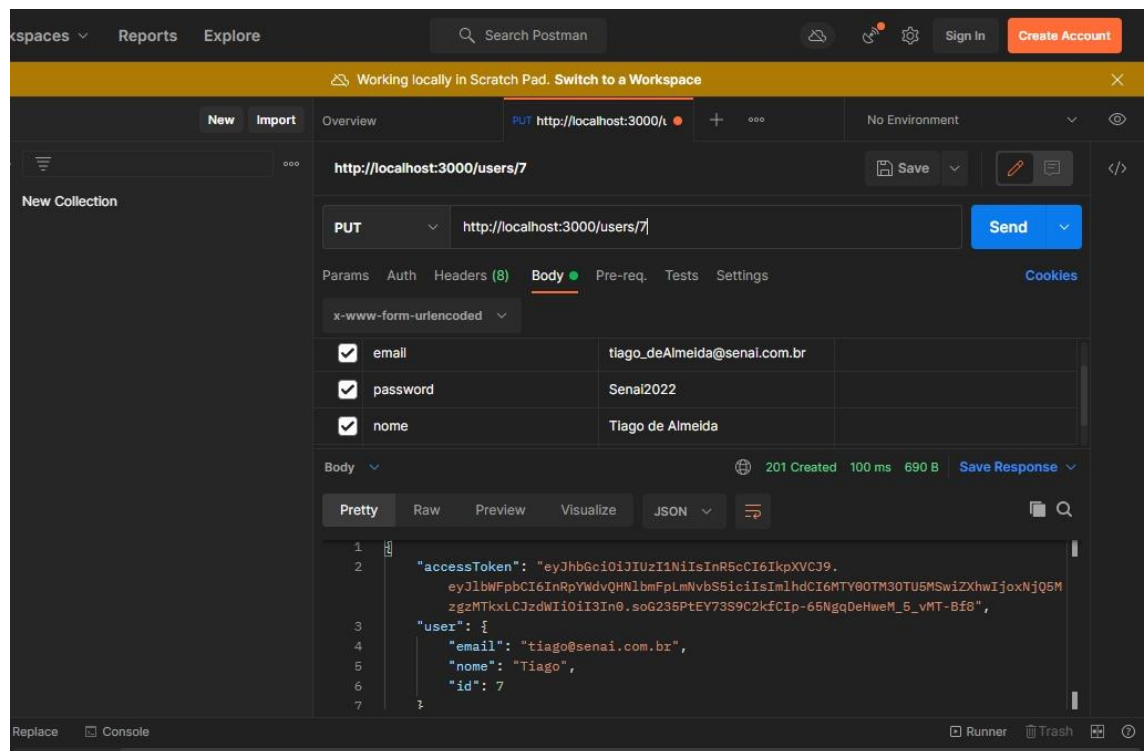
Método GET dos usuários do meu database. Existem 5 usuários (começando do ID 2) e criei mais um para essa atividade



