

TESTES DE API COM INSOMNIA

O sistema foi amplamente testado com a ferramenta **Insomnia**, com foco na simulação de operações reais realizadas por pequenas e médias empresas. Todos os endpoints foram organizados de forma modular para garantir agilidade e clareza nos testes.

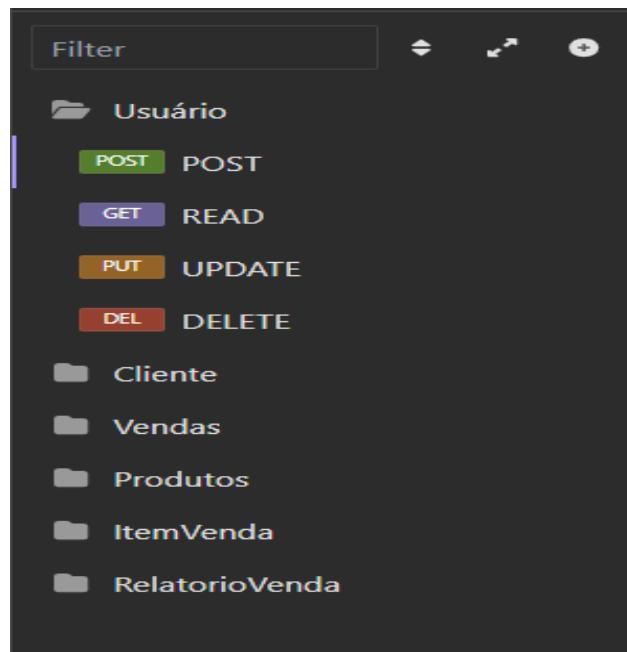
Configuração do Ambiente de Testes

As requisições foram agrupadas por entidade, facilitando a manutenção e testes recorrentes. Para cada entidade (Usuários, Clientes, Vendas, Produtos, ItensVenda, RelatorioVenda), foram criadas pastas com os seguintes métodos:

- **POST** – Criar novo registro
- **GET** – Buscar todos ou por ID
- **PUT** – Atualizar
- **DELETE** – Remover

Testando Operações CRUD

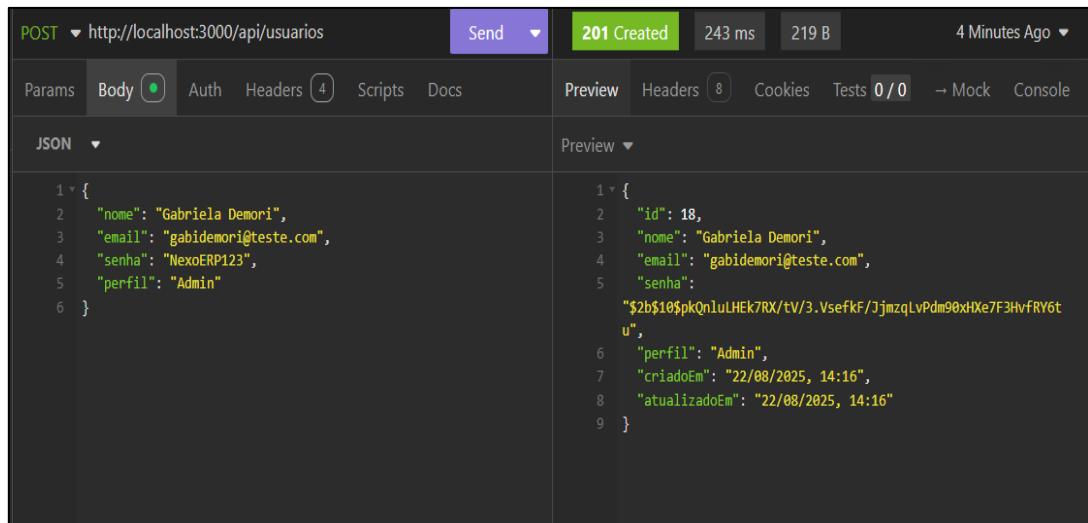
Estrutura organizada por entidade: cada conjunto de requisições (POST, GET, PUT, DELETE) foi agrupado em pastas específicas para facilitar a manutenção, a leitura e a execução recorrente dos testes da API.



Usuários

POST – Criar novo usuário

- ✓ Teste de criação de novo usuário via requisição POST. Registro criado com sucesso e retorno contendo o ID do novo usuário.



The screenshot shows a Postman request for a POST operation at `http://localhost:3000/api/usuarios`. The response status is `201 Created` with a response time of 243 ms and a body size of 219 B, timestamped 4 Minutes Ago. The request body is a JSON object with fields: nome, email, senha, and perfil. The response body is also a JSON object containing the created user's details, including id, nome, email, senha, and timestamps for creation and update.

Params	Body	Auth	Headers	Scripts	Docs	Preview	Headers	Cookies	Tests	→ Mock	Console
						Preview	Headers 8	Cookies	Tests 0 / 0	→ Mock	Console

Request Body (JSON):

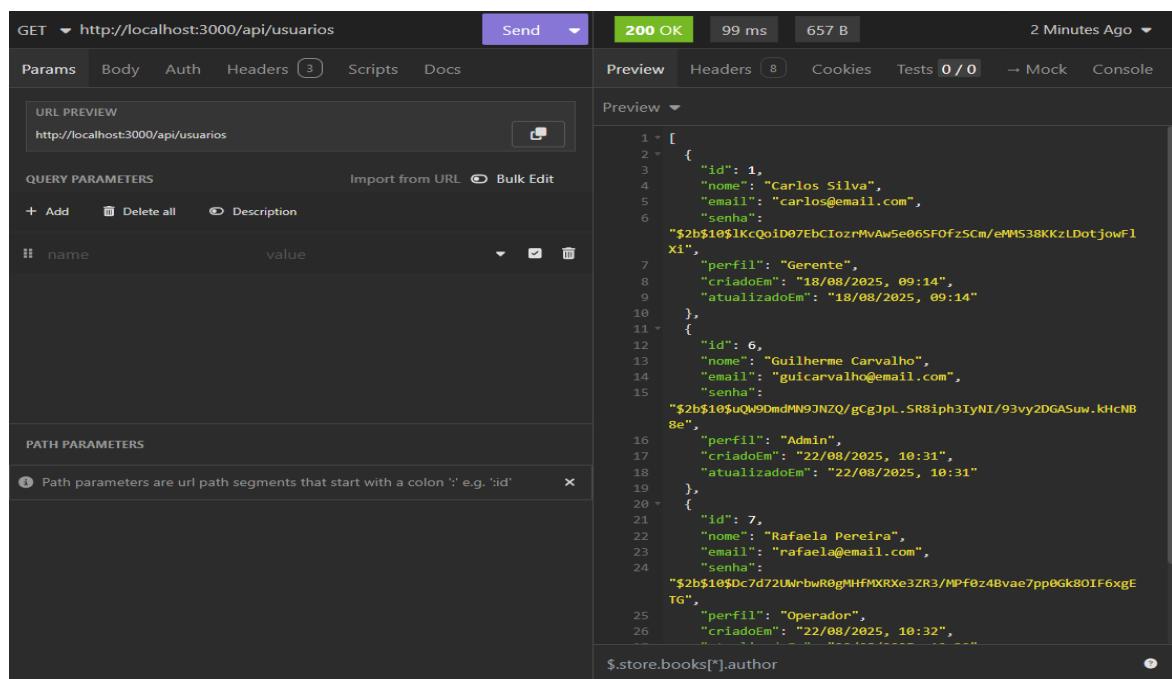
```
1 v {  
2   "nome": "Gabriela Demori",  
3   "email": "gabidemori@teste.com",  
4   "senha": "NexoERP123",  
5   "perfil": "Admin"  
6 }
```

Response Body (JSON):

```
1 v {  
2   "id": 18,  
3   "nome": "Gabriela Demori",  
4   "email": "gabidemorig@teste.com",  
5   "senha":  
6     "$2b$10$pkQnluLHeK7RX/tV/3.VsefkF/JjmzqLvPdm90xHxe7F3HvfRY6tu",  
7   "perfil": "Admin",  
8   "criadoEm": "22/08/2025, 14:16",  
9   "atualizadoEm": "22/08/2025, 14:16"  
9 }
```

GET – Buscar usuários

- 🔍 Testes de busca de todos os usuários e por ID individual. Verificação de retorno correto e status 200 OK.



The screenshot shows a Postman request for a GET operation at `http://localhost:3000/api/usuarios`. The response status is `200 OK` with a response time of 99 ms and a body size of 657 B, timestamped 2 Minutes Ago. The response body is a JSON array containing multiple user objects, each with id, nome, email, senha, perfil, criadoEm, and atualizadoEm fields.

Params	Body	Auth	Headers	Scripts	Docs	Preview	Headers 8	Cookies	Tests 0 / 0	→ Mock	Console
						Preview	Headers 8	Cookies	Tests 0 / 0	→ Mock	Console

URL PREVIEW: `http://localhost:3000/api/usuarios`

QUERY PARAMETERS:

- + Add
- Delete all
- Description

name	value
------	-------

PATH PARAMETERS:

Path parameters are url path segments that start with a colon ':'. e.g. ':id' ×

Response Body (JSON):

```
1 v [  
2   {  
3     "id": 1,  
4     "nome": "Carlos Silva",  
5     "email": "carlos@email.com",  
6     "senha":  
7       "$2b$10$1KcQoId07EbCIozrMvAw5e065FOfzScm/eMMS38KKzLDot:jowFlX1",  
8     "perfil": "Gerente",  
9     "criadoEm": "18/08/2025, 09:14",  
10    "atualizadoEm": "18/08/2025, 09:14"  
10  },  
11  {  
12    "id": 6,  
13    "nome": "Guilherme Carvalho",  
14    "email": "guicarvalho@email.com",  
15    "senha":  
16      "$2b$10$uOW9mdMM9JNZQ/gCgJpL.SR8iph3IyNI/93vy2DGASuw.kHcNB8e",  
17    "perfil": "Admin",  
18    "criadoEm": "22/08/2025, 10:31",  
19    "atualizadoEm": "22/08/2025, 10:31"  
19  },  
20  {  
21    "id": 7,  
22    "nome": "Rafaela Pereira",  
23    "email": "rafaela@email.com",  
24    "senha":  
25      "$2b$10$Dc7d2UrbwR0gMHfMXRXe3ZR3/MPF0z4Bvae7pp9GkBOIF6xgETG",  
26    "perfil": "Operador",  
26    "criadoEm": "22/08/2025, 10:32",  
26  }]  
$store.books[""].author
```

GET ▾ <http://localhost:3000/api/usuarios/1>

Send ▾ **200 OK** 37 ms 213 B Just Now ▾

Params	Body	Auth	Headers (3)	Scripts	Docs	Preview	Headers (8)	Cookies	Tests 0 / 0	→ Mock	Console	
+ Add	Delete all	Description	Preview ▾									
Accept	*		<pre>1 v { 2 "id": 1, 3 "nome": "Carlos Silva", 4 "email": "carlos@email.com", 5 "senha": 6 '\$2b\$10\$lKcQoiD07EbCiozrMvAwSe06SFOfzSCm/eMMS38KKzLDotjowFlX 7 "perfil": "Gerente", 8 "criadoEm": "18/08/2025, 09:14", 9 "atualizadoEm": "18/08/2025, 09:14" }</pre>									
Host	<calculated at runtime>											
User-Agent	insomnia/11.4.0											

PUT – Atualizar usuário

🔗 Requisição PUT testando atualização dos dados de um usuário existente. Alterações persistidas com sucesso.

PUT ▾ <http://localhost:3000/api/usuarios/17>

Send ▾ **200 OK** 44 ms 169 B Just Now ▾

Params	Body (1)	Auth	Headers (4)	Scripts	Docs	Preview	Headers (8)	Cookies	Tests 0 / 0	→ Mock	Console
JSON ▾											
<pre>1 v { 2 "nome": "Gabriela Demori", 3 "email": "gabidemori@gmail.com", 4 "senha": "NexoERP123", 5 "perfil": "Admin" 6 }</pre>											
<pre>1 v { 2 "id": 17, 3 "nome": "Gabriela Demori", 4 "email": "gabidemori@gmail.com", 5 "senha": "NexoERP123", 6 "perfil": "Admin", 7 "criadoEm": "22/08/2025, 13:32", 8 "atualizadoEm": "22/08/2025, 14:25" 9 }</pre>											

DELETE – Remover usuário

🌐 Exclusão de usuário via DELETE. Teste validado com sucesso, retornando status 200 OK.

DELETE ▾ <http://localhost:3000/api/usuarios/17>

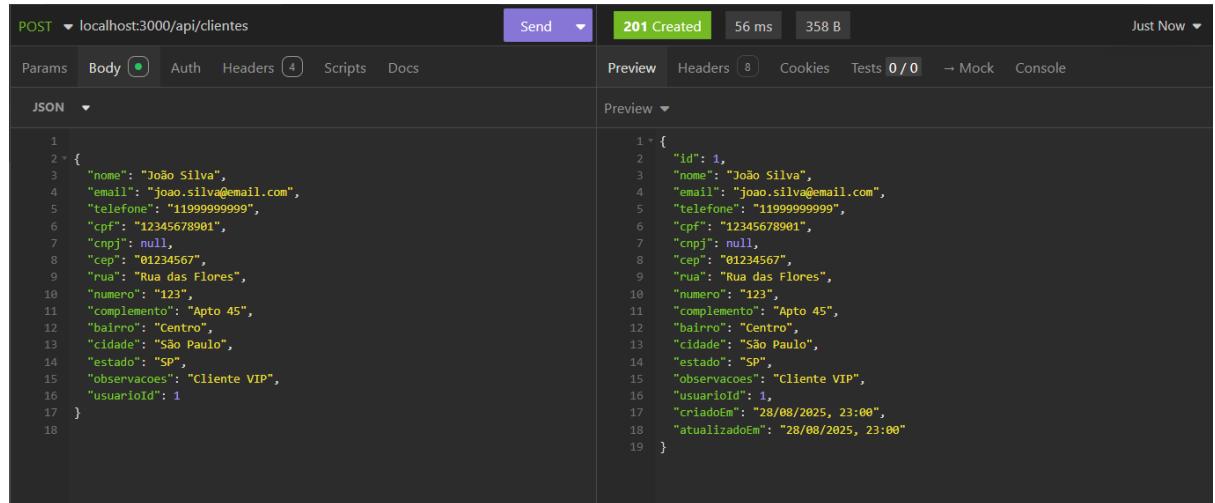
Send ▾ **200 OK** 70 ms 45 B Just Now ▾

Params	Body	Auth	Headers (3)	Scripts	Docs	Preview	Headers (8)	Cookies	Tests 0 / 0	→ Mock	Console
No Body ▾											
<pre>1 v { 2 "mensagem": "Usuário deletado com sucesso." 3 }</pre>											

Clientes

POST – Criar novo cliente

- ✓ Registro de cliente criado com sucesso. Resposta inclui dados completos e código de status 201 Created.



POST localhost:3000/api/clientes Send 201 Created 56 ms 358 B Just Now

Params Body Auth Headers 4 Scripts Docs

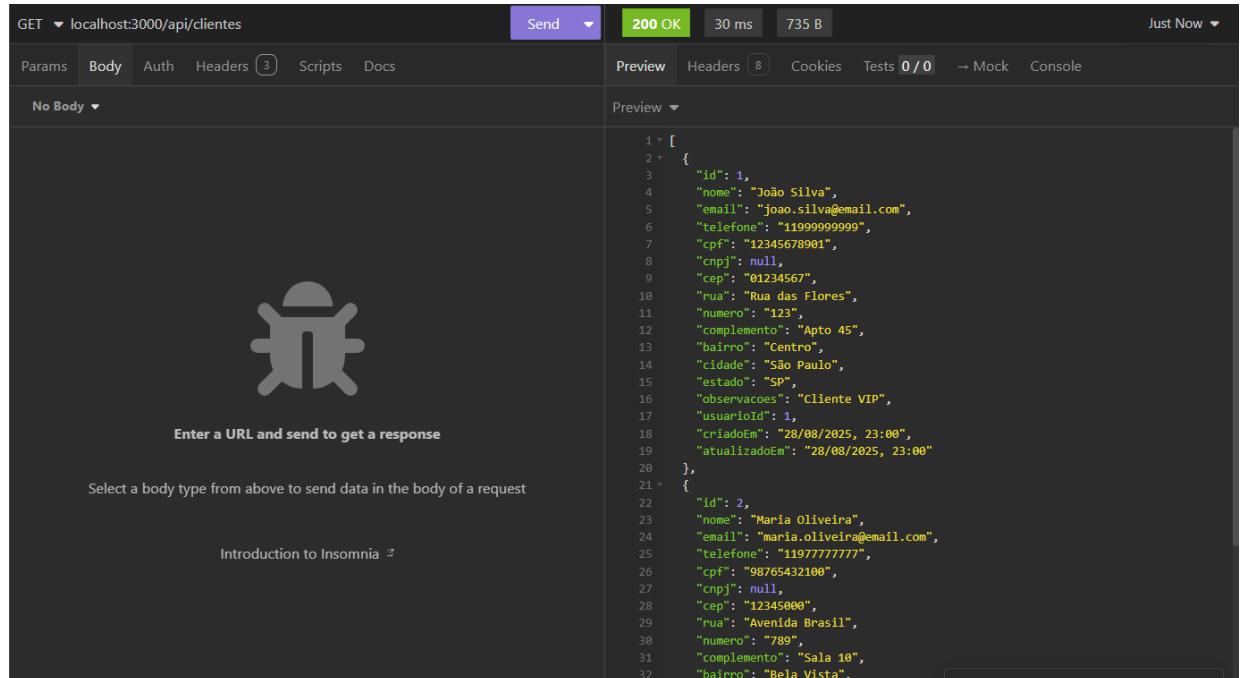
JSON Preview

```
1
2 {
3   "nome": "João Silva",
4   "email": "joao.silva@email.com",
5   "telefone": "11999999999",
6   "cpf": "12345678901",
7   "cnpj": null,
8   "cep": "01234567",
9   "rua": "Rua das Flores",
10  "numero": "123",
11  "complemento": "Apto 45",
12  "bairro": "Centro",
13  "cidade": "São Paulo",
14  "estado": "SP",
15  "observacoes": "Cliente VIP",
16  "usuarioId": 1
17 }
18 }
```

1 {
2 "id": 1,
3 "nome": "João Silva",
4 "email": "joao.silva@email.com",
5 "telefone": "11999999999",
6 "cpf": "12345678901",
7 "cnpj": null,
8 "cep": "01234567",
9 "rua": "Rua das Flores",
10 "numero": "123",
11 "complemento": "Apto 45",
12 "bairro": "Centro",
13 "cidade": "São Paulo",
14 "estado": "SP",
15 "observacoes": "Cliente VIP",
16 "usuarioId": 1,
17 "criadoEm": "28/08/2025, 23:00",
18 "atualizadoEm": "28/08/2025, 23:00"
19 }

GET – Buscar clientes

- 🔍 Retorno de lista de clientes e consulta individual por ID. Todos os testes retornaram conforme esperado.



GET localhost:3000/api/clientes Send 200 OK 30 ms 735 B Just Now

Params Body Auth Headers 3 Scripts Docs

No Body Preview

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia

```
1 [
2   {
3     "id": 1,
4     "nome": "João Silva",
5     "email": "joao.silva@email.com",
6     "telefone": "11999999999",
7     "cpf": "12345678901",
8     "cnpj": null,
9     "cep": "01234567",
10    "rua": "Rua das Flores",
11    "numero": "123",
12    "complemento": "Apto 45",
13    "bairro": "Centro",
14    "cidade": "São Paulo",
15    "estado": "SP",
16    "observacoes": "Cliente VIP",
17    "usuarioId": 1,
18    "criadoEm": "28/08/2025, 23:00",
19    "atualizadoEm": "28/08/2025, 23:00"
20  },
21  {
22    "id": 2,
23    "nome": "Maria Oliveira",
24    "email": "maria.oliveira@email.com",
25    "telefone": "11977777777",
26    "cpf": "98765432100",
27    "cnpj": null,
28    "cep": "12345000",
29    "rua": "Avenida Brasil",
30    "numero": "789",
31    "complemento": "Sala 10",
32    "bairro": "Bela Vista",
33  }
34 ]
```

GET ▾ localhost:3000/api/clientes/1

Send ▾ 200 OK 127 ms 358 B Just Now ▾

Params	Body	Auth	Headers (3)	Scripts	Docs	Preview	Headers (8)	Cookies	Tests 0 / 0	→ Mock	Console	
No Body ▾						Preview ▾						
						1 + { 2 "id": 1, 3 "nome": "João silva", 4 "email": "joao.silva@email.com", 5 "telefone": "11999999999", 6 "cpf": "12345678901", 7 "cnpj": null, 8 "cep": "01234567", 9 "rua": "Rua das Flores", 10 "numero": "123", 11 "complemento": "Apto 45", 12 "bairro": "Centro", 13 "cidade": "São Paulo", 14 "estado": "SP", 15 "observacoes": "Cliente VIP", 16 "usuarioId": 1, 17 "criadoEm": "28/08/2025, 23:00", 18 "atualizadoEm": "28/08/2025, 23:00" 19 }						
						Enter a URL and send to get a response						
						Select a body type from above to send data in the body of a request						

PUT – Atualizar cliente

Atualização de dados do cliente realizada com sucesso. Resposta confirma persistência das alterações.

PUT ▾ localhost:3000/api/clientes/2

Send ▾ 200 OK 45 ms 374 B Just Now ▾

Params	Body (1)	Auth	Headers (4)	Scripts	Docs	Preview	Headers (8)	Cookies	Tests 0 / 0	→ Mock	Console	
No Body ▾						Preview ▾						
						1 + { 2 "id": 2, 3 "nome": "Maria Oliveira", 4 "email": "maria.oliveira@email.com", 5 "telefone": "11977777777", 6 "cpf": "98765432330", 7 "cnpj": null, 8 "cep": "12345000", 9 "rua": "Avenida Brasil", 10 "numero": "789", 11 "complemento": "Sala 10", 12 "bairro": "Bela Vista", 13 "cidade": "Rio de Janeiro", 14 "estado": "RJ", 15 "observacoes": "Cliente nova", 16 "usuarioId": 1 17 }						
						1 + { 2 "id": 2, 3 "nome": "Maria Oliveira", 4 "email": "maria.oliveira@email.com", 5 "telefone": "11977777777", 6 "cpf": "98765432330", 7 "cnpj": null, 8 "cep": "12345000", 9 "rua": "Avenida Brasil", 10 "numero": "789", 11 "complemento": "Sala 10", 12 "bairro": "Bela Vista", 13 "cidade": "Rio de Janeiro", 14 "estado": "RJ", 15 "observacoes": "Cliente nova", 16 "usuarioId": 1, 17 "criadoEm": "28/08/2025, 23:02", 18 "atualizadoEm": "28/08/2025, 23:05" 19 }						
						Enter a URL and send to get a response						
						Select a body type from above to send data in the body of a request						

DELETE – Remover cliente

Cliente removido da base. Teste de exclusão validado sem erros.

DELETE ▾ localhost:3000/api/clientes/1

Send ▾ 200 OK 67 ms 44 B Just Now ▾

Params	Body	Auth	Headers (3)	Scripts	Docs	Preview	Headers (8)	Cookies	Tests 0 / 0	→ Mock	Console
No Body ▾						Preview ▾					
						1 + { 2 "mensagem": "Cliente deletado com sucesso." 3 }					
						Enter a URL and send to get a response					
						Select a body type from above to send data in the body of a request					

Produtos

POST – Criar novo produto

✓ Produto e Serviço adicionado ao catálogo com sucesso. Teste retornou status 201 Created.

POST ▾ http://localhost:3000/api/produtos	Send ▾	201 Created	54 ms	221 B	Just Now ▾
Params Body <input checked="" type="radio"/> Auth Headers (4) Scripts Docs	Preview Headers (8) Cookies Tests 0 / 0 → Mock Console				
JSON ▾	Preview ▾				
<pre>1 v { 2 "nome": "Consultoria de TI", 3 "descricao": "Serviço de consultoria técnica 4 especializada", 5 "preco": 350.00, 6 "estoque": 0, 7 "tipo": "Serviço", 8 "status": "Ativo" 9 }</pre>	<pre>1 v { 2 "id": 18, 3 "nome": "Consultoria de TI", 4 "descricao": "Serviço de consultoria técnica 5 especializada", 6 "preco": 350, 7 "estoque": 0, 8 "tipo": "Serviço", 9 "status": "Ativo", 10 "criadoEm": "22/08/2025, 16:47", 11 "atualizadoEm": "22/08/2025, 16:47" 12 }</pre>				
POST ▾ http://localhost:3000/api/produtos	Send ▾	201 Created	54 ms	221 B	Just Now ▾
Params Body <input checked="" type="radio"/> Auth Headers (4) Scripts Docs	Preview Headers (8) Cookies Tests 0 / 0 → Mock Console				
JSON ▾	Preview ▾				
<pre>1 v { 2 "nome": "Consultoria de TI", 3 "descricao": "Serviço de consultoria técnica 4 especializada", 5 "preco": 350.00, 6 "estoque": 0, 7 "tipo": "Serviço", 8 "status": "Ativo" 9 }</pre>	<pre>1 v { 2 "id": 18, 3 "nome": "Consultoria de TI", 4 "descricao": "Serviço de consultoria técnica 5 especializada", 6 "preco": 350, 7 "estoque": 0, 8 "tipo": "Serviço", 9 "status": "Ativo", 10 "criadoEm": "22/08/2025, 16:47", 11 "atualizadoEm": "22/08/2025, 16:47" 12 }</pre>				

GET – Buscar produtos

🔍 Testes de consulta geral e individual de produtos e serviços retornaram com sucesso os dados armazenados.

GET ▾ http://localhost:3000/api/produtos	Send ▾	200 OK	26 ms	2.2 KB	Just Now ▾
Params Body Auth Headers (3) Scripts Docs	Preview Headers (8) Cookies Tests 0 / 0 → Mock Console				
No Body ▾	Preview ▾				
	<pre>96 "tipo": "Produto", 97 "status": "Ativo", 98 "criadoEm": "22/08/2025, 16:42", 99 "atualizadoEm": "22/08/2025, 16:42" 100 }, 101 [102 { 103 "id": 17, 104 "nome": "Notebook Elite", 105 "descricao": "Notebook i7 16GB RAM 512GB SSD", 106 "preco": 4599.9, 107 "estoque": 20, 108 "tipo": "Produto", 109 "status": "Ativo", 110 "criadoEm": "22/08/2025, 16:46", 111 "atualizadoEm": "22/08/2025, 16:46" 112 }, 113 { 114 "id": 18, 115 "nome": "Consultoria de TI", 116 "descricao": "Serviço de consultoria técnica 117 especializada", 118 "preco": 350, 119 "estoque": 0, 120 "tipo": "Serviço", 121 "status": "Ativo", 122 "criadoEm": "22/08/2025, 16:47", 123 "atualizadoEm": "22/08/2025, 16:47" 124 } 125]</pre>				
Enter a URL and send to get a response					
Select a body type from above to send data in the body of a request					
Introduction to Insomnia ↗					

PUT – Atualizar produto

💡 Informações de produto atualizadas corretamente via PUT. Testes confirmam persistência dos dados.

PUT ▾ http://localhost:3000/api/produtos/18	Send ▾	200 OK	55 ms	221 B	Just Now ▾
Params	Body (1)	Auth	Headers (4)	Scripts	Docs
JSON ▾	Preview ▾				
<pre>1 v { 2 "nome": "Consultoria de TI", 3 "descricao": "Serviço de consultoria técnica especializada", 4 "preco": 380.00, 5 "estoque": 0, 6 "tipo": "Serviço", 7 "status": "Ativo" 8 }</pre>	<pre>1 v { 2 "id": 18, 3 "nome": "Consultoria de TI", 4 "descricao": "Serviço de consultoria técnica especializada", 5 "preco": 380, 6 "estoque": 0, 7 "tipo": "Serviço", 8 "status": "Ativo", 9 "criadoEm": "22/08/2025, 16:47", 10 "atualizadoEm": "22/08/2025, 16:48" 11 }</pre>				

DELETE – Remover produto

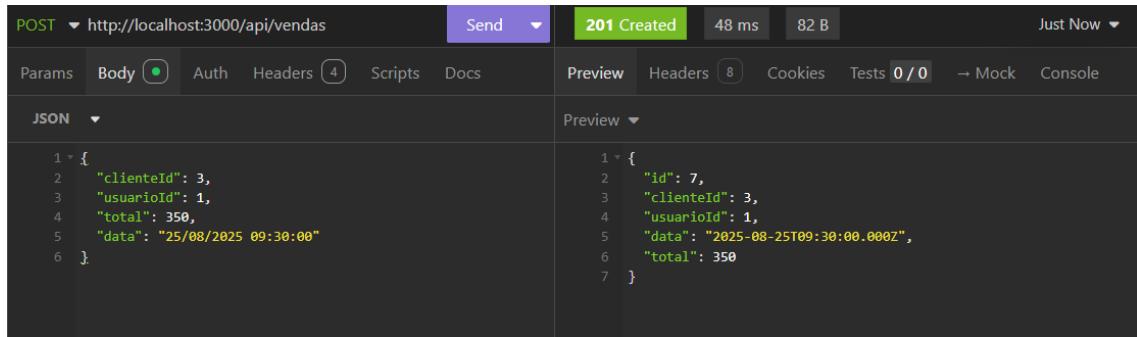
🗑 Produto removido da base. Operação executada corretamente e sem erros.

DELETE ▾ http://localhost:3000/api/produtos/18	Send ▾	200 OK	27 ms	42 B	Just Now ▾
Params	Body	Auth	Headers (3)	Scripts	Docs
No Body ▾	Preview ▾				
	<pre>1 v { 2 "message": "Produto deletado com sucesso" 3 }</pre>				
<p>Enter a URL and send to get a response</p> <p>Select a body type from above to send data in the body of a request</p> <p>Introduction to Insomnia ↗</p> <p>\$store.books[*].author</p>					

Vendas

POST – Criar nova venda

- Nova venda registrada via POST. Dados da venda e ID retornados corretamente.



POST <http://localhost:3000/api/vendas> Send 201 Created 48 ms 82 B Just Now

Params Body Auth Headers 4 Scripts Docs Preview Headers 8 Cookies Tests 0 / 0 → Mock Console

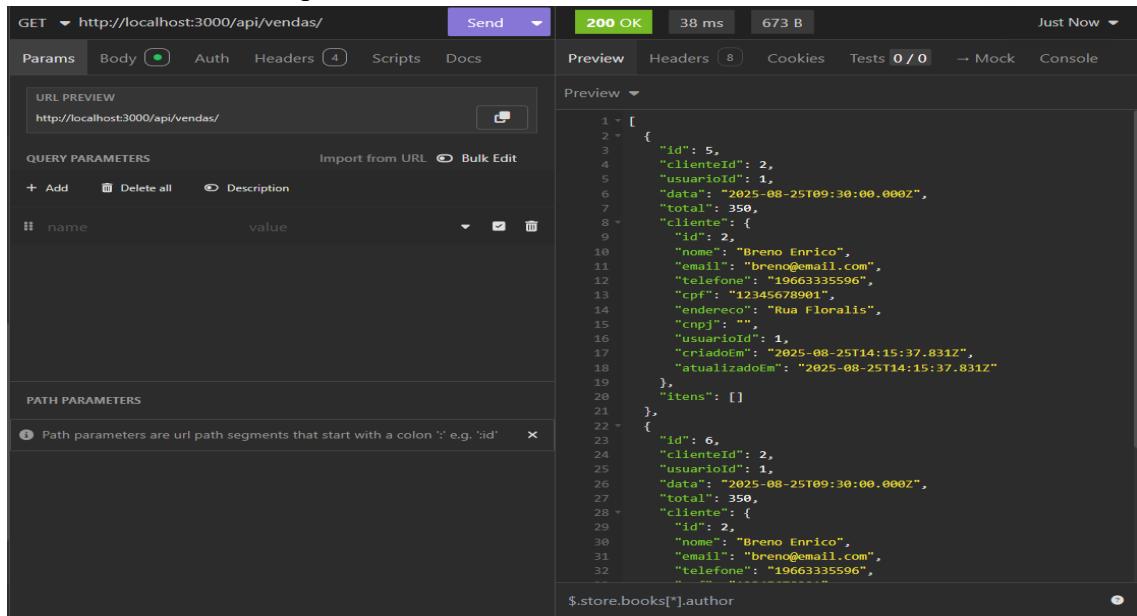
JSON Preview

```
1: {  
2:   "clienteId": 3,  
3:   "usuarioId": 1,  
4:   "total": 350,  
5:   "data": "2025-08/2025 09:30:00"  
6: }
```

```
1: {  
2:   "id": 7,  
3:   "clienteId": 3,  
4:   "usuarioId": 1,  
5:   "data": "2025-08-25T09:30:00.000Z",  
6:   "total": 350  
7: }
```

GET – Buscar vendas

- Consulta de todas as vendas realizadas e busca por venda específica. Testes retornaram os dados esperados.



GET <http://localhost:3000/api/vendas/> Send 200 OK 38 ms 673 B Just Now

Params Body Auth Headers 4 Scripts Docs Preview Headers 8 Cookies Tests 0 / 0 → Mock Console

URL PREVIEW <http://localhost:3000/api/vendas/>

QUERY PARAMETERS Import from URL Bulk Edit

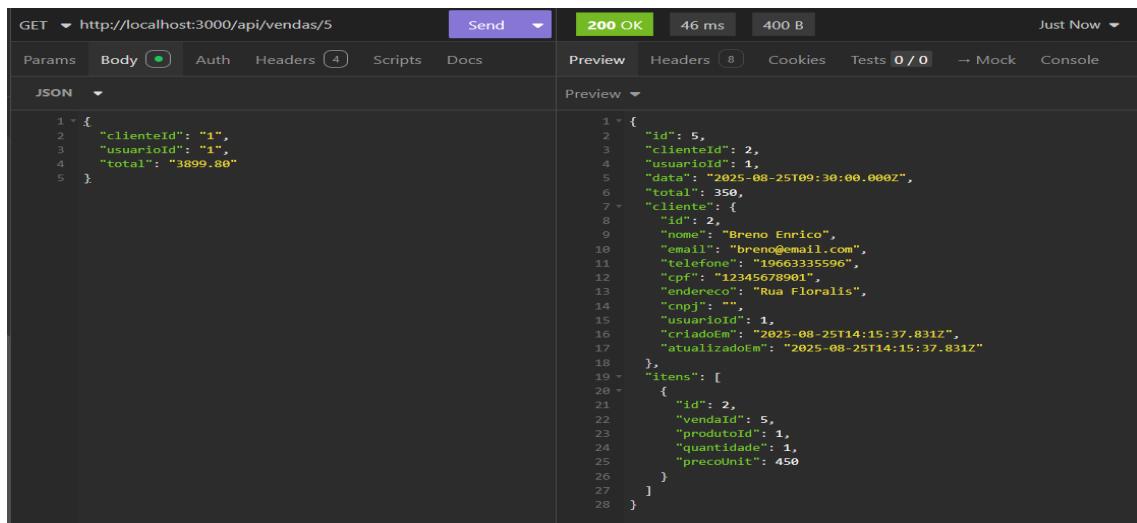
+ Add Delete all Description

name	value

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':'. e.g. ':id' X

```
1: [  
2:   {  
3:     "id": 5,  
4:     "clienteId": 2,  
5:     "usuarioId": 1,  
6:     "data": "2025-08-25T09:30:00.000Z",  
7:     "total": 350,  
8:     "cliente": {  
9:       "id": 2,  
10:      "nome": "Breno Enrico",  
11:      "email": "breno@email.com",  
12:      "telefone": "19663335596",  
13:      "cpf": "12345678901",  
14:      "endereco": "Rua Floralis",  
15:      "cnpj": "",  
16:      "usuarioId": 1,  
17:      "criadoEm": "2025-08-25T14:15:37.831Z",  
18:      "atualizadoEm": "2025-08-25T14:15:37.831Z"  
19:    },  
20:    "itens": []  
21:  },  
22:  {  
23:    "id": 6,  
24:    "clienteId": 2,  
25:    "usuarioId": 1,  
26:    "data": "2025-08-25T09:30:00.000Z",  
27:    "total": 350,  
28:    "cliente": {  
29:      "id": 2,  
30:      "nome": "Breno Enrico",  
31:      "email": "breno@email.com",  
32:      "telefone": "19663335596",  
33:      "cnpj": "",  
34:      "endereco": "Rua Floralis",  
35:      "usuarioId": 1,  
36:      "criadoEm": "2025-08-25T14:15:37.831Z",  
37:      "atualizadoEm": "2025-08-25T14:15:37.831Z"  
38:    },  
39:    "itens": []  
40:  }]
```



GET <http://localhost:3000/api/vendas/5> Send 200 OK 46 ms 400 B Just Now

Params Body Auth Headers 4 Scripts Docs Preview Headers 8 Cookies Tests 0 / 0 → Mock Console

JSON Preview

```
1: {  
2:   "clienteId": "1",  
3:   "usuarioId": "1",  
4:   "total": "3899.80"  
5: }
```

```
1: {  
2:   "id": 5,  
3:   "clienteId": 2,  
4:   "usuarioId": 1,  
5:   "data": "2025-08-25T09:30:00.000Z",  
6:   "total": 350,  
7:   "cliente": {  
8:     "id": 2,  
9:     "nome": "Breno Enrico",  
10:    "email": "breno@email.com",  
11:    "telefone": "19663335596",  
12:    "cpf": "12345678901",  
13:    "endereco": "Rua Floralis",  
14:    "cnpj": "",  
15:    "usuarioId": 1,  
16:    "criadoEm": "2025-08-25T14:15:37.831Z",  
17:    "atualizadoEm": "2025-08-25T14:15:37.831Z"  
18:  },  
19:   "itens": [  
20:     {  
21:       "id": 2,  
22:       "vendaId": 5,  
23:       "produtoId": 1,  
24:       "quantidade": 1,  
25:       "precoUnit": 450  
26:     }  
27:   ]  
28: }
```

PUT – Atualizar venda

⌚ Alteração nos dados da venda aplicada com sucesso. Requisição PUT validada.

The screenshot shows a POSTMAN interface with the following details:

- Method:** PUT
- URL:** http://localhost:3000/api/vendas/5
- Status:** 200 OK
- Time:** 61 ms
- Size:** 82 B
- Headers:** Just Now
- Params:**
- Body (JSON):**

```
1 ✘ {  
2   "clienteId": 2,  
3   "usuarioId": 1,  
4   "total": 450,  
5   "data": "25/08/2025 09:30:00"  
6 }
```
- Preview:** Shows the JSON response with the same fields and values as the request body.

DELETE – Remover venda

🗑 Venda excluída com sucesso. Teste finalizado com resposta 201.

The screenshot shows a POSTMAN interface with the following details:

- Method:** DELETE
- URL:** http://localhost:3000/api/vendas/5
- Status:** 200 OK
- Time:** 42 ms
- Size:** 40 B
- Headers:** Just Now
- Params:**
- Body (No Body):**
- Preview:** Shows a JSON response with a single key "message" containing the value "Venda deletada com sucesso".

ItensVenda

POST – Adicionar item à venda

- ✓ Novo item vinculado a uma venda com sucesso. Teste validou retorno e vinculação correta.

POST <http://localhost:3000/api/itensvenda> Send 201 Created 202 ms 65 B Just Now

Params	Body	Auth	Headers	Scripts	Docs	Preview	Headers	Cookies	Tests	→ Mock	Console
						Preview	Headers 8		0 / 0		

JSON ▾

```
1 ↴ {  
2   "vendaId": 6,  
3   "productId": 1,  
4   "quantidade": 1,  
5   "precoUnit": 450  
6 }
```

1 ↴ {
2 "id": 4,
3 "vendaId": 6,
4 "productId": 1,
5 "quantidade": 1,
6 "precoUnit": 450
7 }

GET – Buscar itens de venda

- 🔍 Consulta de itens vinculados às vendas. Dados retornados com precisão.

GET <http://localhost:3000/api/itensvenda> Send 200 OK 27 ms 67 B Just Now

Params	Body	Auth	Headers	Scripts	Docs	Preview	Headers	Cookies	Tests	→ Mock	Console
						Preview	Headers 8		0 / 0		

No Body ▾

1 ↴ [
2 {
3 "id": 4,
4 "vendaId": 6,
5 "productId": 1,
6 "quantidade": 1,
7 "precoUnit": 450
8 }
9]

GET <http://localhost:3000/api/itensvenda/4> Send 200 OK 52 ms 65 B Just Now

Params	Body	Auth	Headers	Scripts	Docs	Preview	Headers	Cookies	Tests	→ Mock	Console
						Preview	Headers 8		0 / 0		

No Body ▾

1 ↴ {
2 "id": 4,
3 "vendaId": 6,
4 "productId": 1,
5 "quantidade": 1,
6 "precoUnit": 450
7 }

PUT – Atualizar item de venda

- 🔄 Alteração em item de venda realizada com sucesso. Resposta confirma atualização.

PUT <http://localhost:3000/api/itensvenda/4> Send 200 OK 46 ms 65 B Just Now

Params	Body	Auth	Headers	Scripts	Docs	Preview	Headers	Cookies	Tests	→ Mock	Console
						Preview	Headers 8		0 / 0		

JSON ▾

```
1 ↴ {  
2   "vendaId": 6,  
3   "productId": 1,  
4   "quantidade": 1,  
5   "precoUnit": 450  
6 }
```

1 ↴ {
2 "id": 4,
3 "vendaId": 6,
4 "productId": 1,
5 "quantidade": 1,
6 "precoUnit": 450
7 }

DELETE – Remover item de venda

🗑 Item desvinculado/removido da venda. Teste executado com sucesso.

DELETE ▾ http://localhost:3000/api/itensvenda/4	Send ▾	200 OK	26 ms	50 B	Just Now ▾
Params Body Auth Headers (3) Scripts Docs	Preview Headers (8) Cookies Tests 0 / 0 → Mock Console				
No Body ▾	Preview ▾	1 ~ { 2 "mensagem": "Item de venda deletado com sucesso." 3 }			

RelatoriosVenda

POST – Adicionar relatório de venda

✓ Novo relatório vinculado com sucesso. Teste validou retorno e associação correta.

POST ▾ http://localhost:3000/api/relatoriosvenda	Send ▾	201 Created	34 ms	211 B	Just Now ▾
Params Body (1) Auth Headers (4) Scripts Docs	Preview Headers (8) Cookies Tests 0 / 0 → Mock Console				
JSON ▾	Preview ▾	1 ~ { 2 "id": 5, 3 "vendaId": 4, 4 "dataRelatorio": "2025-08-25T14:31:39.937Z", 5 "observacoes": "Relatório gerado automaticamente após a finalização da venda.", 6 "criadoEm": "2025-08-25T14:31:39.937Z" 7 } 1 ~ { 2 "id": 5, 3 "vendaId": 4, 4 "dataRelatorio": "2025-08-25T14:31:39.937Z", 5 "observacoes": "Relatório gerado automaticamente após a finalização da venda.", 6 "criadoEm": "25/08/2025, 11:31", 7 "atualizadoEm": "25/08/2025, 15:31" 8 }			

GET – Buscar relatórios de venda

🔍 Consulta de relatórios de venda realizada. Dados retornados com precisão.

GET ▾ http://localhost:3000/api/relatoriosvenda/	Send ▾	200 OK	25 ms	213 B	1 Minute Ago ▾
Params Body Auth Headers (3) Scripts Docs	Preview Headers (8) Cookies Tests 0 / 0 → Mock Console				
No Body ▾	Preview ▾	1 ~ [2 ~ { 3 "id": 5, 4 "vendaId": 4, 5 "dataRelatorio": "2025-08-25T14:31:39.937Z", 6 "observacoes": "Relatório gerado automaticamente após a finalização da venda.", 7 "criadoEm": "25/08/2025, 11:31", 8 "atualizadoEm": "25/08/2025, 15:31" 9 } 10]			

GET ▾ http://localhost:3000/api/relatoriosvenda/5 Send ▾ 200 OK 47 ms 211 B Just Now ▾

Params	Body	Auth	Headers 3	Scripts	Docs
No Body ▾			Preview ▾ <pre>1 > { 2 "id": 5, 3 "vendaId": 4, 4 "dataRelatorio": "2025-08-25T14:31:39.937Z", 5 "observacoes": "Relatório gerado automaticamente após a finalização da venda.", 6 "criadoEm": "25/08/2025, 11:31", 7 "atualizadoEm": "25/08/2025, 15:31" 8 }</pre>		

PUT – Atualizar relatório de venda

🕒 Alteração em relatório de venda realizada com sucesso. Resposta confirma atualização.

GET ▾ http://localhost:3000/api/relatoriosvenda/5 Send ▾ 200 OK 25 ms 211 B Just Now ▾

Params	Body	Auth	Headers 4	Scripts	Docs
JSON ▾			Preview ▾ <pre>1 > { 2 "vendaId": 4, 3 "dataRelatorio": "2025-08-26T14:31:39.937Z", 4 "observacoes": "Relatório gerado automaticamente após a finalização da venda.", 5 "criadoEm": "2025-08-25T14:31:39.937Z" 6 } 7</pre> <pre>1 > { 2 "id": 5, 3 "vendaId": 4, 4 "dataRelatorio": "2025-08-25T14:31:39.937Z", 5 "observacoes": "Relatório gerado automaticamente após a finalização da venda.", 6 "criadoEm": "25/08/2025, 11:31", 7 "atualizadoEm": "25/08/2025, 15:31" 8 }</pre>		

DELETE – Remover relatório de venda

🗂 Relatório de venda removido com sucesso. Teste executado e validado corretamente.

DELETE ▾ http://localhost:3000/api/relatoriosvenda/5 Send ▾ 200 OK 54 ms 56 B Just Now ▾

Params	Body	Auth	Headers 3	Scripts	Docs
No Body ▾			Preview ▾ <pre>1 > { 2 "mensagem": "Relatório de venda deletado com sucesso." 3 }</pre>		