

Relatório Registro de Vendas --- Erro duplicação de quantidade no Registro de Vendas

Clientes utilizados para testes:

GET

http://localhost:3000/cliente

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 21 msSize: 981 BSave as example

PrettyRawPreviewVisualizeJSON

```
11  },
12  {
13    "id": 2,
14    "cpf": "01234567890",
15    "email": "cliente22222@email.com",
16    "nome": "Nome do Cliente 22222",
17    "telefone": "(22) 9876-5432",
18    "endereco": "Endereço do Cliente 2222",
19    "data_nascimento": "2005-05-15T00:00:00.000Z",
20    "data_criacao": "2024-04-14T03:16:31.651Z"
21  },
22  {
23    "id": 4,
24    "cpf": "12345678901",
25    "email": "cliente123@example.com",
26    "nome": "Ana Silva",
27    "telefone": "(31) 9876-5432",
28    "endereco": "Rua das Palmeiras, 789",
29    "data_nascimento": "2000-09-10T00:00:00.000Z",
30    "data_criacao": "2024-04-14T03:39:58.606Z"
31  }
32 ]
```

Fornecedor utilizado nos testes:

bayarea / fornecedor / todos

Save

GET

http://localhost:3000/fornecedor

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 22 msSize: 451 BSave as example

PrettyRawPreviewVisualizeJSON

```
1  {
2    "id": 1,
3    "nome_fantasia": "Nome Fantasia ABC",
4    "cnpj": "12345678901234",
5    "tipo_pessoa": "PJ",
6    "razao_social": "Razão Social XYZ",
7    "inscricao_estadual": "123456",
8    "data_registro": "01/01/2023"
9  }
10
11
```

Produtos utilizados nos testes:

bayarea / produto / id

GET

http://localhost:3000/produto/6

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 17 msSize: 571 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 6,
3   "nome": "Arroz Integral",
4   "marca": "Tio João",
5   "descricao": "Arroz integral rico em fibras e nutrientes",
6   "nome_categoria": "Grãos e Cereais",
7   "cnpj_fornecedor": "12345678901234",
8   "codigo_produto": "ARZ123",
9   "unidade_medida": "kg",
10  "preco": 9.99,
11  "estoque_atual": 50,
12  "data_criacao": "2024-04-14T03:21:53.845Z"
13 }
```

GET

http://localhost:3000/produto/7

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 33 msSize: 604 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 7,
3   "nome": "Sabonete Líquido Refrescante",
4   "marca": "Natura",
5   "descricao": "Sabonete líquido com fragrância refrescante de menta e hortelã",
6   "nome_categoria": "Higiene Pessoal",
7   "cnpj_fornecedor": "12345678901234",
8   "codigo_produto": "SAB789",
9   "unidade_medida": "L",
10  "preco": 15.5,
11  "estoque_atual": 30,
12  "data_criacao": "2024-04-14T03:23:39.006Z"
13 }
```