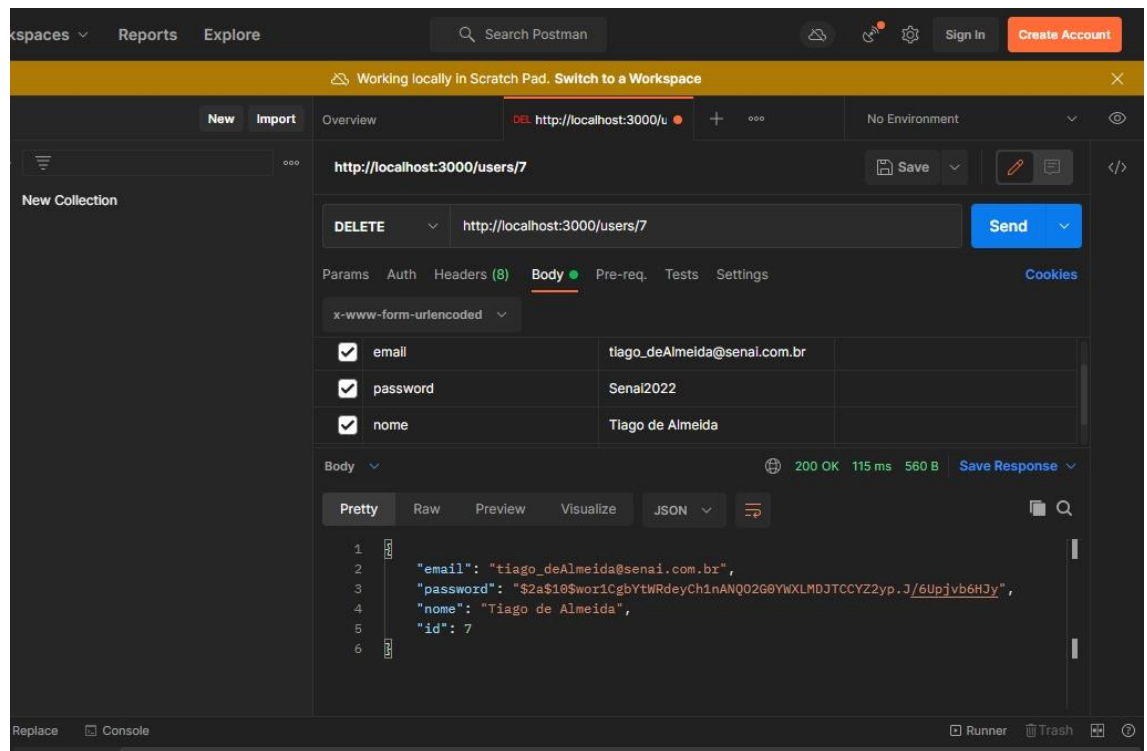
Criação de mais um usuário com o método POST. O novo usuário terá o ID 7.



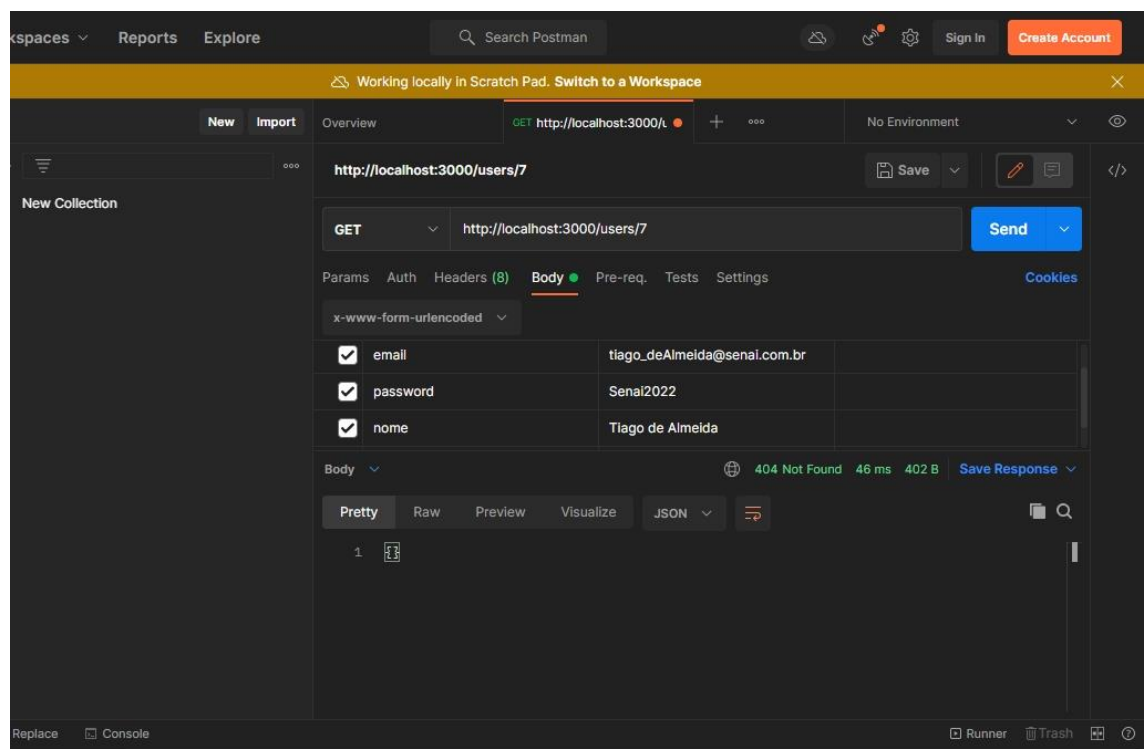
O novo usuário de ID 7 está visível abaixo e o método PUT, para atualizar as credenciais do usuário, será usado.



