

Mapeamento de Issues e melhorias

Resultados dos testes

| Total de testes realizados | Aprovados | Reprovados | % Aprovados | % Reprovados |
|----------------------------|-----------|------------|-------------|--------------|
| 45 | 38 | 7 | 84,4% | 15,6% |

RESUMO DOS TESTES

| ID do Teste | Resultado |
|-------------|--|
| CT001 |  Aprovado |
| CT002 |  Aprovado |
| CT003 |  Reprovado |
| CT004 |  Reprovado |
| CT005 |  Aprovado |
| CT006 |  Reprovado |
| CT007 |  Reprovado |
| CT008 |  Aprovado |
| CT009 |  Aprovado |
| CT010 |  Aprovado |
| CT011 |  Aprovado |
| CT012 |  Aprovado |
| CT013 |  Aprovado |
| CT014 |  Aprovado |
| CT015 |  Aprovado |
| CT016 |  Aprovado |
| CT017 |  Aprovado |
| CT018 |  Aprovado |
| CT019 |  Aprovado |
| CT020 |  Aprovado |
| CT021 |  Aprovado |
| CT022 |  Aprovado |
| CT023 |  Aprovado |

| | |
|-------|-------------|
| CT024 | ✓ Aprovado |
| CT025 | ✓ Aprovado |
| CT026 | ✓ Aprovado |
| CT027 | ✓ Aprovado |
| CT028 | ✓ Aprovado |
| CT029 | ✓ Aprovado |
| CT030 | ✓ Aprovado |
| CT031 | ✓ Aprovado |
| CT032 | ✓ Aprovado |
| CT033 | ✓ Aprovado |
| CT034 | ✓ Aprovado |
| CT035 | ✓ Aprovado |
| CT036 | ✓ Aprovado |
| CT037 | ✓ Aprovado |
| CT038 | ✓ Aprovado |
| CT039 | ✗ Reprovado |
| CT040 | ✗ Reprovado |
| CT041 | ✗ Reprovado |
| CT042 | ✓ Aprovado |
| CT043 | ✓ Aprovado |
| CT044 | ✓ Aprovado |

| ID do Teste | Cenário de Teste | Status |
|---------------------------|-------------------------|------------|
| CT001 | Criar usuário válido | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 201 | Response 201 | |

Evidência (Print)

The screenshot shows the Postman application interface. At the top, it says "ServeRest / Usuarios / Post / CreateUser CT001 - Response 201". Below that, there's a "POST" method dropdown and a URL field containing "({url}) /usuarios". On the left, there are tabs for "Params", "Headers", and "Body". Under "Body", the "JSON" tab is selected, and there is a JSON payload entered:

```

1 {
2   "nome": "Fulano da Silva",
3   "email": "cicilandoqa.com.br",
4   "password": "teste",
5   "administrador": "true"
6 }

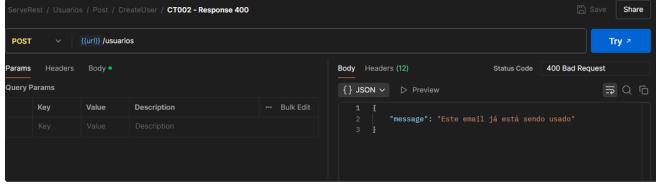
```

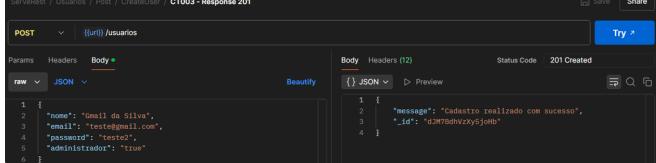
On the right side of the interface, there are tabs for "Body", "Headers (12)", and "Status Code". The "Status Code" tab is selected and shows "201 Created". Below the status code, there is a "Preview" section displaying the response body:

```

1 {
2   "message": "Cadastro realizado com sucesso",
3   "_id": "5f700pyite0MBP10"
4 }

```

| ID do Teste | Cenário de Teste | Status |
|--|--|----------|
| CT002 | Criar usuário com e-mail já cadastrado | Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 Mensagem: “E-mail já cadastrado” | Response 400 Mensagem: “E-mail já cadastrado” | |
| Evidência (Print) | | |
| |  | |

| ID do Teste | Cenário de Teste | Status |
|--|--|-----------|
| CT003 | Criar usuário com domínio gmail.com | Reprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 Não é possível realizar o cadastro com gmail”. | Response 201 | |
| Evidência (Print) | | |
| |  | |

| ID do Teste | Cenário de Teste | Status |
|---------------------------|---|-----------|
| CT004 | Criar usuário com e-mail de domínio “hotmail.com” | Reprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 201 | Response 201 | |
| Evidência (Print) | | |

ServeRest / Usuarios / Post / CreateUser CT001 - Response 201

POST `((url)) /usuarios`

Params Headers Body `JSON` Beautify

```

1 [
2   "name": "Juliano da Silva",
3   "email": "juliano@email.com.br",
4   "password": "teste",
5   "administrador": "true"
6 ]

