

# História de Usuário

## Histórico de Revisão

Data	Versão	Descrição	Autor
07/11/2023	1.0	Desafio Matera	Pedro Henrique

## Descrição de escrita (Gherkin)

Dado que

Quando

Então

E

## Cenários de teste

### Exercício 1 – Funcional (Frontend)

Ct 01	<b>Cenário 1:</b> Consulta específica de usuário  <b>Dado</b> que o usuário preencha um ou todos os campos requeridos <b>Quando</b> o usuário filtrar as informações <b>Então</b> o sistema deve retornar os valores informados no filtro
Ct 02	<b>Cenário 2:</b> Consulta total de usuários  <b>Dado</b> que não selecione nenhum filtro <b>Quando</b> filtrar as informações <b>Então</b> deverá retornar todos os usuários cadastrados
Ct 03	<b>Cenário 3:</b> Limpar Filtro  <b>Dado</b> que o usuário preencheu todos os filtros E quer limpar os todos filtros para uma nova busca <b>Quando</b> clicar no botão “Limpar Filtros” <b>Então</b> deverá ser “resetado” todos os filtros pro padrão
Ct 04	<b>Cenário 4:</b> Filtrar o campo nome com números e caracteres especiais  <b>Dado</b> que informe um nome de usuário com número ou caractere especial <b>Quando</b> preencher com algum desses valores

	<b>Então</b> o sistema não deve retornar nada <b>E</b> informar que usuário cadastrado não existe
<b>Ct 05</b>	<b>Cenário 5:</b> Consultar um e-mail inválido ou não cadastrado na base  <b>Dado</b> que informe um e-mail inválido <b>Quando</b> for preenchido esse campo <b>Então</b> o sistema deve informar que e-mail não existe ou não cadastrado na base

## Exercício 2 – Funcional (Backend)

<b>Ct 06</b>	<b>Cenário 1:</b> Consultar um registro único  <b>Dado</b> que queira retornar somente um valor <b>Quando</b> limitar a quantidade de resultados retornados <b>Então</b> o response deve ter somente um valor
<b>Ct 07</b>	<b>Cenário 2:</b> Consultar o limite total de registros  <b>Dado</b> que queira retornar todos os valores cadastrados <b>Quando</b> informar o valor total <b>Então</b> deve ser retornado todos os valores cadastrados
<b>Ct 08</b>	<b>Cenário 3:</b> Consultar a página dois para consultar os registros  <b>Dado</b> que tenha interesse de verificar os registros da página dois <b>Quando</b> passar o parâmetro da página dois <b>Então</b> o sistema deve retornar o total de registros da página dois
<b>Ct 09</b>	<b>Cenário 4:</b> Consultar limite de registros informando valor zero  <b>Dado</b> que informe o valor zero na consulta de registro <b>Quando</b> preencher o valor com zero <b>Então</b> o sistema deverá retornar um erro indicando que o valor não é válido
<b>Ct 10</b>	<b>Cenário 5:</b> Consultar registro com valor "inexistente"  <b>Dado</b> que queira consultar um registro que não existe <b>Quando</b> informar um valor invalido <b>Então</b> deverá retornar uma mensagem de erro

## Protótipo

## Exercício 1

**matera**

USUÁRIOS

Módulo ▾

Consultar

Development Mode  
1.4.0

Consulta de usuários

Filtros

Tipo de pessoa ▾

Nome

E-mail

LIMPAR FILTROS

FILTRAR

## Exercício 2

## Ct06

The screenshot shows a REST client interface with a GET request to `https://catfact.ninja/breeds?limit=1`. The 'Query Params' section shows a single parameter: `limit=1`. The 'Body' section shows the JSON response in 'Pretty' format.

**Request:**

- Method: GET
- URL: `https://catfact.ninja/breeds?limit=1`
- Query Params: `limit=1`

**Response (JSON):**

```
1  {
2    "current_page": 1,
3    "data": [
4      {
5        "breed": "Abyssinian",
6        "country": "Ethiopia",
7        "origin": "Natural/Standard",
8        "coat": "Short",
9        "pattern": "Ticked"
10     }
11  ],
```

## CT07

GET

https://catfact.ninja/breeds?limit=98

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	Key	Value
<input checked="" type="checkbox"/>	limit	98
	Key	Value

Body

Cookies (2)

Headers (16)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
688         "pattern": "Solid"
689     },
690 ],
691 "first_page_url": "https://catfact.ninja/breeds?page=1",
692 "from": 1,
693 "last_page": 1,
694 "last_page_url": "https://catfact.ninja/breeds?page=1",
695 "links": [
696     {
697         "url": null,
698         "label": "Previous",
699         "active": false
700     },
701     {
702         "url": "https://catfact.ninja/breeds?page=1",
703         "label": "1",
704         "active": true
705     },
706     {
707         "url": null,
708         "label": "Next",
709         "active": false
710     }
711 ],
712 "next_page_url": null,
713 "path": "https://catfact.ninja/breeds",
714 "per_page": "98",
715 "prev_page_url": null,
716 "to": 98,
717 "total": 98
718 }
```

## CT08

GET <https://catfact.ninja/breeds?page=2>

Params • Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

Key	Value
<input checked="" type="checkbox"/> page	2
Key	Value

Body Cookies (2) Headers (16) Test Results

Pretty Raw Preview Visualize JSON

```
192     "label": "1",
193     "active": false
194   },
195   {
196     "url": "https://catfact.ninja/breeds?page=2",
197     "label": "2",
198     "active": true
199   },
200   {
201     "url": "https://catfact.ninja/breeds?page=3",
202     "label": "3",
203     "active": false
204   },
205   {
206     "url": "https://catfact.ninja/breeds?page=4",
207     "label": "4",
208     "active": false
209   },
210   {
211     "url": "https://catfact.ninja/breeds?page=3",
212     "label": "Next",
213     "active": false
214   }
215 }
216 "next_page_url": "https://catfact.ninja/breeds?page=3",
217 "path": "https://catfact.ninja/breeds",
218 "per_page": 25,
219 "prev_page_url": "https://catfact.ninja/breeds?page=1",
220 "to": 50,
221 "total": 98
222 }
```

## CT09

HTTP <https://catfact.ninja/breeds?limit=0>

GET <https://catfact.ninja/breeds?limit=0>

Params • Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

	Key	Value
<input checked="" type="checkbox"/>	limit	0
	Key	Value

Body Cookies (2) Headers (16) Test Results

Pretty Raw Preview Visualize JSON

```
137     "label": "4",
138     "active": false
139   },
140   {
141     "url": "https://catfact.ninja/breeds?page=5",
142     "label": "5",
143     "active": false
144   },
145   {
146     "url": "https://catfact.ninja/breeds?page=6",
147     "label": "6",
148     "active": false
149   },
150   {
151     "url": "https://catfact.ninja/breeds?page=7",
152     "label": "7",
153     "active": false
154   },
155   {
156     "url": "https://catfact.ninja/breeds?page=2",
157     "label": "Next",
158     "active": false
159   }
160 ],
161 "next_page_url": "https://catfact.ninja/breeds?page=2",
162 "path": "https://catfact.ninja/breeds",
163 "per_page": 15,
164 "prev_page_url": null,
165 "to": 15,
166 "total": 98
167 }
```

CT10



https://catfact.ninja/breeds?limit=-1

GET



https://catfact.ninja/breeds?limit=-1

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	Key	Value
<input checked="" type="checkbox"/>	limit	-1
	Key	Value

Body

Cookies (2)

Headers (13)

Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "message": "Not Found",
3   "code": 404
4 }
```