

Documento de Testes Manuais — API JSONPlaceholder

QA: Vitor Hugo de Oliveira Machado

1. Objetivo

Definir, estruturar e documentar os principais cenários de testes manuais da API pública JSONPlaceholder,

garantindo que qualquer testador consiga reproduzir os testes, validar comportamento funcional,

identificar falhas ou inconsistências e compreender rapidamente o escopo testado.

2. Escopo dos Testes

Endpoints validados:

- GET /posts
- GET /posts/{id}
- GET /users

Foco:

- Funcionamento funcional básico
- Estrutura das respostas JSON
- Tratamento de cenários inválidos
- Confiabilidade das informações retornadas

3. Ambiente de Teste

URL base: <https://jsonplaceholder.typicode.com>

Ferramentas:

- Postman

4. Estratégia de Teste

Modelagem considerando:

- Casos positivos
- Casos negativos
- Validação de estrutura de dados
- Facilidade de reprodução por terceiros

5. Casos de Teste

CT-01 — Listar todos os posts

Resultado esperado:

- Status 200 OK
- Lista com 100 registros
- Campos: userId, id, title, body

The image displays two screenshots of the Postman API client interface, showing a GET request to the endpoint `JSONPlaceholder_API /posts` using the `JSONPlaceholder_ENV` environment.

Top Screenshot: The request is `GET {{baseUrl}}/posts`. The response status is **200 OK** (highlighted with a red box), with a response time of 38 ms and a size of 8.02 KB. The response body is a JSON array of three posts. Red arrows point to the first post object in the array.

```
1 {
2   {
3     "userId": 1,
4     "id": 1,
5     "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
6     "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
7   },
8   {
9     "userId": 1,
10    "id": 2,
11    "title": "qui est esse",
12    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
13  },
14  {
15    "userId": 1,
16    "id": 3,
```

Bottom Screenshot: The same request is shown, but the response body is scrolled down to show the third post object. A red arrow points to the `"id": 100` field within the third post object.

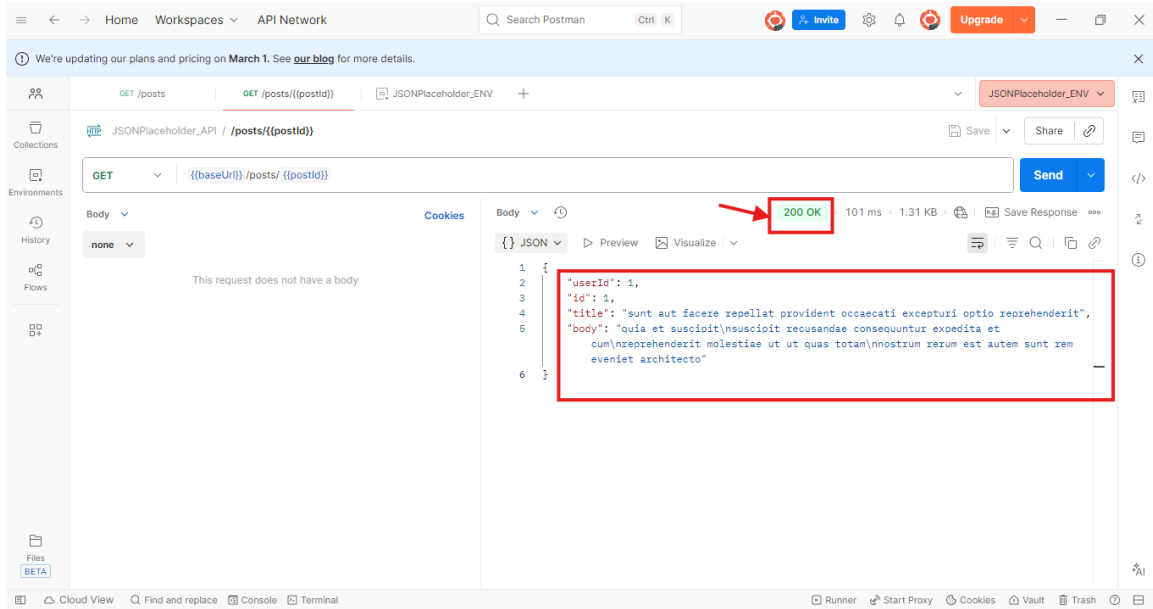
```
585 "userId": 10,
586 "id": 98,
587 "title": "laboriosam dolor voluptates",
588 "body": "doloremque ex facilis sit sint culpa\nsoluta assumenda eligendi non ut eius\nsequi ducimus vel quasi\nveritatis est dolores"
589 },
590 {
591   "userId": 10,
592   "id": 99,
593   "title": "temporibus sit alias delectus eligendi possimus magni",
594   "body": "quo deleniti praesentium dicta non quod\naut est molestias\nmolestias et officia quis nihil\nitaque dolorem quia"
595 },
596 {
597   "userId": 10,
598   "id": 100,
599   "title": "at nam consequatur ea labore ea harum",
600   "body": "cupiditate quo est a modi nesciunt soluta\nnisi voluptas error itaque dicta in\nautem qui minus magnam et distinctio eum\nnaccusamus ratione error aut"
601 },
602 ]
```

