

## 1. Caso de Teste: Create - Criação de Usuário

### Descrição:

Este caso de teste tem como objetivo testar a criação de um novo usuário utilizando o endpoint `/usuarios` da API `https://serverest.dev/`.

### Pré-requisitos:

- Ter acesso à API `https://serverest.dev/`.
- Ter um ambiente de teste disponível.

### Passos:

#### 1. Enviar Requisição POST:

- Método: POST
- URL: `https://serverest.dev/usuarios`
- Headers: Nenhum necessário
- Body:

json:

```
{
  "nome": "Ana Silva",
  "email": "ana@silva.com",
  "password": "ana123",
  "administrador": true
}
```

- Verificar se a resposta possui status code 201 (Created).
- Verificar se a resposta contém o seguinte corpo:

json:

```
{
  "message": "Cadastro realizado com sucesso",
  "_id": "5PVu2zruuuHQXyeB"
}
```

### Critérios de Sucesso:

- A resposta da API deve retornar um status code 201 (Created).
- O corpo da resposta deve conter a mensagem "Cadastro realizado com sucesso" e o ID do usuário criado.

### Critérios de Falha:

- A resposta da API deve retornar um status code diferente de 201 em caso de falha na criação do usuário.

## 2. Caso de Teste: Read - Listagem de Usuários

### Descrição:

Este caso de teste tem como objetivo testar a listagem de usuários utilizando o endpoint `/usuarios` da API `https://serverest.dev/`.

### Pré-requisitos:

- Ter acesso à API `https://serverest.dev/`.
- Ter um ambiente de teste disponível.

### Passos:

#### 1. Enviar Requisição GET:

- Método: GET
- URL: `https://serverest.dev/usuarios`
- Headers: Nenhum necessário
- Verificar se a resposta possui status code 200 (OK).
- Verificar se a resposta contém a lista de usuários.

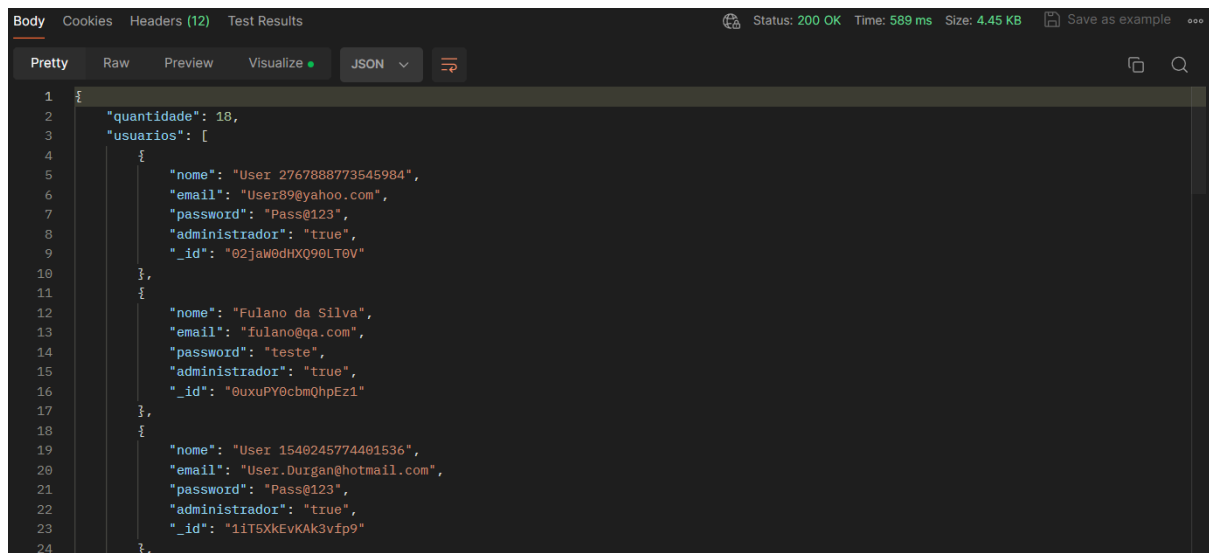
### Critérios de Sucesso:

- A resposta da API deve retornar um status code 200 (OK).
- A resposta deve conter a lista de usuários existentes no sistema.

### Critérios de Falha:

- A resposta da API deve retornar um status code diferente de 200 em caso de falha na leitura dos usuários.

