

**Módulo 2**

**Linguagem de Programação II**

**Turma TI23**

**Atividade 04**

# **Trabalho Final Documentação**

Link do repositório GitHub "trabalho.grupo.petshop":  
<https://github.com/arturbazza/trabalho.grupo.petshop.git>

Integrantes:

**Artur Bazzanella Christmann**

**Diogo Sehn Brum**

**Gustavo Waschburger**

**Tharsila Moraes Simoes**

# Documentação para teste do trabalho utilizando Postman

URL do servidor local: <http://localhost:8080/>

## Cliente

**Criar Cliente:** Para criar um novo cliente, utilize a API de clientes, enviando uma requisição **POST** para o endpoint **/clientes/salvar**. No corpo da requisição, forneça os dados necessários, como nome, CPF e endereços associados ao cliente.

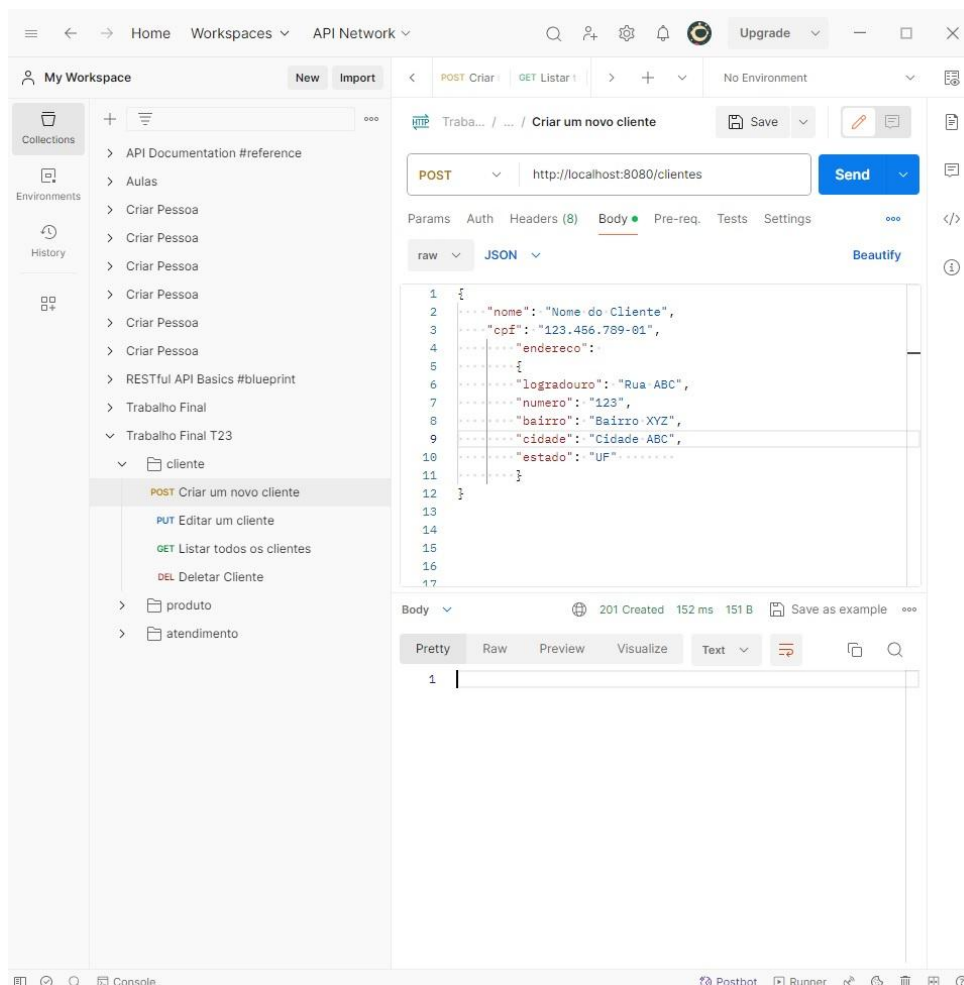
- Método: **POST**
- Endpoint: **/clientes/salvar**
- Corpo da requisição (JSON):

```
{  
  "nome": "Nome do Cliente",  
  "cpf": "123.456.789-01",  
  "endereco": {  
    "logradouro": "Rua ABC",  
    "numero": "123",  
    "bairro": "Bairro XYZ",  
    "cidade": "Cidade ABC",  
    "estado": "UF"  
  }  
}
```

**Modelo (print screen):**

“201 Created

The request has been fulfilled and resulted in a new resource being created.”



**Editar Cliente:** Para editar os detalhes de um cliente existente, faça uma requisição **PUT** para **/id**, onde **{id}** é o identificador único do cliente a ser editado. No corpo da requisição, envie os dados atualizados.

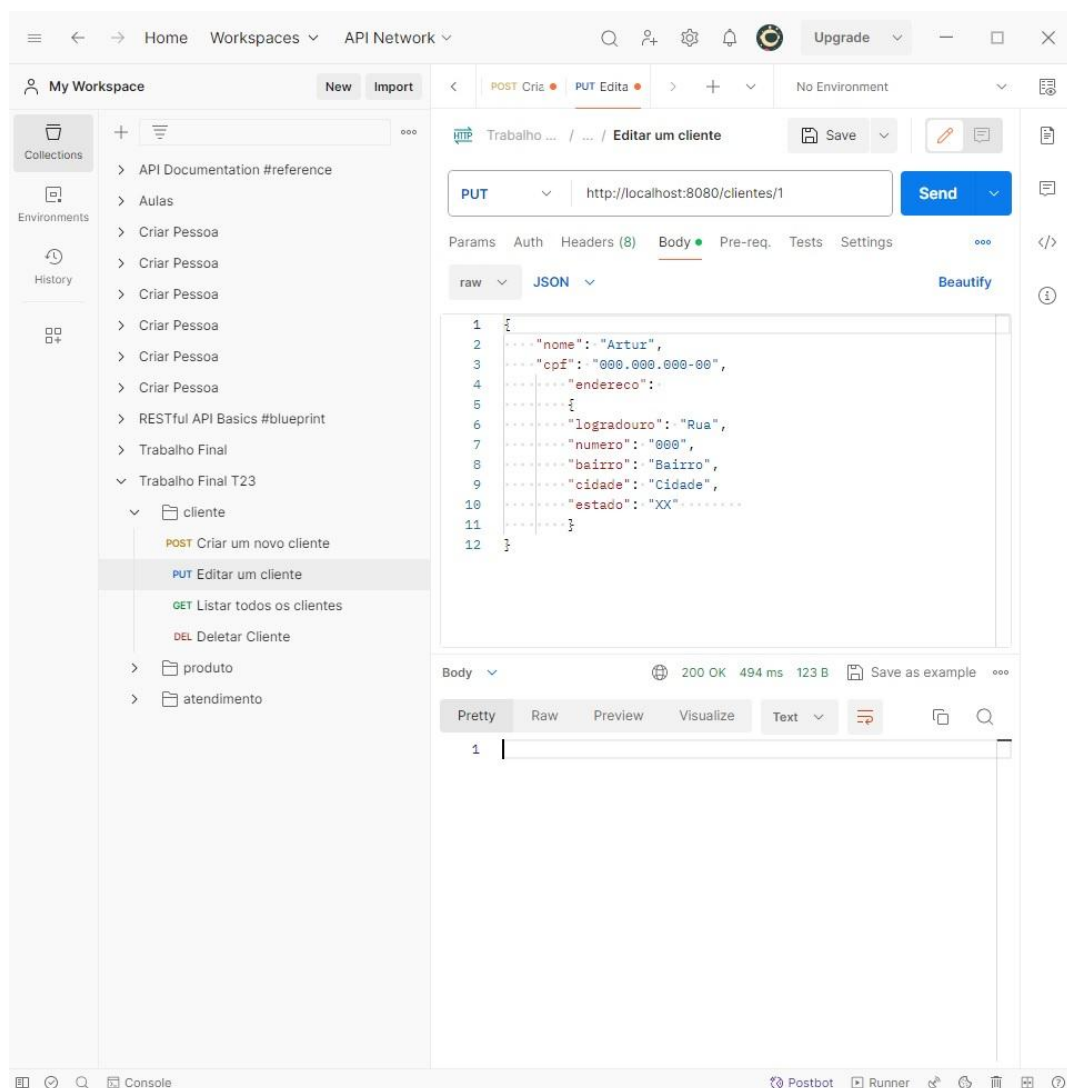
- Método: **PUT**
- Endpoint: **/id**
- Corpo da requisição (JSON):

```
{
  "nome": "Artur",
  "cpf": "000.000.000-00",
  "endereco": {
    "logradouro": "Rua",
    "numero": "000",
    "bairro": "Bairro",
    "cidade": "Cidade",
    "estado": "XX"
  }
}
```

**Modelo (print screen):**

"200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action. "



**Listar Clientes:** Para obter uma lista de todos os clientes, faça uma requisição **GET** para o endpoint **/clientes?nome**. Isso retornará uma lista de objetos Cliente.

- Método: **GET**
- Endpoint: **/clientes?nome**

**Modelo (print screen):**

"200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action. "

The screenshot shows the Postman API client interface. On the left, the 'My Workspace' sidebar lists various collections and environments. The main panel displays a GET request to the endpoint `http://localhost:8080/clientes?nome`. The 'Query Params' section shows a single parameter 'nome'. The response is a 200 OK status with a response time of 55 ms and a body size of 314 B. The response body is displayed in JSON format, showing a list of two client objects.

Key	Value	Description
nome		

```
[{"id": 1, "nome": "Artur", "cpf": "000.000.000-00", "endereco": [{"id": 2, "logradouro": "Rua", "numero": 0, "bairro": "Bairro", "cidade": "Cidade", "estado": null}]}]
```

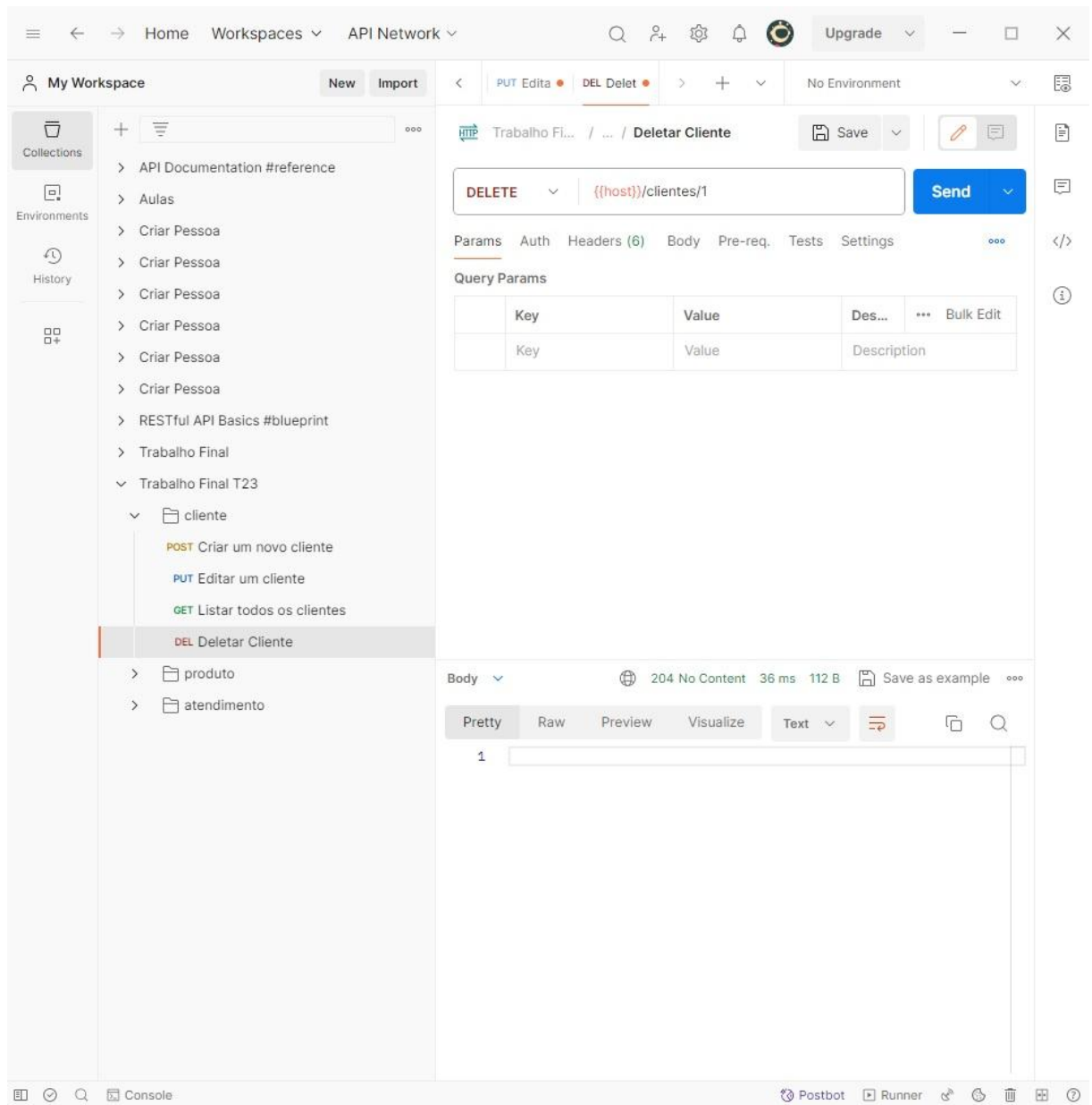
**Deletar Cliente:** Para remover um cliente, envie uma requisição **DELETE** para **/clientes/{id}**, onde **{id}** é o identificador único do cliente a ser removido.

- Método: **DELETE**
- Endpoint: **/clientes/{id}**

**Modelo (print screen):**

"204 No Content"

The server successfully processed the request, but is not returning any content."



## Produto

**Criar Produto:** Crie um novo produto enviando uma requisição **POST** para **/produtos**. No corpo da requisição, forneça os detalhes do produto, como nome, valor, descrição e tipo de animal associado.

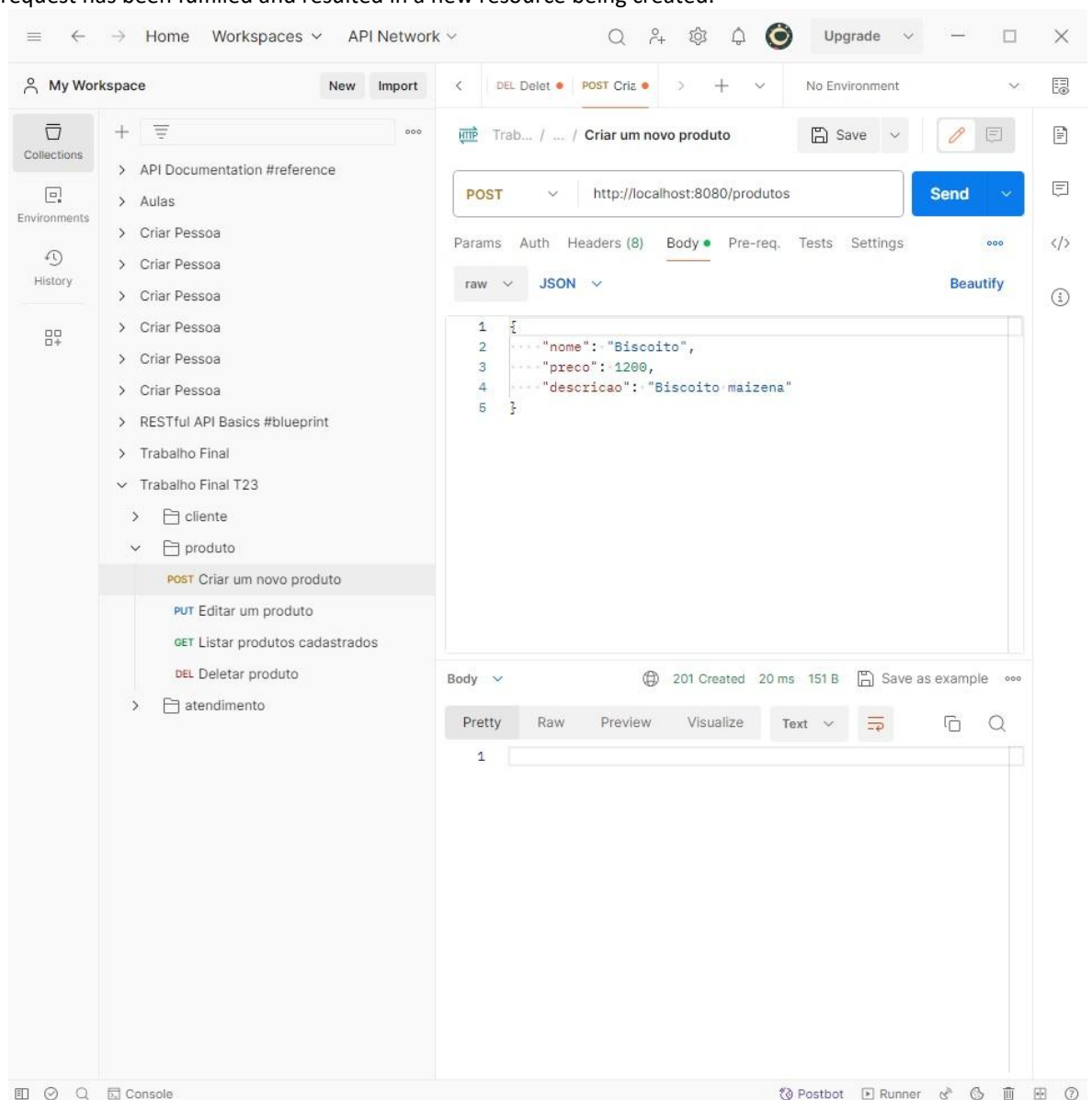
- Método: **POST**
- Endpoint: **/produtos**
- Corpo da requisição (JSON):

```
{  
  "nome": "Biscoito",  
  "preco": 1200,  
  "descricao": "Biscoito maizena"  
}
```

**Modelo (print screen):**

"201 Created"

The request has been fulfilled and resulted in a new resource being created."



**Editar Produto:** Para editar um produto existente, faça uma requisição **PUT** para **/produtos/{id}**, onde **{id}** é o identificador único do produto. No corpo da requisição, envie os dados atualizados.

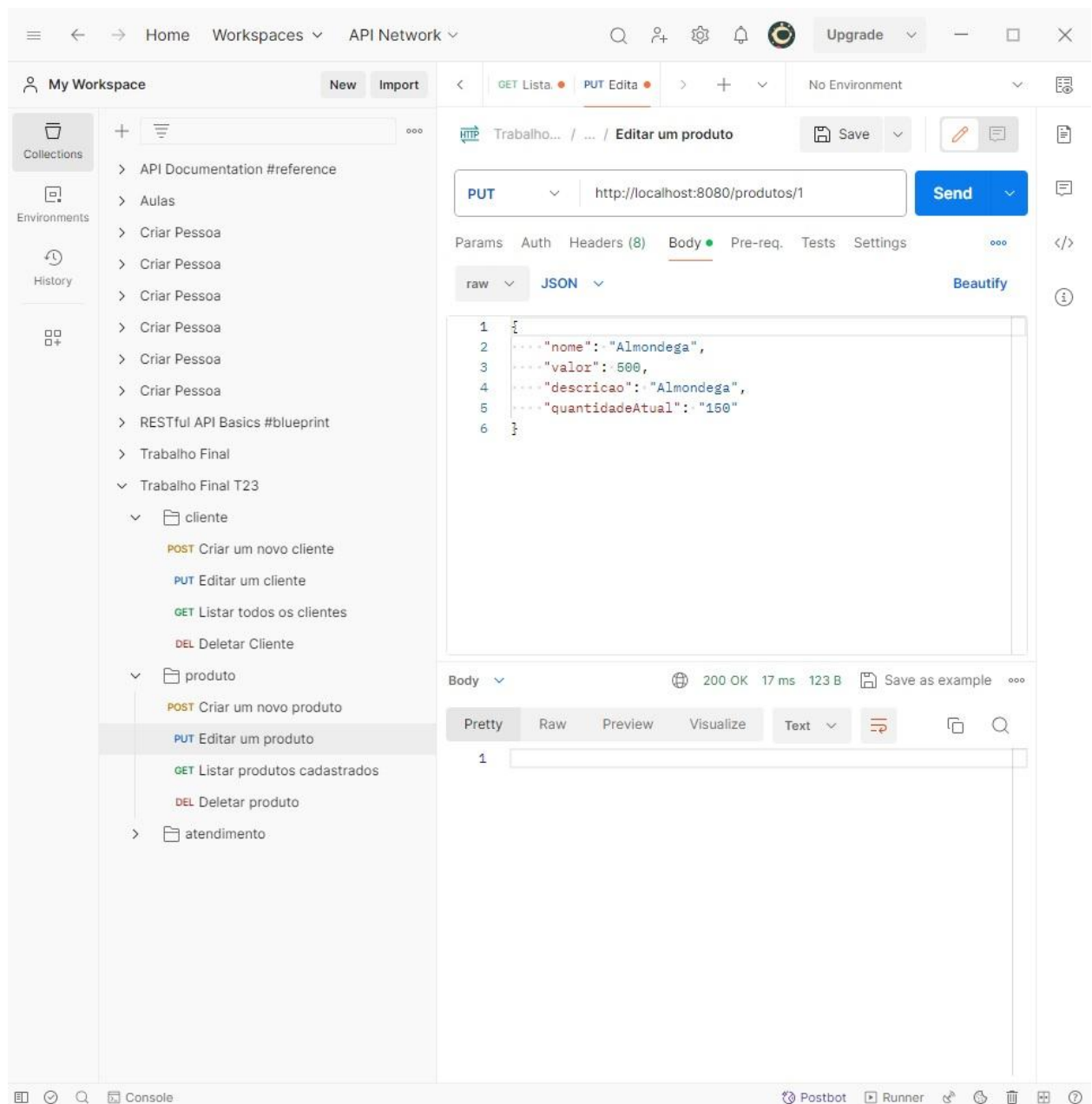
- Método: **PUT**
- Endpoint: **/produtos/{id}**
- Corpo da requisição (JSON):

```
{  
  "nome": "Almondega",  
  "valor": 500,  
  "descricao": "Almondega",  
  "quantidadeAtual": "150"  
}
```

### Modelo (print screen):

"200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action. "



**Listar Produtos:** Obtenha uma lista de todos os produtos fazendo uma requisição **GET** para **/produtos/listar**. Isso retornará uma lista de objetos Produto.

- Método: **GET**

- Endpoint: **/produtos?nome**

**Modelo (print screen):**

"200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action. "

The screenshot displays the Postman API client interface. On the left, the 'My Workspace' sidebar shows a collection of API endpoints under 'Trabalho Final T23'. The main panel shows a GET request to the endpoint `{{host}}/produtos?nome`. The 'Query Params' table lists parameters: `precoMaiorQue` (5), `precoMenorQue` (10), and `nome` (checked). The 'Body' tab shows the response in JSON format, which is a single object with the following fields: `nome` (Biscoito), `valor` (null), `descricao` (Biscoito maizena), and `quantidadeAtual` (null). The status bar at the bottom indicates a 200 OK response with a 13 ms latency and 252 B body size.

Key	Value	Description
<input type="checkbox"/>	<code>precoMaiorQue</code>	5
<input type="checkbox"/>	<code>precoMenorQue</code>	10
<input checked="" type="checkbox"/>	<code>nome</code>	

```
1 {
2   {
3     "nome": "Biscoito",
4     "valor": null,
5     "descricao": "Biscoito maizena",
6     "quantidadeAtual": null
7   }
8 }
```



**Deletar Produto:** Remova um produto enviando uma requisição **DELETE** para **/produtos/{id}**, onde **{id}** é o identificador único do produto a ser removido.

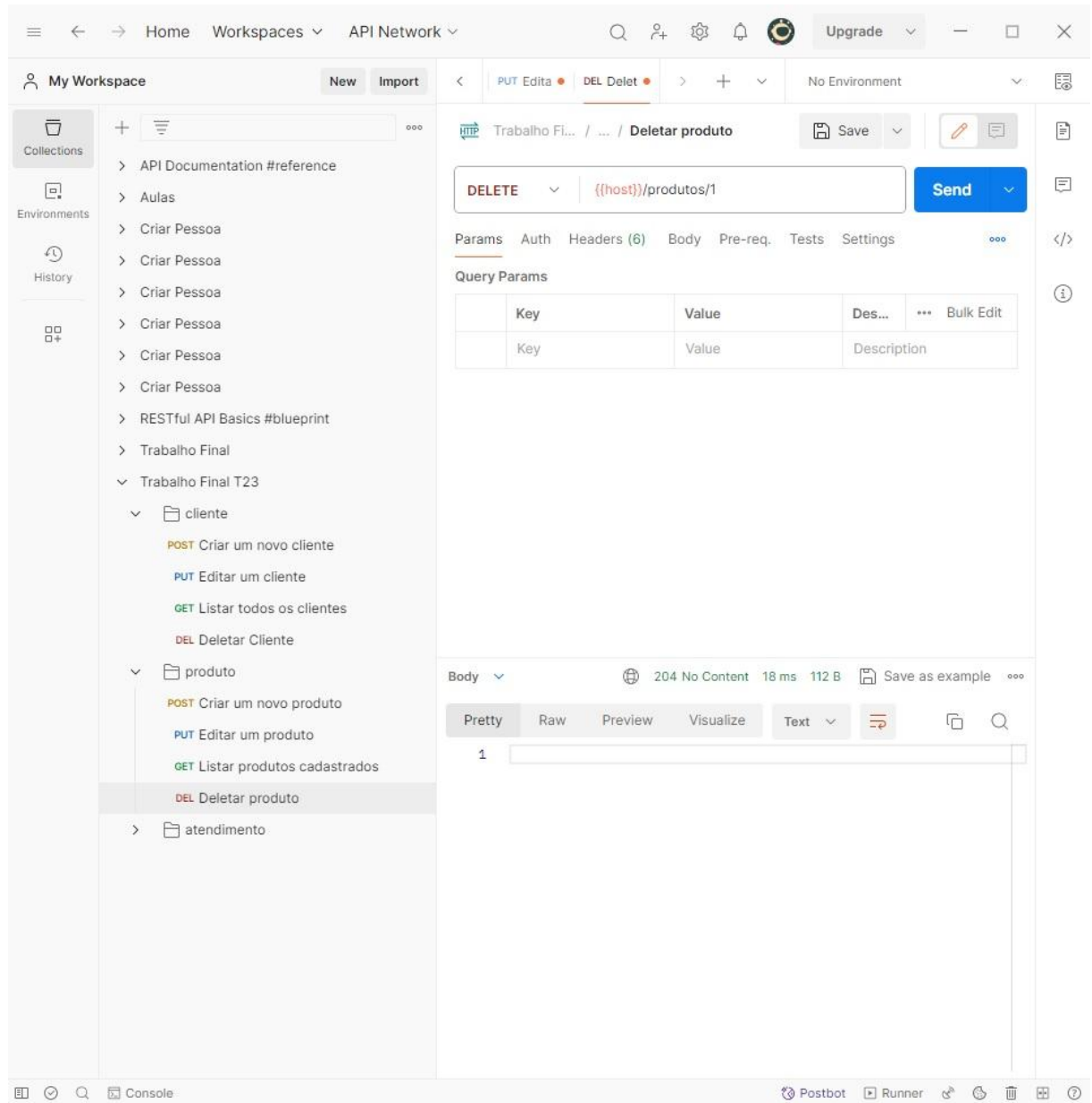
- Método: **DELETE**

- Endpoint: **/produtos/{id}**

**Modelo (print screen):**

"204 No Content"

The server successfully processed the request, but is not returning any content."



## Atendimento

**Iniciar Atendimento:** Inicie um novo atendimento enviando uma requisição **POST** para **/atendimentos**. No corpo da requisição, forneça os dados iniciais, como nome do atendente, veterinário, cliente, pet e produtos associados.