CT-02 — Buscar post por ID válido (/posts/1)

- Status 200 OK

- id = 1

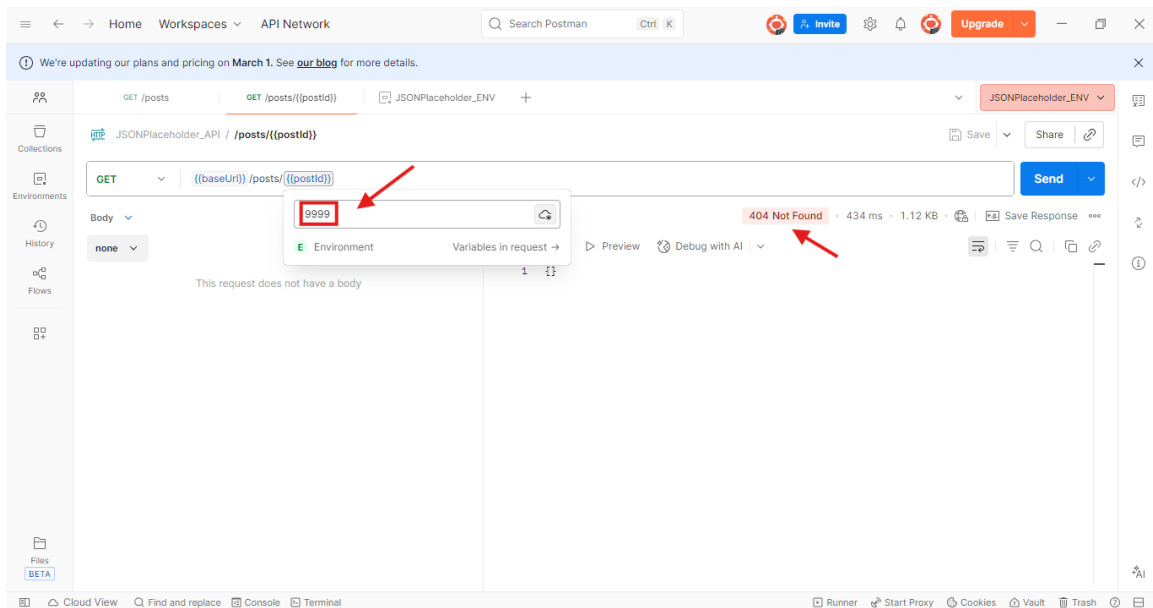
- title e body preenchidos



CT-03 — Buscar post inexistente (/posts/9999)

- Status 404 ou {}

- Nenhum erro 500



CT-04 — Listar usuários

- Status 200 OK

- Lista com 10 usuários

- Campos: id, name, username, email válido

The image displays two screenshots of the Postman API client interface, demonstrating a successful GET request to the `/users` endpoint of the `JSONPlaceholder_API`.

Top Screenshot: The request is a GET to `JSONPlaceholder_API /users`. The status is **200 OK** (highlighted with a red box). The response body (highlighted with a red box) is a JSON array of user objects. The first two users are:

```
{
  "id": 1,
  "name": "Mr. Dennis Schulist",
  "username": "Leopoldo_Costezy",
  "email": "Kaziley_Dach@jasper.info",
  "address": {
    "street": "Noberto Crossing",
    "suite": "Apt. 980",
    "city": "South Christy",
    "zipcode": "23886-1337",
    "geo": {
      "lat": "-71.4197",
      "lng": "71.7478"
    }
  },
  "phone": "1-477-935-8478 x6438",
  "website": "ola.org",
  "company": {
    "name": "Conside-Lockman",
    "catchPhrase": "Synchronised bottom-line interface",
    "bs": "e-enable innovative applications"
  }
},
{
  "id": 2,
  "name": "Kurtis Weissnat",
  "username": "Elayn.Skiles",
  "email": "Telly.Hoeger@billy.biz",
  "address": {
    "street": "Max Trail",
    "suite": "Suite 200",
    "city": "Houmaouth",
    "zipcode": "88884-1899",
    "geo": {
      "lat": "24.8918",
      "lng": "21.8984"
    }
  },
  "phone": "1-477-935-8478 x6438",
  "website": "ola.org",
  "company": {
    "name": "Conside-Lockman",
    "catchPhrase": "Synchronised bottom-line interface",
    "bs": "e-enable innovative applications"
  }
}
```