```

Body Headers (12) Status Code 201 Created

```

{"message": "Cadastro realizado com sucesso",
 "_id": "sIT0q3ViteLONQ10"}

```

Try Save Share

| ID do Teste | Cenário de Teste | Status |
|--------------------|--|---|
| CT005 | Criar usuário com formato de e-mail inválido | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 201 | Response 201 | |
| Evidência (Print) | | |

ServeRest / Usuarios / Post / CreateUser CT005 - Response 400

POST `((url)) /usuarios`

Params Headers Body `JSON` Beautify

```

1 [
2   "name": "Hotmail da Silva",
3   "email": "testemail.com",
4   "password": "teste",
5   "administrador": "true"
6 ]

```

Body Headers (12) Status Code 400 Bad Request

```

{"email": "email deve ser um email valido"}

```

Try Save Share

| ID do Teste | Cenário de Teste | Status |
|--------------------|--|--|
| CT006 | Criar usuário com senha menor que 5 caracteres | ✗ Reprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 201 | |
| Evidência (Print) | | |

CreateUser ((url)) /usuarios / Post / CreateUser

POST `((url)) /usuarios`

Params Auth Headers (0) Body Scripts Settings Cookies

Body `JSON` Beautify 201 Created 545 ms 544 B

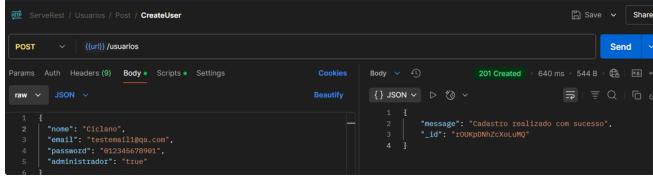
```

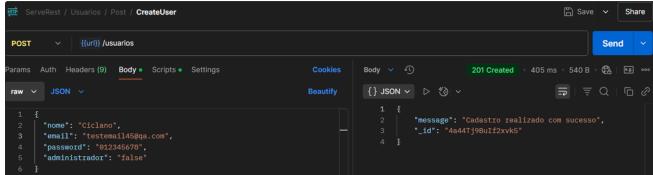
1 [
2   "name": "Hotmail da Silva",
3   "email": "testemail@gu.com",
4   "password": "1",
5   "administrador": "true"
6 ]

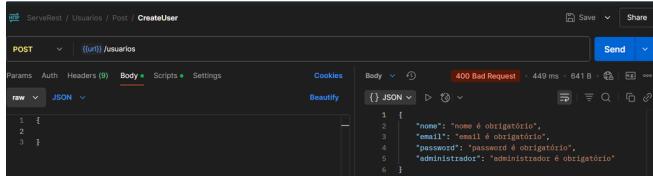
```

Try Save Share

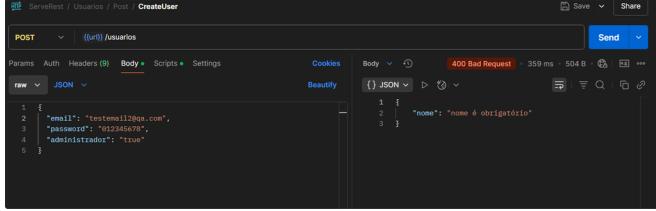
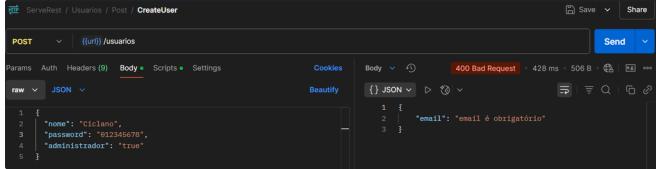
| ID do Teste | Cenário de Teste | Status |
|--------------------|---|--|
| CT007 | Criar usuário com senha maior que 10 caracteres | ✗ Reprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 201 | |

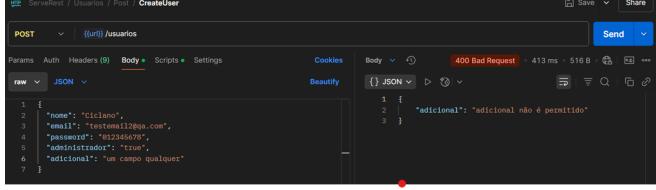
| Evidência (Print) | |
|--|--|
|  | |

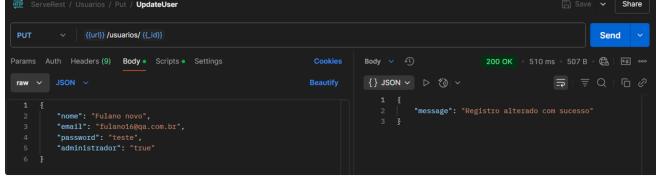
| ID do Teste | Cenário de Teste | Status |
|---------------------------|--|---|
| CT008 | Criar usuário com admin = false | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 201 | Response 201 | |
| Evidência (Print) | | |
| |  | |