### Status Code:



```
Body  Cookies  Headers (12)  Test Results  Status: 200 OK  Time: 589 ms  Size: 4.45 KB  Save as example  ...
Pretty  Raw  Preview  Visualize  JSON  ...
1  {
2    "quantidade": 10,
3    "usuarios": [
4      {
5        "nome": "User 2767888773545984",
6        "email": "User89@yahoo.com",
7        "password": "Pass@123",
8        "administrador": "true",
9        "_id": "02jaw0dHXQ90LT0V"
10      },
11      {
12        "nome": "Fulano da Silva",
13        "email": "fulano@qa.com",
14        "password": "teste",
15        "administrador": "true",
16        "_id": "0uxuPY0cbmQhpEz1"
17      },
18      {
19        "nome": "User 1540245774401536",
20        "email": "User.Durgan@hotmail.com",
21        "password": "Pass@123",
22        "administrador": "true",
23        "_id": "1iT5XkEvKak3vIp9"
24      }
25    ]
26  }
```

```
Body Cookies Headers (12) Test Results Status: 200 OK Time: 589 ms Size: 4.45 KB Save as example ...
Pretty Raw Preview Visualize JSON
25 {
26   "name": "Random function () { [native code] }",
27   "email": "Lucas68@hotmail.com",
28   "password": "Pass@123",
29   "administrador": "true",
30   "_id": "58Iee8PK7IL7C0Mo"
31 },
32 {
33   "name": "Ana Silva",
34   "email": "ana.silva@novodominio.com",
35   "password": "novaSenha",
36   "administrador": "true",
37   "_id": "5PVu2zruuuHQXyeB"
38 },
39 {
40   "name": "User 3197526461120512",
41   "email": "User_Prosacco@hotmail.com",
42   "password": "Pass@123",
43   "administrador": "true",
44   "_id": "5Zzb6GhA0V1sfH3s"
45 },
46 {
47   "name": "User 1872517943787520",
48   "email": "User_Smitham@hotmail.com",
49   "password": "Pass@123",
50   "administrador": "true",
```

```
Body Cookies Headers (12) Test Results Status: 200 OK Time: 589 ms Size: 4.45 KB Save as example ...
Pretty Raw Preview Visualize JSON
51   "_id": "7s0Re4Pv2jXtNiWn"
52 },
53 {
54   "name": "User 8169655894016000",
55   "email": "User_Zulauf@yahoo.com",
56   "password": "Pass@123",
57   "administrador": "true",
58   "_id": "AmzJhH9vqvXLxqjZ"
59 },
60 {
61   "name": "Random function () { [native code] }",
62   "email": "Lucas5@gmail.com",
63   "password": "Pass@123",
64   "administrador": "true",
65   "_id": "CfwnBTaf29KAghtE"
66 },
67 {
68   "name": "User 772484379786803200",
69   "email": "User.Feesaat@hotmail.com",
70   "password": "Pass@123",
71   "administrador": "true",
72   "_id": "GHBCV0aXu2C0fny7"
73 },
74 {
75   "name": "User 7724879786803200",
76   "email": "User.Feest@hotmail.com",
```

```
Body Cookies Headers (12) Test Results Status: 200 OK Time: 589 ms Size: 4.45 KB Save as example ...
Pretty Raw Preview Visualize JSON
76   "email": "User.Feest@hotmail.com",
77   "password": "Pass@123",
78   "administrador": "true",
79   "_id": "JK7DAlaX6Mf1z0t8"
80 },
81 {
82   "name": "User 459813492359168",
83   "email": "User.Padberg@hotmail.com",
84   "password": "Pass@123",
85   "administrador": "true",
86   "_id": "bAzNpbU6crFk6q0K"
87 },
88 {
89   "name": "User 4820946493898752",
90   "email": "Lucas.Adams@yahoo.com",
91   "password": "Pass@123",
92   "administrador": "true",
93   "_id": "ck7SEjG4LwIVifoo"
94 },
95 {
96   "name": "Random function () { [native code] }",
97   "email": "Lucas0@hotmail.com",
98   "password": "Pass@123",
99   "administrador": "true",
100   "_id": "jVj6KKnNh1XBHft0"
101 },
```

BodyCookiesHeaders (12)Test Results

Status: 200 OKTime: 589 msSize: 4.45 KBSave as example

PrettyRawPreviewVisualizeJSON

```
102 {
103   "name": "Random function () { [native code] }",
104   "email": "Lucas.Smith@hotmail.com",
105   "password": "Pass@123",
106   "administrador": "true",
107   "_id": "mjGibRKENoRH8YtR"
108 },
109 {
110   "name": "User 998749285384192",
111   "email": "User84@hotmail.com",
112   "password": "Pass@123",
113   "administrador": "true",
114   "_id": "pBK9Spi7BoZFgr9N"
115 },
116 {
117   "name": "User 3207269567168512",
118   "email": "User91@yahoo.com",
119   "password": "Pass@123",
120   "administrador": "true",
121   "_id": "rJSzmRNKTIrNKhZl"
122 },
123 {
124   "name": "User 7916389832589312",
125   "email": "User.Vandervort40@yahoo.com",
126   "password": "Pass@123",
127   "administrador": "true",
128   "id": "rWokwPsj712gBVCz"
```