Método PUT foi aplicado para modificar o **nome** e **e-mail** do usuário de ID 7 como visto abaixo. O método DELETE será aplicado para deletar esse usuário do banco de dados.



DELETE foi aplicado e o usuário de ID 7 não aparece mais no banco de dados.

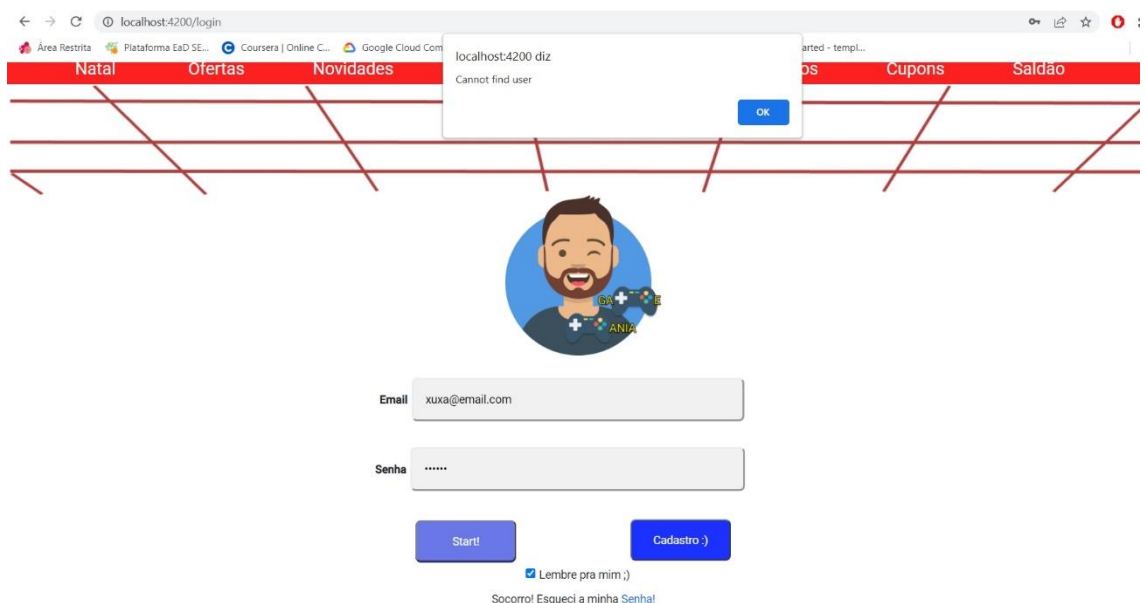


Arquivo db.json com todos os usuários cadastrados

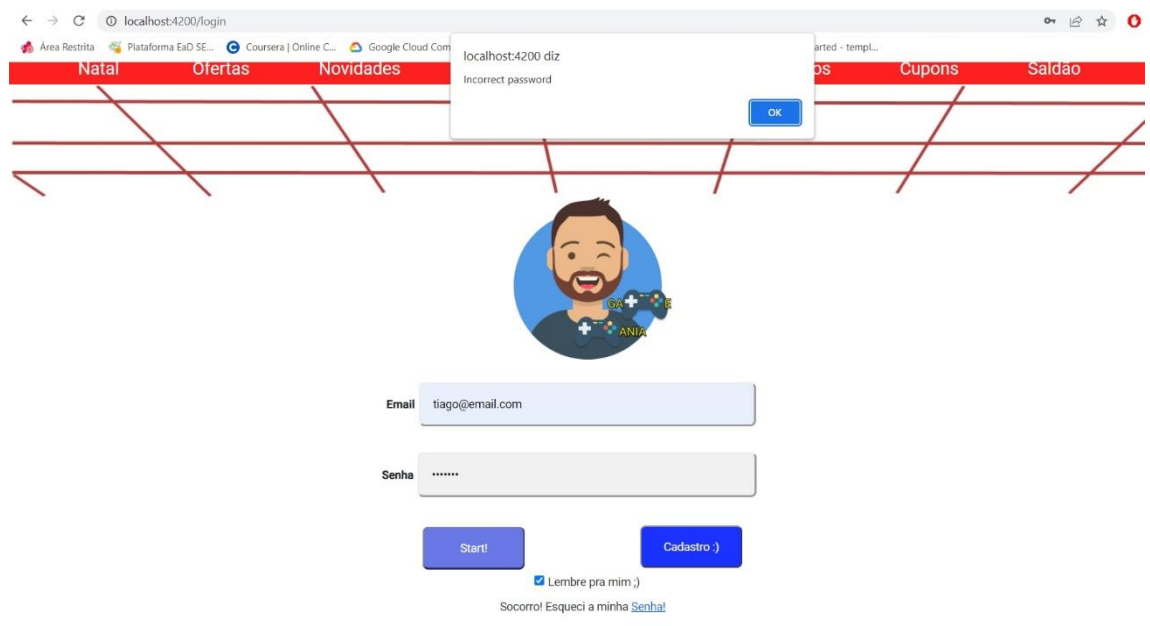
```
db.json
db.json > [ ] users > {} 2 > email
1 {
2   "users": [
3     {
4       "email": "sinisio@email.com",
5       "password": "$2a$10$pVyJ9NPVOR70q0kZaeGuy.FNjE8Yo573tojm4Y4Mr2Aacfb5YM150",
6       "nome": "Sinisio",
7       "id": 2
8     },
9     {
10      "email": "tiago@email.com",
11      "password": "$2a$10$BMC9VBjpQZaA13qTIYTVBekj/3IRA0pjB3zQt.w7/2lWJRzCQP9Y.",
12      "nome": "Tiago",
13      "id": 3
14    },
15    {
16      "email": "silvia@email.com",
17      "password": "$2a$10$QC.7gZ4lR9U0o4no3NtYIOPws8.ZHz3FuGNWgRbkJYcc00X4sw/ym",
18      "nome": "Silvia",
19      "id": 4
20    },
21    {
22      "email": "maria@email.com",
23      "password": "$2a$10$VR6WR7UV3vYOIP91S0ZRJe5JONq2r8UYRuTdNUJayHVXKQD589nCA0",
24      "nome": "Maria",
25      "id": 5
26    },
27    {
28      "email": "joao@email.com",
29      "password": "$2a$10$NCKYsxsBwMkT1at0tFt.musa.16JcaboGJAPXplzdtptz02N9CJ9.",
30      "nome": "João",
31      "id": 6
32    }
33  ],
34  "password": "1234567890"
35 }
```

2. Atividade Online 2 - Tela de Login (Configurar mensagens de retorno de erros)

Mensagem de erro quando o usuário não está cadastrado no banco de dados.