| ID do Teste | Cenário de Teste | Status |
|---------------------------|--|---|
| CT009 | Enviar payload vazio | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |
| |  | |

| ID do Teste | Cenário de Teste | Status |
|---------------------------|--|---|
| CT010 | Verificar se os campos obrigatórios (nome, email, senha, admin) são validados corretamente | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |

| Evidência (Print) | | |
|--|--|--|
|  | | |
|  | | |

| ID do Teste | Cenário de Teste | Status |
|---------------------------|--|--|
| CT011 | Validar campos extras ou desnecessário no payload | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |
| |  | |

| ID do Teste | Cenário de Teste | Status |
|---------------------------|--|--|
| CT012 | Atualizar usuário existente com novos dados válido | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 200 | Response 200 | |
| Evidência (Print) | | |
| |  | |

| ID do Teste | Cenário de Teste | Status |
|--------------------|--------------------------------------|------------|
| CT013 | Atualizar usuário com ID inexistente | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 201 | Response 201 | |
| Evidência (Print) | | |

| ID do Teste | Cenário de Teste | Status |
|--------------------|---|------------|
| CT014 | Atualizar usuário com e-mail já existente | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |

```

PUT http://localhost:8080/usuarios/{id}
{
  "name": "Fulano modificado",
  "email": "fulano77@qa.com.br",
  "password": "teste",
  "administrator": "false"
}

```

```

{
  "message": "Este email já está sendo usado"
}

```

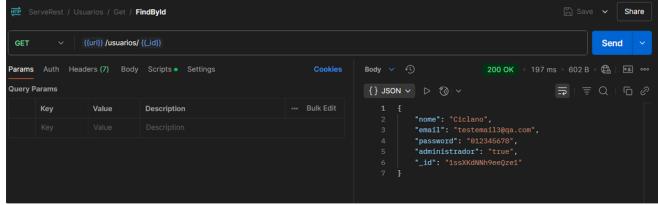
| ID do Teste | Caso de Teste | Status |
|--------------------|--------------------------|------------|
| CT015 | Listar todos os usuários | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 200 | Response 200 | |
| Evidência (Print) | | |

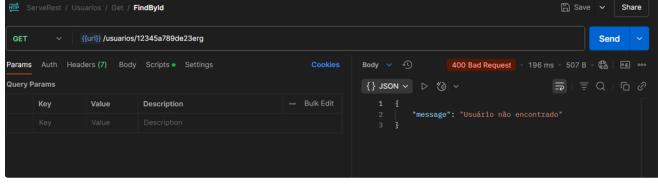
| Key | Value | Description |
|---------------|---------------|-------------|
| _id | (Ld) | |
| nome | Fulano da ... | |
| email | fulano@qa... | |
| password | teste | |
| administrator | true | |
| administrador | false | |

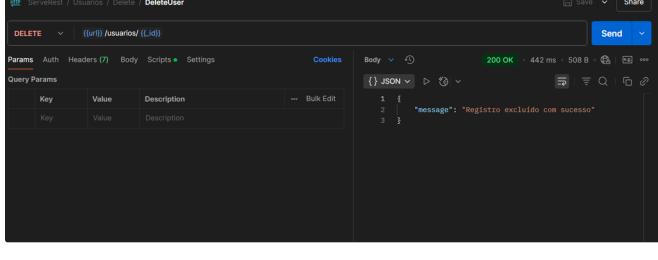
```

{
  "quantidade": 19,
  "usuarios": [
    {
      "name": "Fulano da Silva",
      "email": "fulano@qa.com",
      "password": "teste",
      "administrator": "true",
      "_id": "4e0a0d020000000000000001"
    },
    {
      "name": "Agen QA",
      "email": "Agen_Teste_{{randomInt}}@qa.com",
      "password": "123456",
      "administrator": "true",
      "_id": "4f0a0d020000000000000001"
    },
    {
      "name": "Jorge Bantolim",
      "email": "jorge@bantolim.com",
      "password": "secreta",
      "administrator": "false",
      "_id": "490a0d020000000000000001"
    }
  ]
}

```

| ID do Teste | Caso de Teste | Status |
|--|------------------------------|---|
| CT016 | Listar usuário por ID válido | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 200 | Response 200 | |
| Evidência (Print) | | |
|  | | |

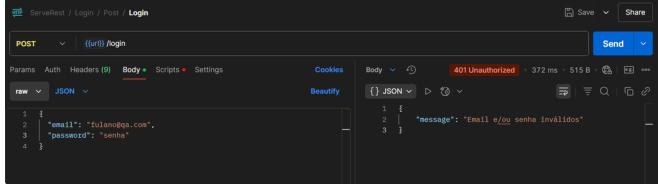
| ID do Teste | Caso de Teste | Status |
|--|-----------------------------------|---|
| CT017 | Listar usuário por ID inexistente | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |
|  | | |

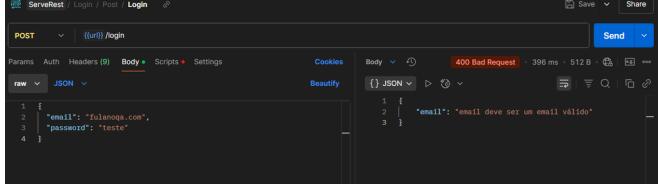
| ID do Teste | Caso de Teste | Status |
|--|---------------------------|---|
| CT018 | Deletar usuário existente | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 200 | Response 200 | |
| Evidência (Print) | | |
|  | | |

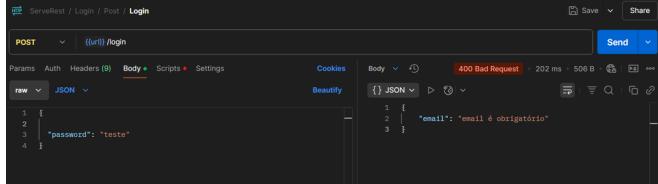
| ID do Teste | Caso de Teste | Status |
|--------------------|--|----------|
| CT019 | Deletar usuário com ID inválido ou inexistente | Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |
| | | |

| ID do Teste | Caso de Teste | Status |
|--------------------|-------------------------------|----------|
| CT020 | Login com credenciais válidas | Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 200 | Response 200 | |
| Evidência (Print) | | |
| | | |

| ID do Teste | Caso de Teste | Status |
|--------------------|---------------------------------|----------|
| CT021 | Login com e-mail não cadastrado | Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |
| | | |

| ID do Teste | Caso de Teste | Status |
|--|---------------------------|---|
| CT022 | Login com senha incorreta | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |
|  | | |

| ID do Teste | Caso de Teste | Status |
|--|--------------------------------------|---|
| CT023 | Login com e-mail em formato inválido | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |
|  | | |

| ID do Teste | Caso de Teste | Status |
|--|------------------------------|---|
| CT024 | Login com payload incompleto | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |
|  | | |

Serverless / Login / Post / Login

POST `((url))/login`

Params Auth Headers (0) Body Scripts Settings Cookies

Body `{} JSON` **Send** `400 Bad Request` 430 ms 512 B

```

1 [
2   "email": "fulano@qa.com"
3 ]
  
```

Body `{} JSON` **Send** `400 Bad Request` 430 ms 512 B

```

1 [
2   "password": "password é obrigatório"
3 ]
  
```

| ID do Teste | Caso de Teste | Status |
|---------------------------|-------------------------|---|
| CT025 | Login com payload vazio | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |

Serverless / Login / Post / Login

POST `((url))/login`

Params Auth Headers (0) Body Scripts Settings Cookies

Body `{} JSON` **Send** `400 Bad Request` 192 ms 550 B

```

1 [
2   "email": "fulano@qa.com"
3 ]
  
```

Body `{} JSON` **Send** `400 Bad Request` 192 ms 550 B

```

1 [
2   "password": "password é obrigatório"
3 ]
4 ]
  
```

| ID do Teste | Caso de Teste | Status |
|---------------------------|---|---|
| CT026 | Verificar se token expira após 10 minutos | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |

Serverless / Login / Post / Login

POST `((url))/login`

Params Auth Headers (0) Body Scripts Settings Cookies

Body `{} JSON` **Send** `200 OK` 188 ms 723 B

```

1 [
2   "email": "fulano@qa.com",
3   "password": "teste"
4 ]
  
```

Body `{} JSON` **Send** `200 OK` 188 ms 723 B

```

1 {
2   "message": "Login realizado com sucesso",
3   "authorization": "Bearer eyJhbGciOiJIaTwkIjew3HgC1IwWcJ9
4   eyJlImphcG1mz1b6fubmBxY5S3b91LLCjwWnzd2pyZC16
5   t4k1c3R1i1iam0101oxhQ2NjQ0DE3LC1ehA10jE3N0V2
6   N0tmt6P,
7   pGDM11gtJ0NGFvK21u4vaCMuAmnnagQm82P93H0"
  
```

Serverless / Produtos / Post / CreateProduct

POST `((url))/produtos`

Params Auth Headers (0) Body Scripts Settings Cookies

Query Params

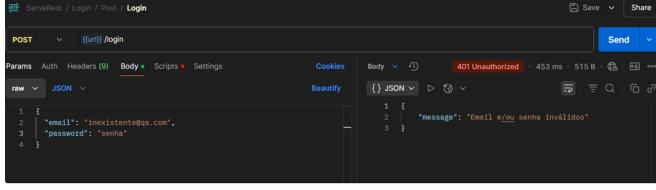
| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

Body `{} JSON` **Send** `401 Unauthorized` 447 ms 571 B

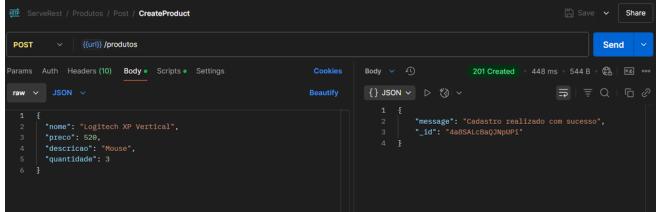
```

1 {
2   "message": "Token de acesso ausente, inválido,
3   expirado ou usuário do token não existe mais"
  