bayarea / produto / id

GET

http://localhost:3000/produto/8

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

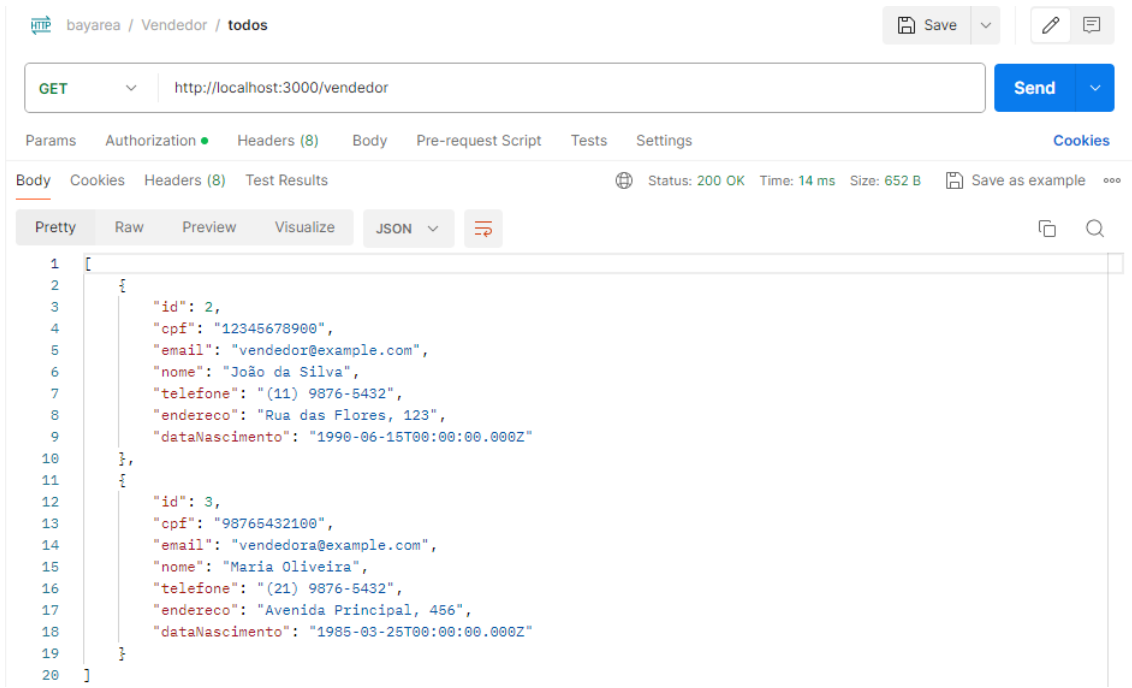
BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 12 msSize: 563 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 8,
3   "nome": "Leite Desnatado",
4   "marca": "Nestlé",
5   "descricao": "Leite desnatado com baixo teor de gordura",
6   "nome_categoria": "Laticínios",
7   "cnpj_fornecedor": "12345678901234",
8   "codigo_produto": "LEI456",
9   "unidade_medida": "L",
10  "preco": 3.99,
11  "estoque_atual": 80,
12  "data_criacao": "2024-04-14T03:24:11.441Z"
13 }
```

Vendedores utilizados nos testes:



bayarea / Vendedor / todos

GET http://localhost:3000/vendedor

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

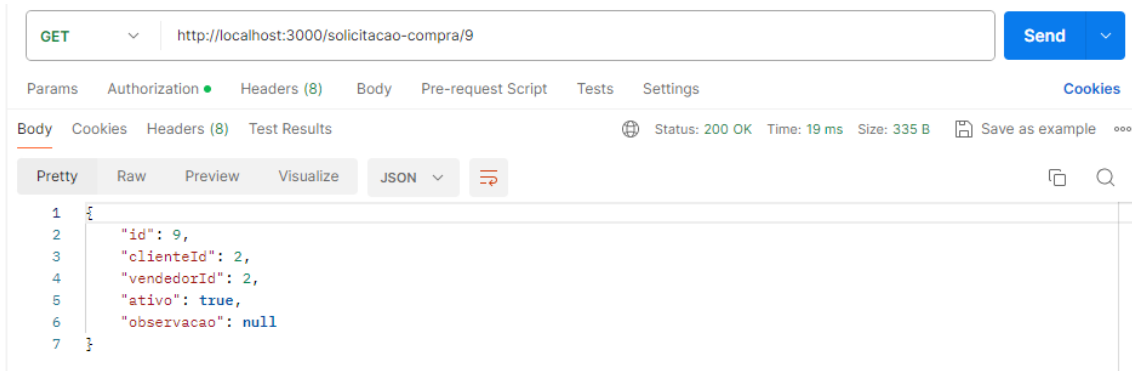
Body Cookies Headers (8) Test Results Status: 200 OK Time: 14 ms Size: 652 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 2,
4     "cpf": "12345678900",
5     "email": "vendedor@example.com",
6     "nome": "João da Silva",
7     "telefone": "(11) 9876-5432",
8     "endereco": "Rua das Flores, 123",
9     "dataNascimento": "1990-06-15T00:00:00.000Z"
10  },
11  {
12    "id": 3,
13    "cpf": "98765432100",
14    "email": "vendedora@example.com",
15    "nome": "Maria Oliveira",
16    "telefone": "(21) 9876-5432",
17    "endereco": "Avenida Principal, 456",
18    "dataNascimento": "1985-03-25T00:00:00.000Z"
19  }
20 ]
```

Processo de Registro:

1. Criar solicitação de compra entre os clientes e vendedores (**cliente id 2 com vendedor id 2** | **cliente id 4 e vendedor id 3**)