BodyCookiesHeaders (12)Test Results

Status: 200 OKTime: 589 msSize: 4.45 KBSave as example

PrettyRawPreviewVisualizeJSON

```
105   "password": "Pass@123",
106   "administrador": "true",
107   "_id": "mjGibRKENoRH8YtR"
108 },
109 {
110   "name": "User 998749285384192",
111   "email": "User84@hotmail.com",
112   "password": "Pass@123",
113   "administrador": "true",
114   "_id": "pBK9Spi7BoZFgr9N"
115 },
116 {
117   "name": "User 3207269567168512",
118   "email": "User91@yahoo.com",
119   "password": "Pass@123",
120   "administrador": "true",
121   "_id": "rJSzmRNKTIrNKhZl"
122 },
123 {
124   "name": "User 7916389832589312",
125   "email": "User.Vandervort40@yahoo.com",
126   "password": "Pass@123",
127   "administrador": "true",
128   "_id": "rWokwPsj712gBVCz"
129 }
130 ]
131 }
```

## 2.1 Caso de Teste: Read - Leitura do Usuário Criado

### Descrição:

Este caso de teste tem como objetivo testar a listagem de usuários utilizando o endpoint `/usuarios` da API `https://serverest.dev/`.

### Pré-requisitos:

- Ter acesso à API `https://serverest.dev/`.
- Ter um ambiente de teste disponível.

### Passos:

#### 1. Enviar Requisição GET:

- Método: GET
- URL: `https://serverest.dev/usuarios/dIHTEQw7JDWV4Hzb`
- Headers: Nenhum necessário
- Verificar se a resposta possui status code 200 (OK).
- Verificar se a resposta contém a lista de usuários.

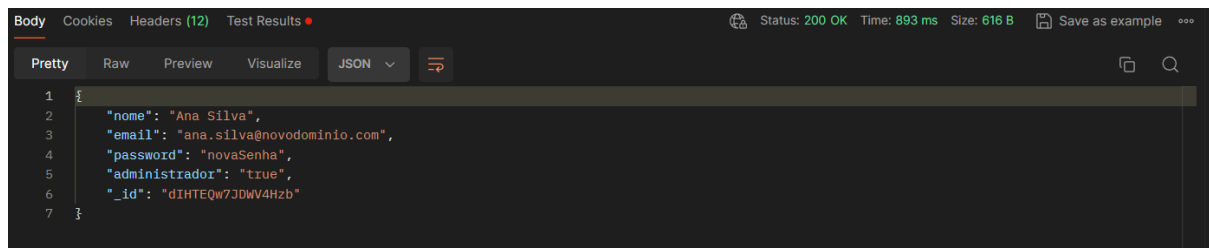
### Critérios de Sucesso:

- A resposta da API deve retornar um status code 200 (OK).
- A resposta deve conter o usuário existente no sistema.

### Critérios de Falha:

- A resposta da API deve retornar um status code diferente de 200 em caso de falha na leitura dos usuários.

### Status Code:



### 3. Caso de Teste: Update - Atualização de Usuário

Descrição:

Este caso de teste tem como objetivo testar a atualização de um usuário utilizando o endpoint `/usuarios/{id}` da API `https://serverest.dev/`.

Pré-requisitos:

- Ter acesso à API `https://serverest.dev/`.
- Ter um ambiente de teste disponível.

Passos:

1. Enviar Requisição PUT:

- Método: PUT
- URL: `https://serverest.dev/usuarios/5PVu2zruuuHGXyeB`
- Headers: Nenhum necessário
- Body:

json

```
{
  "nome": "Ana Silva",
  "email": "ana.silva@novodominio.com",
  "password": "novaSenha",
  "administrador": "true"
}
```

- Verificar se a resposta possui status code 200 (OK).
- Verificar se a resposta contém a mensagem "Registro alterado com sucesso".

Critérios de Sucesso:

- A resposta da API deve retornar um status code 200 (OK).
- A resposta deve conter a mensagem "Registro alterado com sucesso".

Critérios de Falha:

- A resposta da API deve retornar um status code diferente de 200 em caso de falha na atualização do usuário.

Status Code:



#### 4. Caso de Teste: Delete - Exclusão de Usuário

##### Descrição:

Este caso de teste tem como objetivo testar a exclusão de um usuário utilizando o endpoint `/usuarios/{id}` da API `https://serverest.dev/`.

##### Pré-requisitos:

- Ter acesso à API `https://serverest.dev/`.
- Ter um ambiente de teste disponível.

##### Passos:

###### 1. Enviar Requisição DELETE:

- Método: DELETE
- URL: `https://serverest.dev/usuarios/5PVu2zruuuHGXyeB` (substitua `5PVu2zruuuHGXyeB` pelo ID do usuário que deseja excluir)
- Headers: Nenhum necessário
- Verificar se a resposta possui status code 200 (OK).
- Verificar se a resposta contém a mensagem "Registro excluído com sucesso".

##### Critérios de Sucesso:

- A resposta da API deve retornar um status code 200 (OK).
- A resposta deve conter a mensagem "Registro excluído com sucesso".

##### Critérios de Falha:

- A resposta da API deve retornar um status code diferente de 200 em caso de falha na exclusão do usuário.

##### Status Code:

