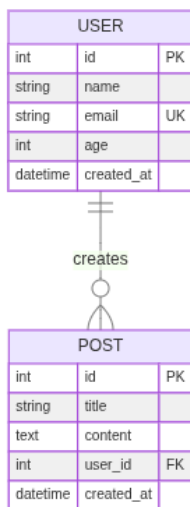


# Atividade 10 - Implementação de CRUD com Prisma no NestJS

## Mermaid ER Diagram — USER / POST



Abra este arquivo no navegador (ou use um servidor local) para renderizar o diagrama.





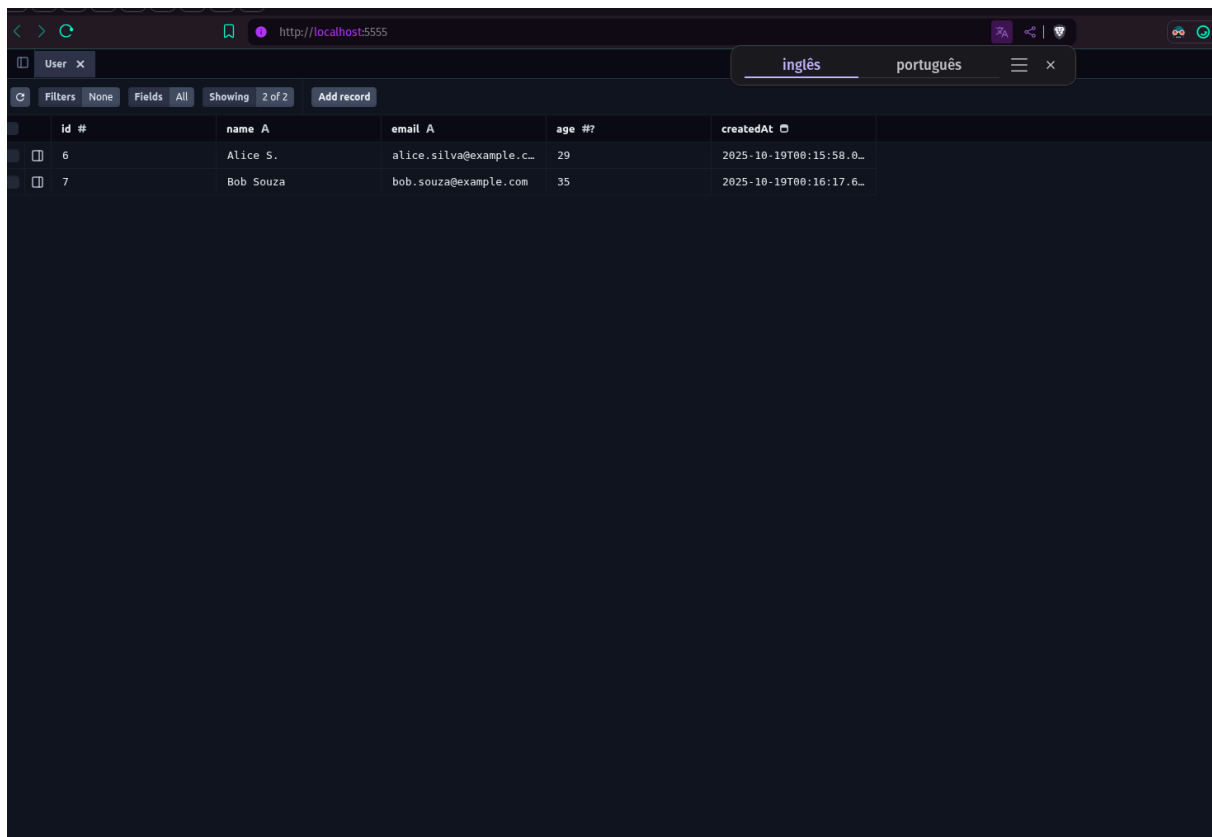
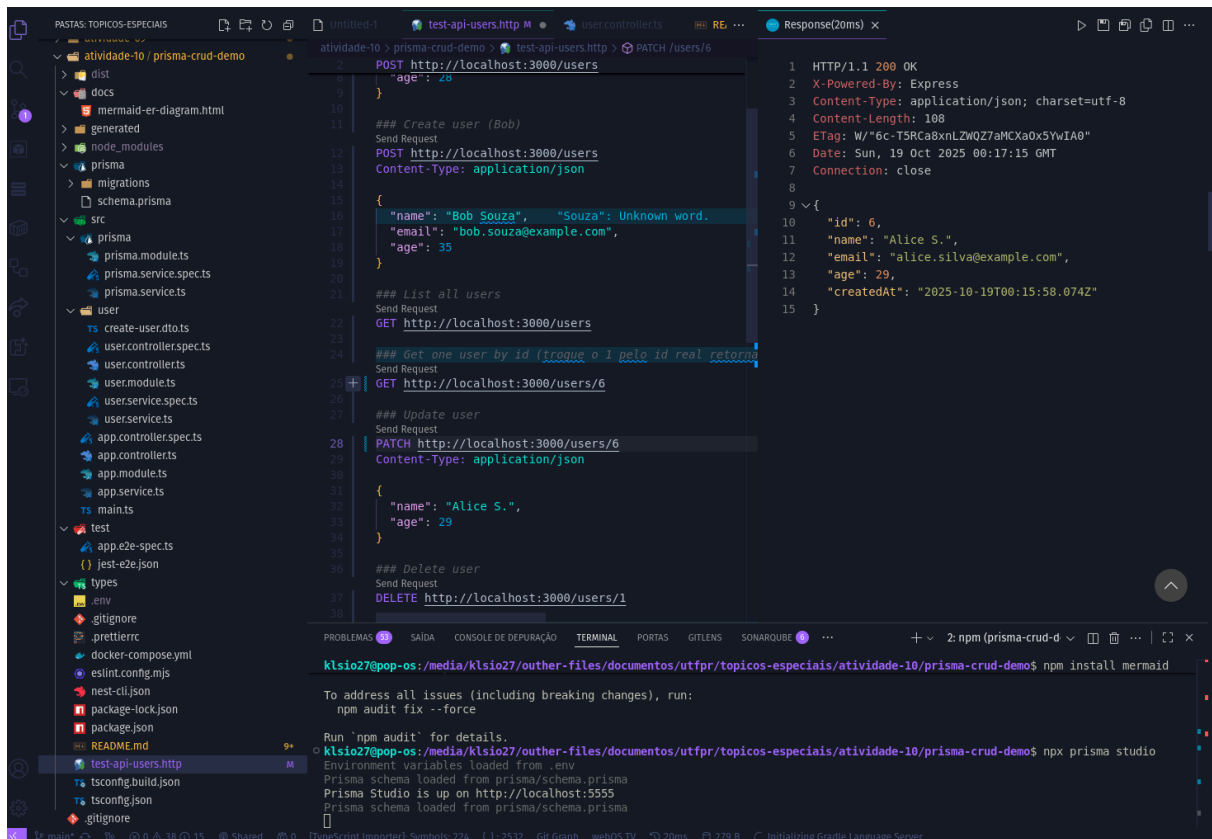