GET http://localhost:3000/solicitacao-compra/9

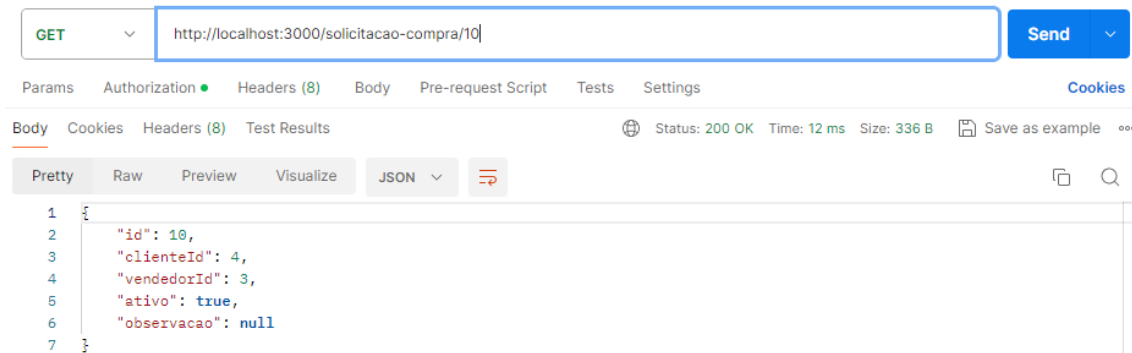
Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 19 ms Size: 335 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 9,
3   "clienteId": 2,
4   "vendedorId": 2,
5   "ativo": true,
6   "observacao": null
7 }
```



GET http://localhost:3000/solicitacao-compra/10

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 12 ms Size: 336 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 10,
3   "clienteId": 4,
4   "vendedorId": 3,
5   "ativo": true,
6   "observacao": null
7 }
```

2. Criar solicitação de produtos (**Verificando erro de duplicação**)

bayarea / Solicitação produto / criar

POST http://localhost:3000/produto-solicitacao

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "solicitacaoCompraId": 9,
3   "codigo_produto": "ARZ123",
4   "quantidade": 2,
5   "controle": "Controle opcional",
6   "descricao": "Descrição opcional"
7 }
```

Body Cookies Headers (8) Test Results Status: 201 Created Time: 41 ms Size: 486 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 19,
3   "solicitacaoCompraId": 9,
4   "codigo_produto": "ARZ123",
5   "nome_produto": "Arroz Integral",
6   "quantidade": 2,
7   "valor_unit": 9.99,
8   "valor_subtotal": 19.98,
9   "controle": "Controle opcional",
10  "descricao": "Descrição opcional"
11 }
```

bayarea / Solicitação produto / criar

POST http://localhost:3000/produto-solicitacao

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "solicitacaoCompraId": 9,
3   "codigo_produto": "SAB789",
4   "quantidade": 2,
5   "controle": "Controle opcional",
6   "descricao": "Descrição opcional"
7 }
```

Body Cookies Headers (8) Test Results Status: 201 Created Time: 24 ms Size: 498 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 20,
3   "solicitacaoCompraId": 9,
4   "codigo_produto": "SAB789",
5   "nome_produto": "Sabonete Líquido Refrescante",
6   "quantidade": 2,
7   "valor_unit": 15.5,
8   "valor_subtotal": 31,
9   "controle": "Controle opcional",
10  "descricao": "Descrição opcional"
11 }
```

bayarea / Solicitação produto / criar

POST http://localhost:3000/produto-solicitacao

Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "solicitacaoCompraId": 9,
3   "codigo_produto": "LEI456",
4   "quantidade": 2,
5   "controle": "Controle opcional",
6   "descricao": "Descrição opcional"
7 }
```

Body Cookies Headers (8) Test Results Status: 201 Created Time: 32 ms Size: 486 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 21,
3   "solicitacaoCompraId": 9,
4   "codigo_produto": "LEI456",
5   "nome_produto": "Leite Desnatado",
6   "quantidade": 2,
7   "valor_unit": 3.99,
8   "valor_subtotal": 7.98,
9   "controle": "Controle opcional",
10  "descricao": "Descrição opcional"
11 }
```

3. Verificar duplicidade de produtos

bayarea / Solicitação produto / Por solicitação

GET http://localhost:3000/produto-solicitacao/solicitacao/9

Send

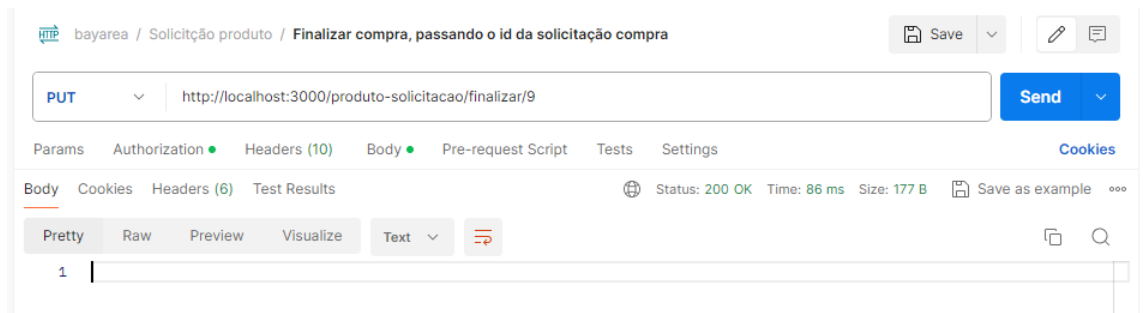
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 22 ms Size: 924 B Save as example