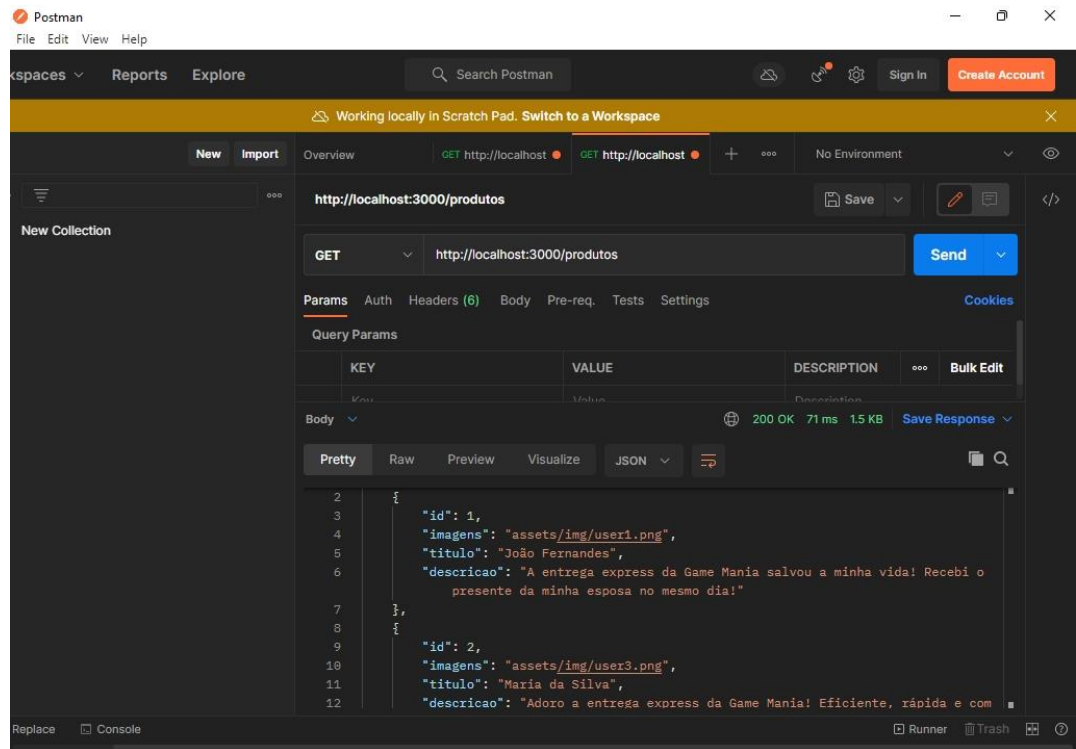
Mensagem de erro quando a senha inserida está incorreta.



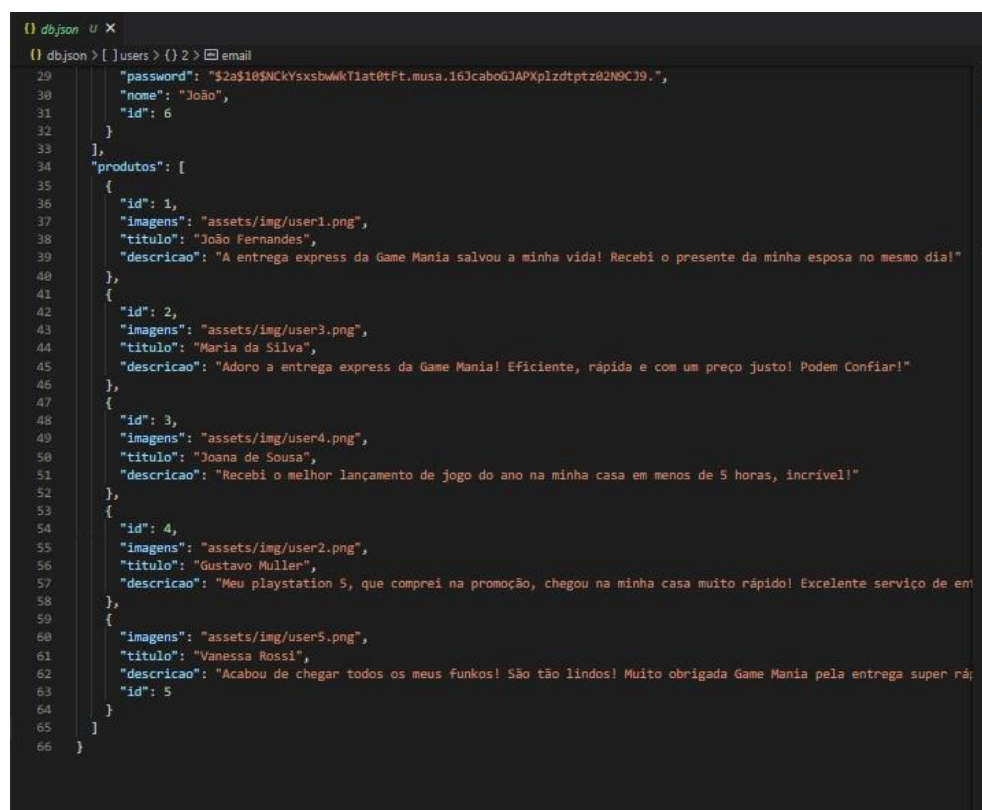
3. Encontro Remoto 1 - Tela Home Game Mania (Implementar método GET -> Print do postman e código)

Implementei o método GET na minha tela de *entrega* porque a minha *home* já contém muitos detalhes e informações, e não queria desconfigurar tudo.

Banco de dados do Postman com as opiniões dos usuários sobre o método de entrega Express da Game Mania exibido através do GET.



Opiniões dos usuários no arquivo db.json

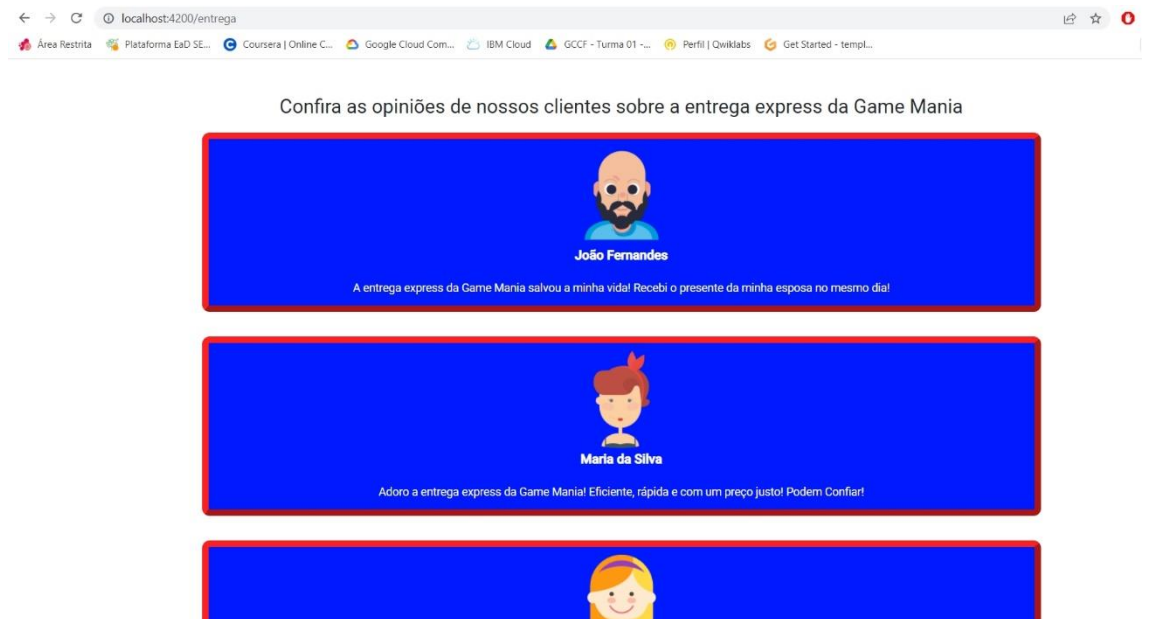


HTML do componente *entrega* exibindo quais informações das opiniões dos clientes serão mostradas na página *entrega*.

```
# entrega.component... M 19
<? entrega.component... M 20
TS entrega.component.spec.ts 21
TS entrega.component.ts M 22
> footer 23
  > header 24
    # header.component.css M 25
    <? header.component.h... M 26
    TS header.component.spec.ts 27
    TS header.component.ts 28
  > home 29
    # home.component.css M 30
    <? home.component.ht... M 31
    TS home.component.spec.ts 32
    TS home.component.ts M 33
  > inicial 34
    35
    36
    37
```

```
<a href="mailto:gamedamania@gmail.com?subject=venho da Game Mania!"></a>
<a href="https://api.whatsapp.com/send?phone=11112345678" target="_blank"></a>
</div>
</div>
<h1 style="text-align: center;">Confira as opiniões de nossos clientes sobre a entrega express da
<section class="clientes container" *ngFor="let produto of listaProdutos">
  <div class="row">
    <div class="prudtos-teste col">
      <img [src]=produto.imagens style="width: 120px;margin-top: 10px;margin-left: 44%;" class="
      <h3><b>{{produto.titulo}}</b></h3>
      <p>{{produto.descricao}}</p>
    </div>
  </div>
</section>
</div>
```

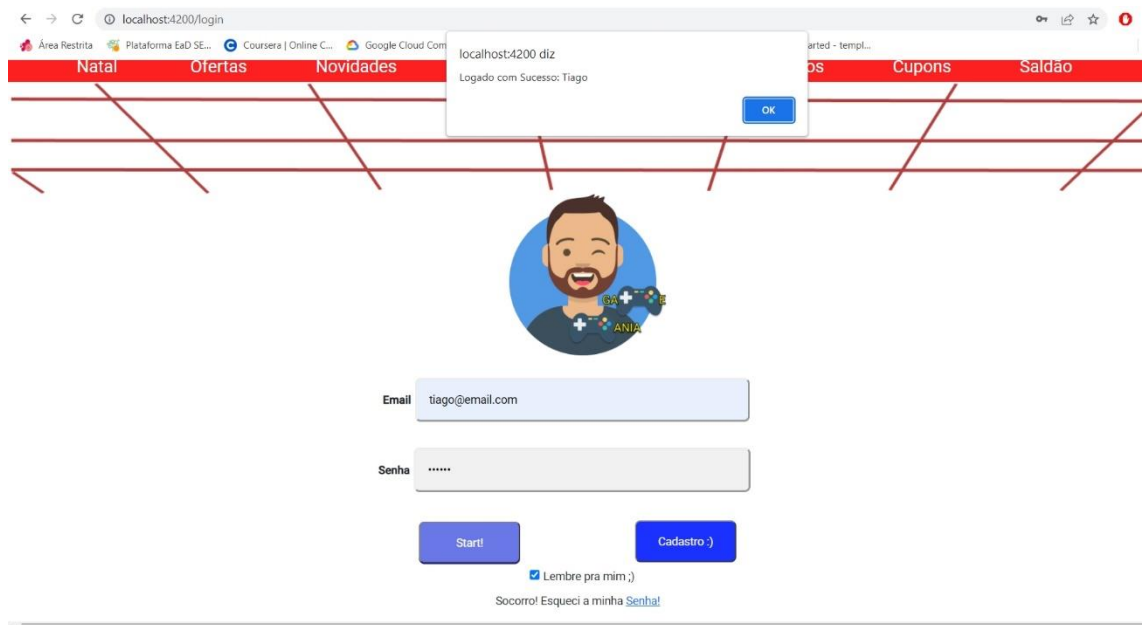
Página de *entrega* mostrando as opiniões dos clientes obtidas através do método GET.



4. Encontro Remoto 2 - Tela de Login (Configurar mensagens para captura e tratamento de exceções para o formulário)

Eu modifiquei o código para que aparecesse uma mensagem de sucesso do login para o usuário, no qual o nome da pessoa que efetuou o login seria exibido na mensagem.

Mensagem exibida nesse caso foi: *Logado com Sucesso: Tiago*



O login foi efetuado com as minhas credenciais, que são:

```
{
  "email": "tiago@email.com",
  "password": "$2a$10$BMC9VBjpQZaA13qTIYTVBekj/3IRA0pjB3zQt.w7/2lWJRzCQP9Y.",
  "nome": "Tiago",
  "id": 3
},
```

Após o usuário clicar em ok na janela de confirmação, o mesmo será redirecionado para a página inicial (página secreta) do website, que apenas pessoas cadastradas no banco de dados tem acesso. Nessa página secreta, eu coloquei outro método GET para exibir o ID, nome e e-mail de todos os usuários cadastrados no banco de dados.

Comandos utilizados no código para evitar o SQL injection.

```

21
22 header.component.ts M
23   home
24     # home.component.css M
25     home.component.html M
26     home.component.spec.ts M
27     home.component.ts M
28   initial
29     # initial.component.css M
30     initial.component.html M
31     initial.component.spec.ts M
32     initial.component.ts M
33   login
34     # login.component.css M
35     login.component.html M
36     login.component.spec.ts M
37     login.component.ts M
38   app-routing.module.ts M
39   # app.component.css M
40   app.component.html M
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```