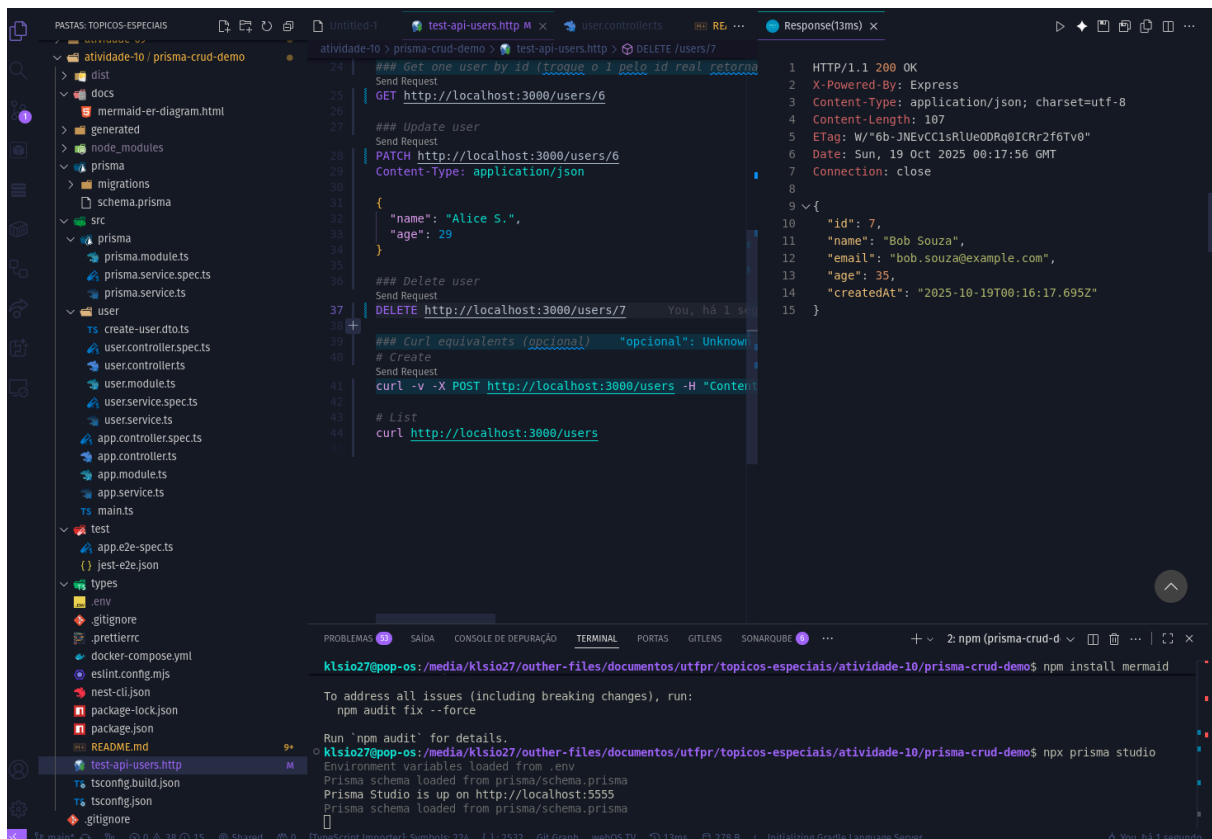
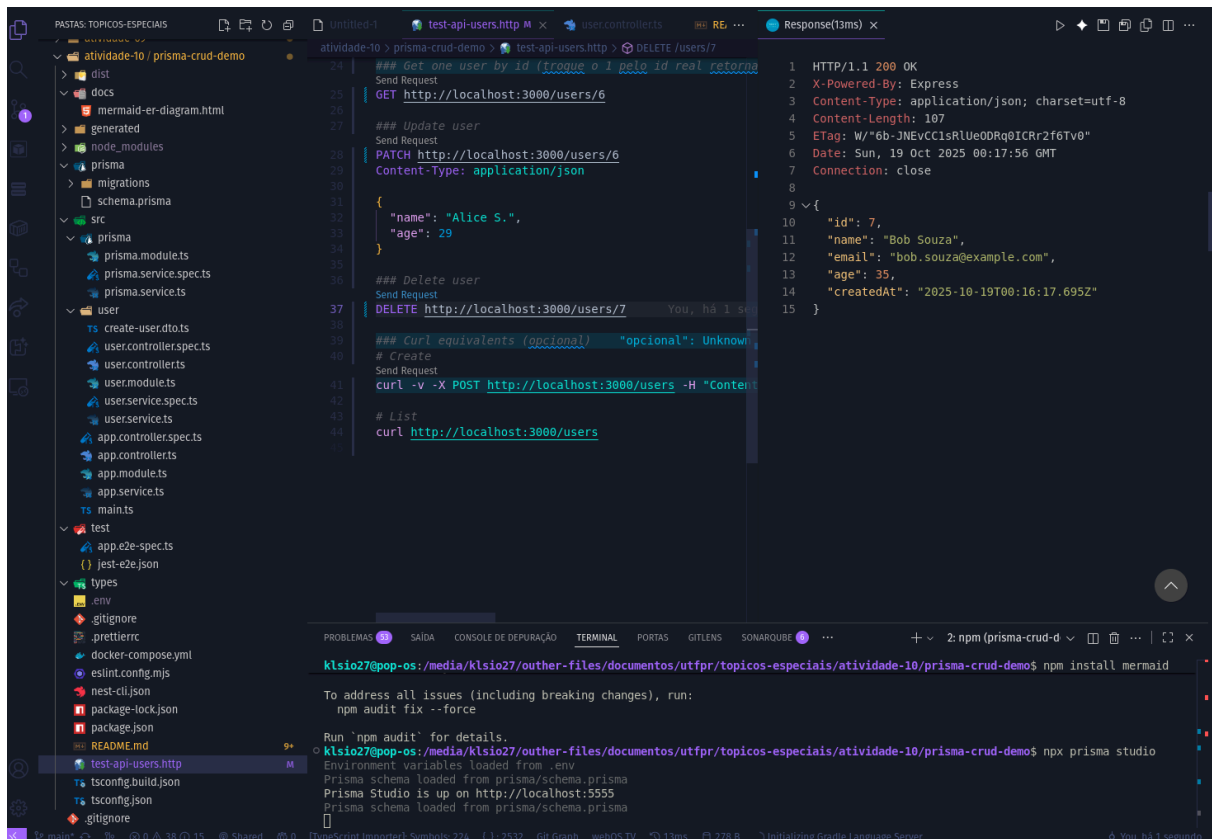
```
1 You, há 21 minutos | 1 author (You)
2 ### Create user (Alice)
3 Send Request
4 POST http://localhost:3000/users
5 Content-Type: application/json
6 {
7   "name": "Alice Silva",
8   "email": "alice.silva@example.com",
9   "age": 28
10 }
11 ### Create user (Bob)
12 Send Request
13 POST http://localhost:3000/users
14 Content-Type: application/json
15 {
16   "name": "Bob Souza",
17   "email": "bob.souza@example.com",
18   "age": 35
19 }
20 ### List all users
21 Send Request
22 GET http://localhost:3000/users
23
24 ### Get one user by id (troque o 1 pelo id real retorne)
25 Send Request
26 GET http://localhost:3000/users/1
27
28 ### Update user
29 Send Request
30 PATCH http://localhost:3000/users/1
31 Content-Type: application/json
32 {
33   "name": "Alice S.",
34   "age": 29
35 }
36 ### Delete user
```