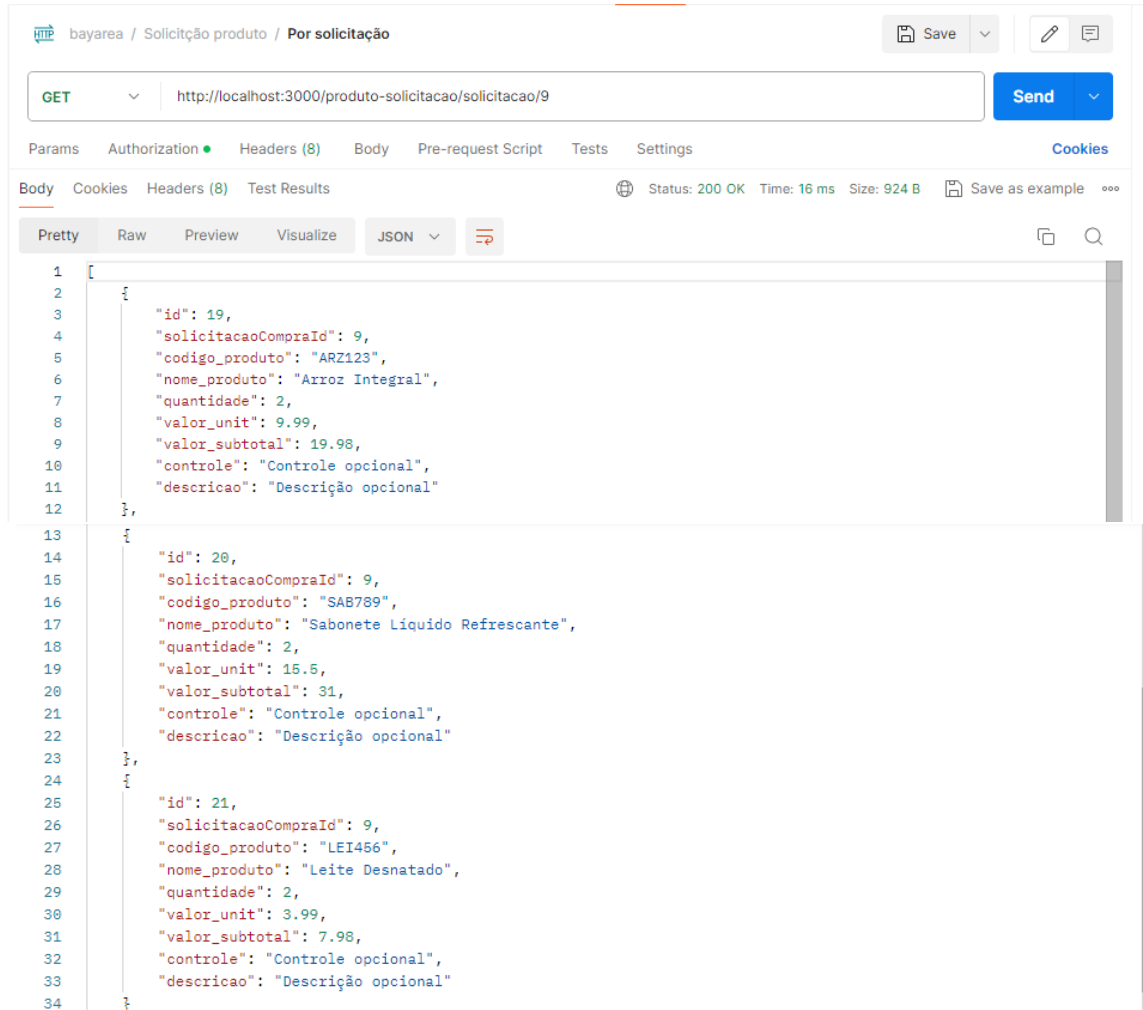
Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 19,
4     "solicitacaoCompraId": 9,
5     "codigo_produto": "ARZ123",
6     "nome_produto": "Arroz Integral",
7     "quantidade": 2,
8     "valor_unit": 9.99,
9     "valor_subtotal": 19.98,
10    "controle": "Controle opcional",
11    "descricao": "Descrição opcional"
12  },
13  {
14    "id": 20,
15    "solicitacaoCompraId": 9,
16    "codigo_produto": "SAB789",
17    "nome_produto": "Sabonete Líquido Refrescante",
18    "quantidade": 2,
19    "valor_unit": 15.5,
20    "valor_subtotal": 31,
21    "controle": "Controle opcional",
22    "descricao": "Descrição opcional"
23  },
24  {
25    "id": 21,
26    "solicitacaoCompraId": 9,
27    "codigo_produto": "LEI456",
28    "nome_produto": "Leite Desnatado",
29    "quantidade": 2,
30    "valor_unit": 3.99,
31    "valor_subtotal": 7.98,
32    "controle": "Controle opcional",
33    "descricao": "Descrição opcional"
34  }
35 ]
```

4. Finalize a compra



5. Verifique a quantidade dos produtos após a finalização



Conclusão: Após vários testes, não foi encontrado problema de duplicação após o produto ser inserido e após a compra for finalizada.

Possível motivo da causa do “erro de duplicação”:

1. Utilizando a outra solicitação de compra (id 10), colocarei 2 produtos (Arroz e Sabonete líquido) no “carrinho”, ambos com quantidade 2

The image displays two screenshots of a REST client interface, likely Postman, showing the process of creating products in a system. Both screenshots are for a POST request to the endpoint `http://localhost:3000/produto-solicitacao`.

Top Screenshot:

- Request Body (JSON):**