```

| ID do Teste | Caso de Teste | Status |
|--------------------|---|--|
| CT027 | Verificar se mensagem de erro não expõe dado sensível | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |



| ID do Teste | Caso de Teste | Status |
|--------------------|---|--|
| CT028 | Cadastrar produto com usuário autenticado | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 201 | Response 201 | |
| Evidência (Print) | | |



| ID do Teste | Caso de Teste | Status |
|--------------------|---|--|
| CT029 | Cadastrar produto com nome já existente | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |

| ID do Teste | Caso de Teste | Status |
|---------------------------|--|--|
| CT030 | Cadastrar produto sem estar autenticado no sistema | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |

| ID do Teste | Caso de Teste | Status |
|---------------------------|--|--|
| CT030 | Cadastrar produto sem estar autenticado no sistema | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |

| ID do Teste | Caso de Teste | Status |
|---------------------------|--|--|
| CT031 | Cadastrar produto com campos obrigatórios ausentes | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |

| ID do Teste | Caso de Teste | Status |
|---------------------------|--------------------------------------|--|
| CT032 | Atualizar produto com ID inexistente | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 201 | Response 201 | |

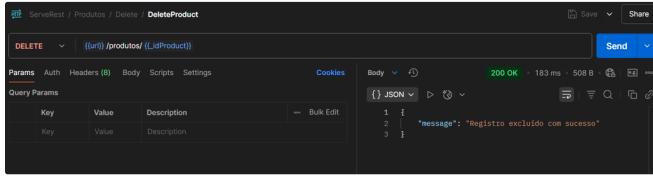
| Evidência (Print) | | |
|-------------------|--|--|
| | | |

| ID do Teste | Caso de Teste | Status |
|--------------------|---|--|
| CT033 | Atualizar produto com nome já utilizado | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |
| | | |

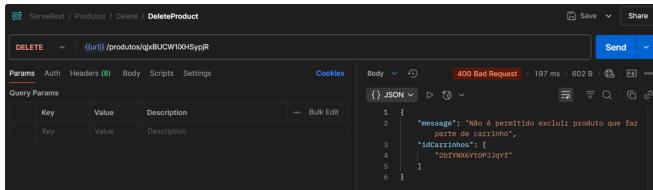
| ID do Teste | Caso de Teste | Status |
|--------------------|--------------------------------------|--|
| CT034 | Atualizar produto com token inválido | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |
| | | |

| ID do Teste | Caso de Teste | Status |
|--------------------|--|--|
| CT035 | Excluir produto existente e fora do carrinho | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |

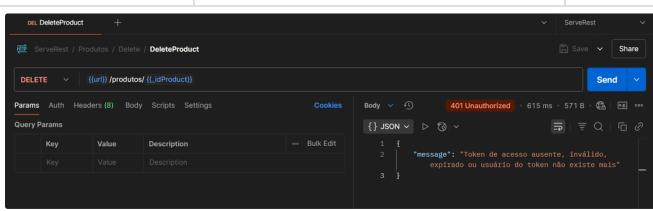
| | | |
|--------------------------|--------------|--|
| Response 200 | Response 200 | |
| Evidência (Print) | | |



| ID do Teste | Caso de Teste | Status |
|---------------------------|--|--|
| CT036 | Tentar excluir produto que está em um carrinho ativo | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |

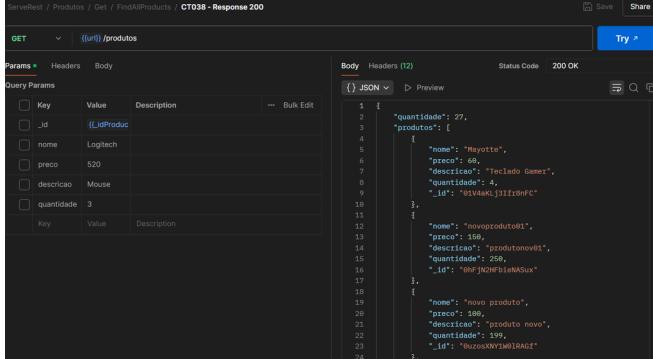


| ID do Teste | Caso de Teste | Status |
|---------------------------|---|--|
| CT037 | Tentar excluir produto sem autenticação | <input checked="" type="checkbox"/> Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |



| ID do Teste | Caso de Teste | Status |
|---------------------------|--------------------------|------------------------------------|
| CT038 | Listar todos os produtos | <input type="checkbox"/> Reprovado |
| Resultado Esperado | Resultado Obtido | OBS: |

| | | |
|--------------------------|--------------|---|
| Response 401 | Response 200 | Rota não deveria estar acessível para usuário não autenticado |
| Evidência (Print) | | |

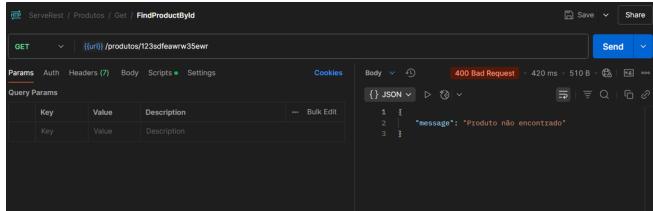


```

1  [
2    {
3      "quantidade": 27,
4      "produtos": [
5        {
6          "nome": "Mayotte",
7          "preco": 69,
8          "descricao": "Telefones Samui",
9          "quantidade": 4,
10         "_id": "69d446LJ3Iz8nFc"
11       },
12       {
13         "nome": "Novo produto01",
14         "preco": 159,
15         "descricao": "produto novo01",
16         "quantidade": 258,
17         "_id": "69d446Hf1ieASuX"
18     },
19     {
20       "nome": "novo produto",
21       "preco": 150,
22       "descricao": "produto novo",
23       "quantidade": 199,
24       "_id": "69d446XY3mWIR4cZ"
25     }
26   ]
27 }
  
```

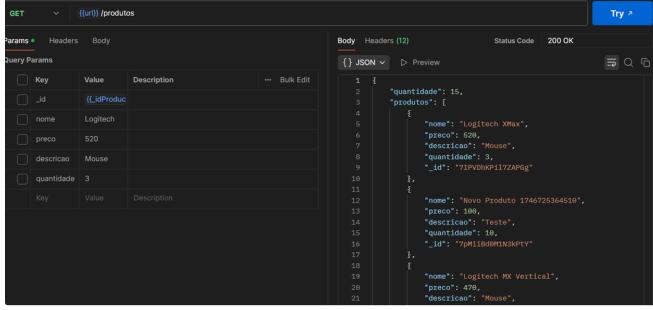
| ID do Teste | Caso de Teste | Status |
|---------------------------|-------------------------|---|
| CT039 | Buscar por ID válido | 🔴 Reprovado |
| Resultado Esperado | Resultado Obtido | OBS: |
| Response 401 | Response 200 | Rota não deveria estar acessível para usuário não autenticado |
| Evidência (Print) | | |

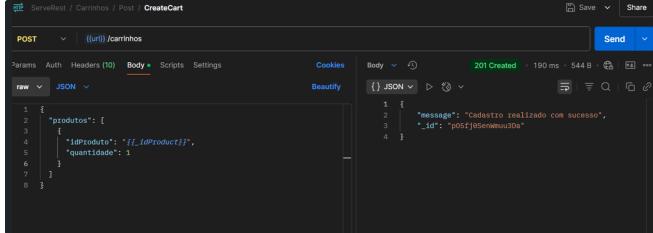
| ID do Teste | Caso de Teste | Status |
|---------------------------|--------------------------------|---|
| CT040 | Buscar produto por ID inválido | 🔴 Reprovado |
| Resultado Esperado | Resultado Obtido | OBS: |
| Response 401 | Response 400 | Rota não deveria estar acessível para usuário não autenticado |
| Evidência (Print) | | |



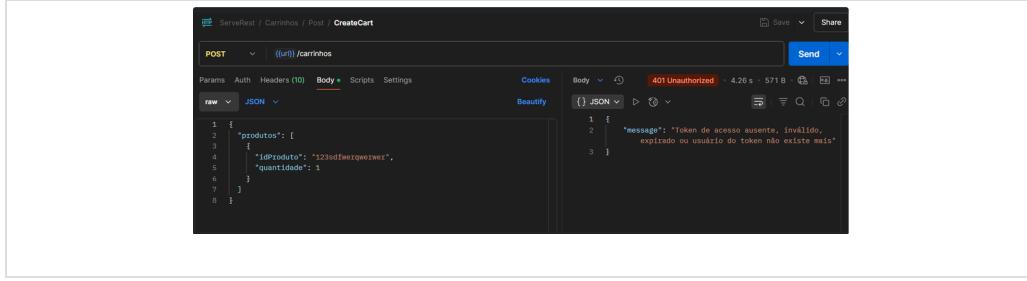
```

1  [
2    {
3      "message": "Produto não encontrado"
4    }
5  ]
  