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 221
5 ETag: W/"dd-B9L/6HqkLkthIPoFId5nnTJgo"
6 Date: Sun, 19 Oct 2025 00:16:29 GMT
7 Connection: close
8
9 {
10   "id": 6,
11   "name": "Alice Silva",
12   "email": "alice.silva@example.com",
13   "age": 28,
14   "createdAt": "2025-10-19T00:15:58.074Z"
15 }
```

```
1 POST http://localhost:3000/users
2 {
3   "name": "Alice Silva",
4   "email": "alice.silva@example.com",
5   "age": 28
6 }
7 ### Create user (Bob)
8 Send Request
9 POST http://localhost:3000/users
10 Content-Type: application/json
11 {
12   "name": "Bob Souza",
13   "email": "bob.souza@example.com",
14   "age": 35
15 }
16 ### List all users
17 Send Request
18 GET http://localhost:3000/users
19
20 ### Get one user by id (troque o 1 pelo id real retorne)
21 Send Request
22 GET http://localhost:3000/users/6
23
24 ### Update user
25 Send Request
26 PATCH http://localhost:3000/users/1
27 Content-Type: application/json
28 {
29   "name": "Alice S.",
30   "age": 29
31 }
32 ### Delete user
```

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 111
5 ETag: W/"6f-eGPZGhM2tWxiiJ0Re4U0tKP7a0"
6 Date: Sun, 19 Oct 2025 00:17:02 GMT
7 Connection: close
8
9 {
10   "id": 6,
11   "name": "Alice Silva",
12   "email": "alice.silva@example.com",
13   "age": 28,
14   "createdAt": "2025-10-19T00:15:58.074Z"
15 }
```





## 2 Descrição Breve de Cada Teste

### Teste: Criando um Usuário ( **POST /users** )

- **Objetivo:** Testar a criação de um novo usuário no sistema
- **Entrada:** `{ "name": "João", "email": "joao@example.com", "age": 25 }`
- **Saída:** `{ "id": 1, "name": "João", "email": "joao@example.com", "age": 25, "createdAt": "..." }`
- **Resultado:** Sucesso, código **201 Created**

### Teste: Listando Usuários ( **GET /users** )

- **Objetivo:** Recuperar a lista de usuários cadastrados
- **Entrada:** Nenhuma (requisição GET simples)
- **Saída:** `[{ "id": 1, "name": "João", "email": "joao@example.com", "age": 25, "createdAt": "..." }, ...]`
- **Resultado:** Sucesso, código **200 OK**

### Teste: Obtendo Usuário por ID ( **GET /users/:id** )

- **Objetivo:** Recuperar um usuário específico pelo seu ID
- **Entrada:** `GET /users/1`
- **Saída:** `{ "id": 1, "name": "João", "email": "joao@example.com", "age": 25, "createdAt": "..." }`
- **Resultado:** Sucesso, código **200 OK** ; se não encontrado, **404 Not Found**

### Teste: Atualizando Usuário ( **PATCH /users/:id** )

- **Objetivo:** Atualizar campos do usuário (nome ou idade)
- **Entrada:** `PATCH /users/1` com body `{ "name": "João S.", "age": 26 }`
- **Saída:** `{ "id": 1, "name": "João S.", "email": "joao@example.com", "age": 26, "createdAt": "..." }`
- **Resultado:** Sucesso, código **200 OK** ; se ID não existir, **404 Not Found**

### Teste: Deletando Usuário ( **DELETE /users/:id** )

- **Objetivo:** Remover o usuário do sistema
- **Entrada:** `DELETE /users/1`
- **Saída:** `{ "id": 1, "name": "João", "email": "joao@example.com", "age": 25, "createdAt": "..." }` (ou objeto de confirmação)



- **Resultado:** Sucesso, código **200 OK** ou **204 No Content**