```
1 {
2   "solicitacaoCompraId": 10,
3   "codigo_produto": "ARZ123",
4   "quantidade": 2,
5   "controle": "Controle opcional",
6   "descricao": "Descrição opcional"
7 }
```
- Response Body (JSON):**

```
1 {
2   "id": 22,
3   "solicitacaoCompraId": 10,
4   "codigo_produto": "ARZ123",
5   "nome_produto": "Arroz Integral",
6   "quantidade": 2,
7   "valor_unit": 9.99,
8   "valor_subtotal": 19.98,
9   "controle": "Controle opcional",
10  "descricao": "Descrição opcional"
11 }
```
- Status:** 201 Created, Time: 52 ms, Size: 487 B.

Bottom Screenshot:

- Request Body (JSON):**

```
1 {
2   "solicitacaoCompraId": 10,
3   "codigo_produto": "SAB789",
4   "quantidade": 2,
5   "controle": "Controle opcional",
6   "descricao": "Descrição opcional"
7 }
```
- Response Body (JSON):**

```
1 {
2   "id": 23,
3   "solicitacaoCompraId": 10,
4   "codigo_produto": "SAB789",
5   "nome_produto": "Sabonete Liquido Refrescante",
6   "quantidade": 2,
7   "valor_unit": 15.5,
8   "valor_subtotal": 31,
9   "controle": "Controle opcional",
10  "descricao": "Descrição opcional"
11 }
```
- Status:** 201 Created, Time: 20 ms, Size: 499 B.

2. Em seguida, colocarei novamente o Arroz integral de código “ARZ123”, com quantidade 2 novamente

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:3000/produto-solicitacao`
- Method:** `POST`
- Body (Request):**

```
1 {
2   "solicitacaoCompraId": 10,
3   "codigo_produto": "ARZ123",
4   "quantidade": 2,
5   "controle": "Controle opcional",
6   "descricao": "Descrição opcional"
7 }
```
- Status:** 201 Created
- Time:** 42 ms
- Size:** 487 B
- Body (Response):**

```
1 {
2   "id": 22,
3   "solicitacaoCompraId": 10,
4   "codigo_produto": "ARZ123",
5   "nome_produto": "Arroz Integral",
6   "quantidade": 4,
7   "valor_unit": 9.99,
8   "valor_subtotal": 39.96,
9   "controle": "Controle opcional",
10  "descricao": "Descrição opcional"
11 }
```

OBS: Anteriormente tinha 2, pois eu inserir 2 no “carrinho”, contudo após inserir novamente mais 2 (Arroz integral), o programa incrementou 2 na quantidade, ficando 4.

Conclusão: Provavelmente no caso de teste que foi diagnosticado um “erro de duplicidade”, já tinha instâncias do produto no “carrinho” (solicitação de compra entre cliente e vendedor), assim ao inserir novamente, ele apenas incrementou na quantidade.