```

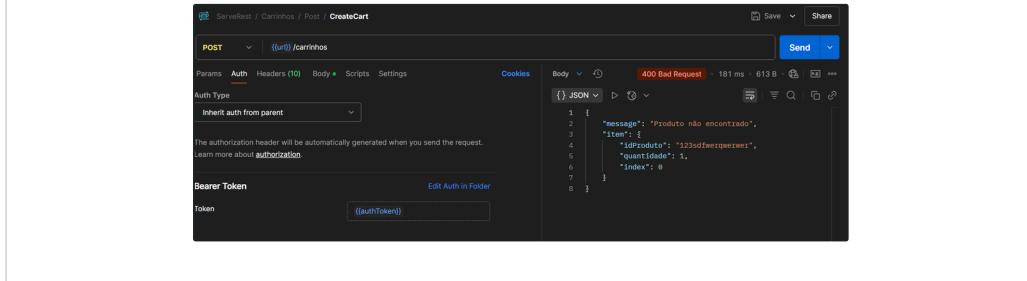
| ID do Teste | Caso de Teste | Status |
|--|----------------------------------|---|
| CT041 | Listar produtos e token inválido | ✖️ Reprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 200 | |
| Evidência (Print) | | |
|  <pre> GET (url) /produtos Params Headers Body Query Params Key Value Description _id (LdProduct) nome Logitech preco 520 descricao Mouse quantidade 3 Body Headers (12) Status Code 200 OK {} JSON D Preview 1 [2 { 3 "quantidade": 3, 4 "produtos": [5 { 6 "name": "logitech MX", 7 "preco": 520, 8 "descricao": "Mouse", 9 "quantidade": 3, 10 "_id": "71PV0DKPL17Z2PG" 11 }, 12 { 13 "name": "Novo Produto 1746725364538", 14 "preco": 100, 15 "descricao": "Teste", 16 "quantidade": 10, 17 "_id": "7jMIL0HMIN1NQPy" 18 }, 19 { 20 "name": "Logitech MX Vertical", 21 "preco": 499, 22 "descricao": "Mouse" 23 }] </pre> | | |

| ID do Teste | Caso de Teste | Status |
|--|--|---|
| CT042 | Criar carrinho com um ou mais produtos válidos | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 201 | Response 201 | |
| Evidência (Print) | | |
|  <pre> POST (url) /carrinhos Params Auth Headers (10) Body Scripts Settings Cookies raw JSON Body POST /carrinhos { "produtos": [{ "idProduto": "{\$_idProduct}", "quantidade": 1 }] } Body Headers (12) Status Code 201 Created 190 ms 544 B {} JSON D Preview 1 [2 { 3 "message": "Cadastro realizado com sucesso", 4 "_id": "postfG5enmnuuDo" 5 }] </pre> | | |

| ID do Teste | Caso de Teste | Status |
|--------------------|---|---|
| CT043 | Criar carrinho sem estar autenticado no sistema | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 401 | Response 401 | |
| Evidência (Print) | | |



| ID do Teste | Caso de Teste | Status |
|---------------------------|--|---|
| CT044 | Criar carrinho com produto inexistente | ✓ Aprovado |
| Resultado Esperado | Resultado Obtido | |
| Response 400 | Response 400 | |
| Evidência (Print) | | |



Reporte de Bugs e Sugestões de Melhoria ↗

RB 001 - Função cadastrar usuário permite cadastro de usuário com domínio de e-mail inválido ↗

1- Descrição detalhada

Ao realizar o teste da rota POST/usuarios verificou-se que é permitido o cadastramento de um usuário com domínio de email inválido (@gmail.com e @hotmail.com).

2- Passo a passo

1. Acessar a rota POST/usuarios
2. Enviar o payload (modelo):

```

1 {
2   "nome": "Gmail da Silva",
3   "email": "teste@gmail.com",
4   "password": "teste2",
5   "administrador": "true"
6 }
```

```

1 {
2   "nome": "Hotmail da Silva",
3   "email": "teste@hotmail.com",
4   "password": "teste2",
5   "administrador": "true"
```

3- Resultado esperado

O sistema deveria exibir uma mensagem de erro ao usuário informando que não é possível realizar o cadastro com e-mails de provedor gmail e hotmail.

4- Resultado Atual

O sistema permitiu o cadastro e retornou o status code 201 Created.

5 - Gravidade

Alta, pois fere a regra de negócio do cliente.

6 - Prioridade

Imediata.

7 - Ambiente onde ocorre

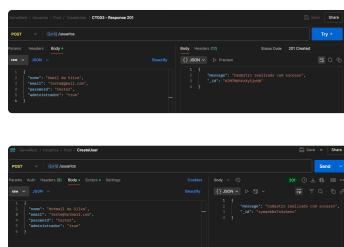
API Serverest

ROTA POST/usuarios

8 - Responsável

A ser definido pela equipe de desenvolvimento

9 - Anexos



10 - Relação com outros defeitos ou requisitos

Não foram identificadas relações com outros defeitos ou requisitos.

RB 002 - Função cadastrar usuário permite a criação de senhas com menos de 5 caracteres e com mais de 10 caracteres ↗

1- Descrição detalhada

Ao realizar o teste da rota POST/usuarios verificou-se que é permitido o cadastramento de um usuário cujo password contenha menos de 5 caracteres ou mais de 10 caracteres, o que, segundo a regra de negócio, não deveria ser permitido.

2- Passo a passo

1. Acessar a rota POST/usuarios
2. Enviar o payload (modelo):

```

1  {
2    "nome": "Hotmail da Silva",
3    "email": "testemail@qa.com",
4    "password": "t",
5    "administrador": "true"
6  }

```

```

1  {

```

```

2   "nome": "Ciclano",
3   "email": "testemail1@qa.com",
4   "password": "01234567890123",
5   "administrador": "true"
6 }
```

3- Resultado esperado

O sistema deveria exibir uma mensagem de erro ao usuário informando que a senha deve conter entre 6 a 10 caracteres.