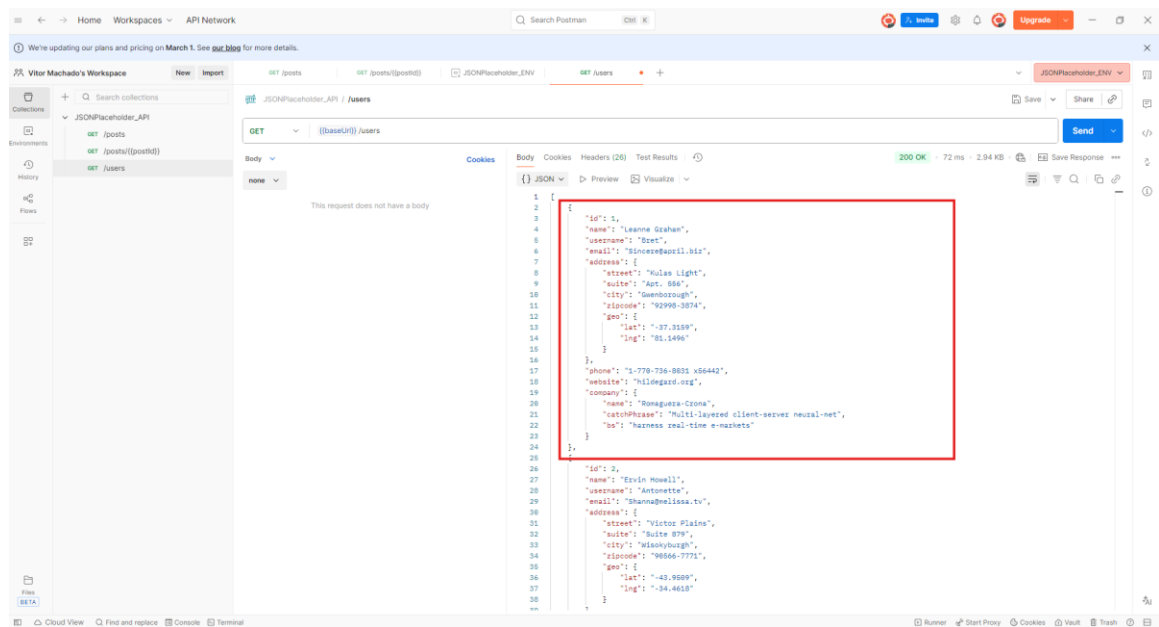
Bottom Screenshot: The same request is shown, but the response body is scrolled down to show the last two users in the array:

```
{
  "id": 18,
  "name": "Clementina DuBuque",
  "username": "Roziah.Stanton",
  "email": "Ray.Padberg@karina.biz",
  "address": {
    "street": "Wattie Turnpike",
    "suite": "Suite 190",
    "city": "Labashouky",
    "zipcode": "78429-2361",
    "geo": {
      "lat": "-38.3386",
      "lng": "87.2233"
    }
  },
  "phone": "824-648-3884",
  "website": "ambrose.net",
  "company": {
    "name": "Hoeger LLC",
    "catchPhrase": "Centralized empowering task-force",
    "bs": "target end-to-end models"
  }
},
{
  "id": 19,
  "name": "Clementina DuBuque",
  "username": "Roziah.Stanton",
  "email": "Ray.Padberg@karina.biz",
  "address": {
    "street": "Wattie Turnpike",
    "suite": "Suite 190",
    "city": "Labashouky",
    "zipcode": "78429-2361",
    "geo": {
      "lat": "-38.3386",
      "lng": "87.2233"
    }
  },
  "phone": "824-648-3884",
  "website": "ambrose.net",
  "company": {
    "name": "Hoeger LLC",
    "catchPhrase": "Centralized empowering task-force",
    "bs": "target end-to-end models"
  }
}
```

CT-05 — Validar formato JSON

- Content-Type application/json

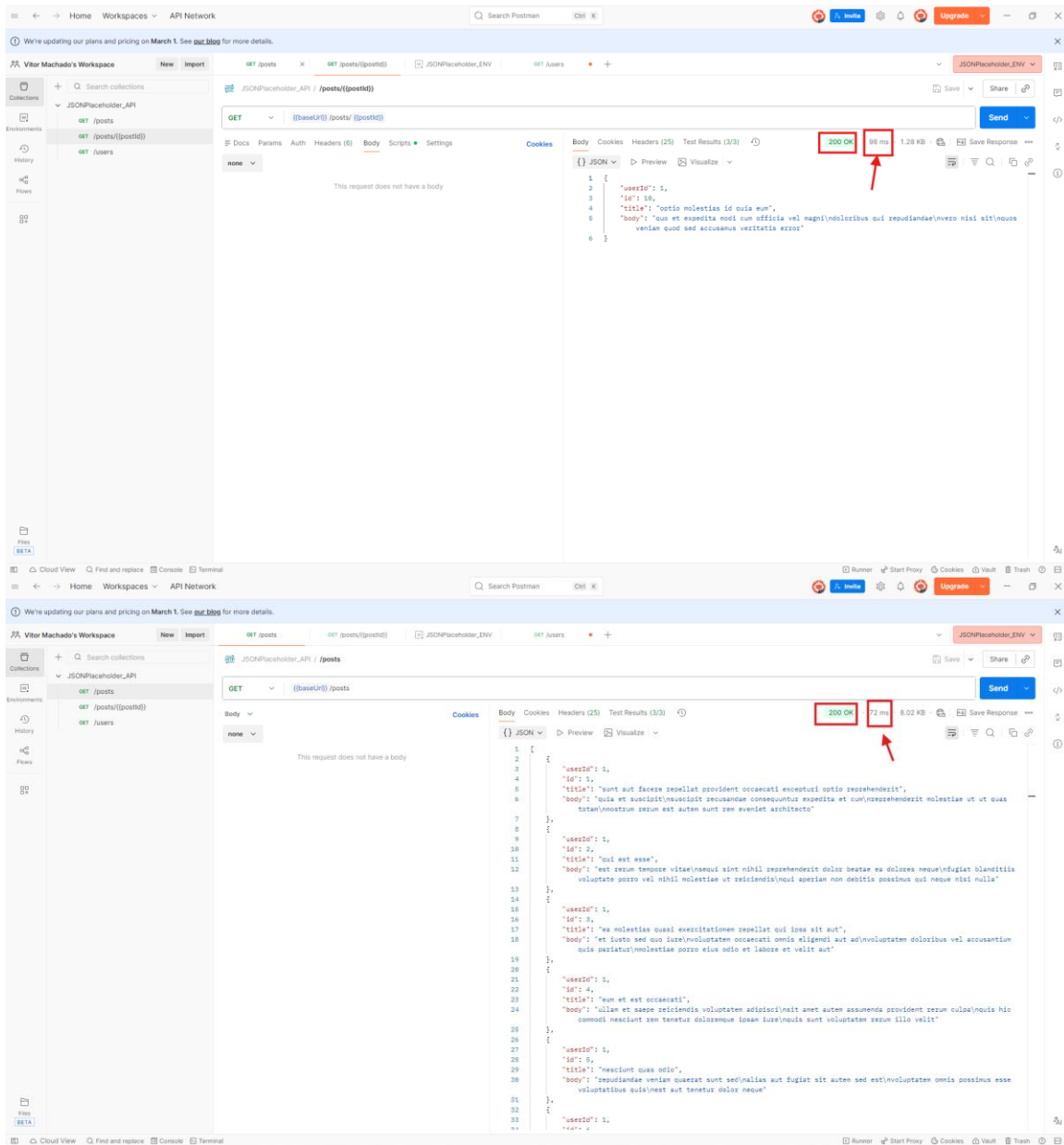
- Estrutura válida sem campos críticos nulos



CT-06 — Tempo de resposta

- Inferior a 2 segundos

- Sem timeout



6. Riscos Identificados

- API pública pode ter lentidão ou inconsistência.
- Dados não persistidos (ambiente mock).
- Alguns erros retornam {} ao invés de 404.

7. Critérios de Aceite

- Status corretos em todos endpoints.
- Estrutura JSON válida.
- Nenhum erro 500.
- Tempo de resposta aceitável.

8. Conclusão

Os testes manuais confirmam funcionamento básico, consistência de dados e previsibilidade em cenários inválidos, permitindo reprodução simples por qualquer testador.