4- Resultado Atual

O sistema permitiu o cadastro e retornou o status code 201 Created.

5 - Gravidade

Alta, pois fere a regra de negócio do cliente.

6 - Prioridade

Imediata.

7 - Ambiente onde ocorre

API Serverest

ROTA POST/usuarios

8 - Responsável

A ser definido pela equipe de desenvolvimento

9 - Anexos

The screenshot shows a Postman interface for a 'CreateUser' endpoint. The request method is POST, and the URL is {{url}}/usuarios. The request body is a JSON object with fields: nome, email, password, and administrador. The response status is 201 Created, and the response body is a JSON object with message: "Cadastro realizado com sucesso" and _id: "rOUKpDNhZcXoLuMQ".

This screenshot shows a second Postman request for the same 'CreateUser' endpoint. The request body is identical to the first one. The response status is 201 Created, and the response body is a JSON object with message: "Cadastro realizado com sucesso" and _id: "cDx6S0bTVD7imwNe".

10 - Relação com outros defeitos ou requisitos

Não foram identificadas relações com outros defeitos ou requisitos.

RB003 - Response 400 da rota /login não mapeada na documentação

1- Descrição detalhada

Ao realizar o teste da rota POST/login verificou-se que não há o mapeamento da Response 400 Bad Request com a mensagem de erro na documentação.

2- Passo a passo

1. Acessar a rota POST/login
2. Enviar o payload (modelos):

```
1 {
2   "email": "fulano.qa.com", "password": "teste"
3 }
```

```
1 {}
```

3- Resultado esperado

A documentação deveria prever a response 400 Bad Request e a mensagem de erro específica para cada caso.

4- Resultado Atual

Não há registro do status code 400 na documentação

5 - Gravidade

Baixa.

6 - Prioridade

Baixa.

7 - Ambiente onde ocorre

API Serverest

ROTA POST/login

8 - Responsável

A ser definido pela equipe de desenvolvimento

9 - Anexos

The screenshot shows the API documentation for the POST /login endpoint. It includes parameters (user object), responses (200 OK and 401 Unauthorized), and a preview of the JSON response body.

Parameters:

```
POST /login Realizar login
A duração do token retornado em autorização é de 600 segundos (10 minutos). Caso esteja expirado irá receber status code 401 (Unauthorized).
Parameters
Name Description
user required Example Value: Model
object (Model)
[{"email": "fulanoqa.com", "password": "teste"}]
Parameter content type application/json
```

Responses:

| Code | Description | Response content type |
|------|------------------------------|-----------------------|
| 200 | Login realizado com sucesso | application/json |
| 401 | E-mail ou/ou senha inválidos | application/json |

Example 200:

```
{"token": "token realizado com sucesso", "refreshToken": "refresh token", "user": {"email": "fulanoqa.com", "password": "teste", "id": 1}}
```

Example 401:

```
{"message": "Email e/ou senha inválidos"}
```

The screenshot shows a failed POST /login request in Postman. The status code is 400 Bad Request, and the error message is 'email deve ser um email válido'.

Params:

| Key | Value | Description |
|-------|--------------|-------------|
| email | fulanoqa.com | |

Body:

```
email: "fulanoqa.com"
```

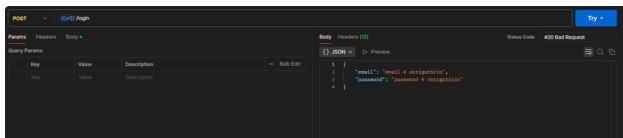
The screenshot shows a failed POST /login request in Postman. The status code is 400 Bad Request, and the error message is 'email é obrigatório'.

Params:

| Key | Value | Description |
|-------|-------|-------------|
| email | | |

Body:

```
email: null
```



10 - Relação com outros defeitos ou requisitos

Não foram identificadas relações com outros defeitos ou requisitos.

RB004 - Usuário consegue acessar a rota GET/produtos mesmo com token inválido ↗

1- Descrição detalhada

Ao realizar o teste da rota GET/produtos verificou-se que o usuário consegue acessar a rota mesmo com um token inválido, o que fere a regra de negócio de que “usuários não autenticados não devem conseguir realizar ações na rota de Produtos”.

2- Passo a passo

1. Acessar a rota GET/produtos
2. Enviar um Bearer Token inválido

3- Resultado esperado

O usuário não deveria conseguir acessar a rota.

4- Resultado Atual

O usuário obtém a lista de produtos mesmo sem estar com o token válido.

5 - Gravidade

Muito Alta, pois fere a regra de negócio da aplicação.

6 - Prioridade

Muito Alta.

7 - Ambiente onde ocorre

API Serverest

ROTA GET/produtos

8 - Responsável

A ser definido pela equipe de desenvolvimento

9 - Anexos



10 - Relação com outros defeitos ou requisitos

Não foram identificadas relações com outros defeitos ou requisitos.