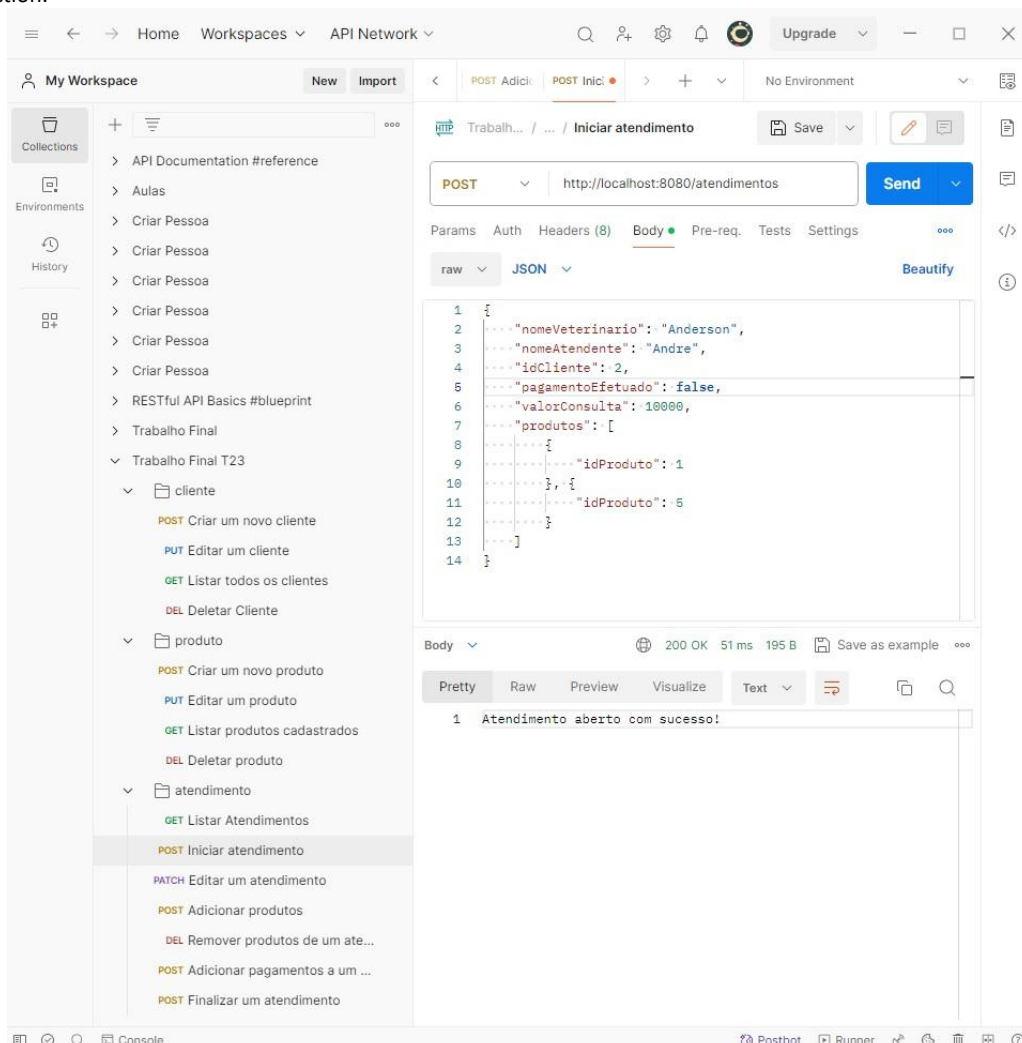
- Método: **POST**
- Endpoint: **/atendimentos**
- Corpo da requisição (JSON):

```
{
  "nomeVeterinario": "Anderson",
  "nomeAtendente": "Andre",
  "idCliente": 2,
  "pagamentoEfetuado": false,
  "valorConsulta": 10000,
  "produtos": [
    {
      "idProduto": 1
    }, {
      "idProduto": 5
    }
  ]
}
```

**Modelo (print screen):**

“200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action.”



**Editar Atendimento:** Para editar um atendimento existente, faça uma requisição **PUT** para **/atendimentos/editar/{id}**, onde **{id}** é o identificador único do atendimento. No corpo da requisição, envie os dados atualizados.

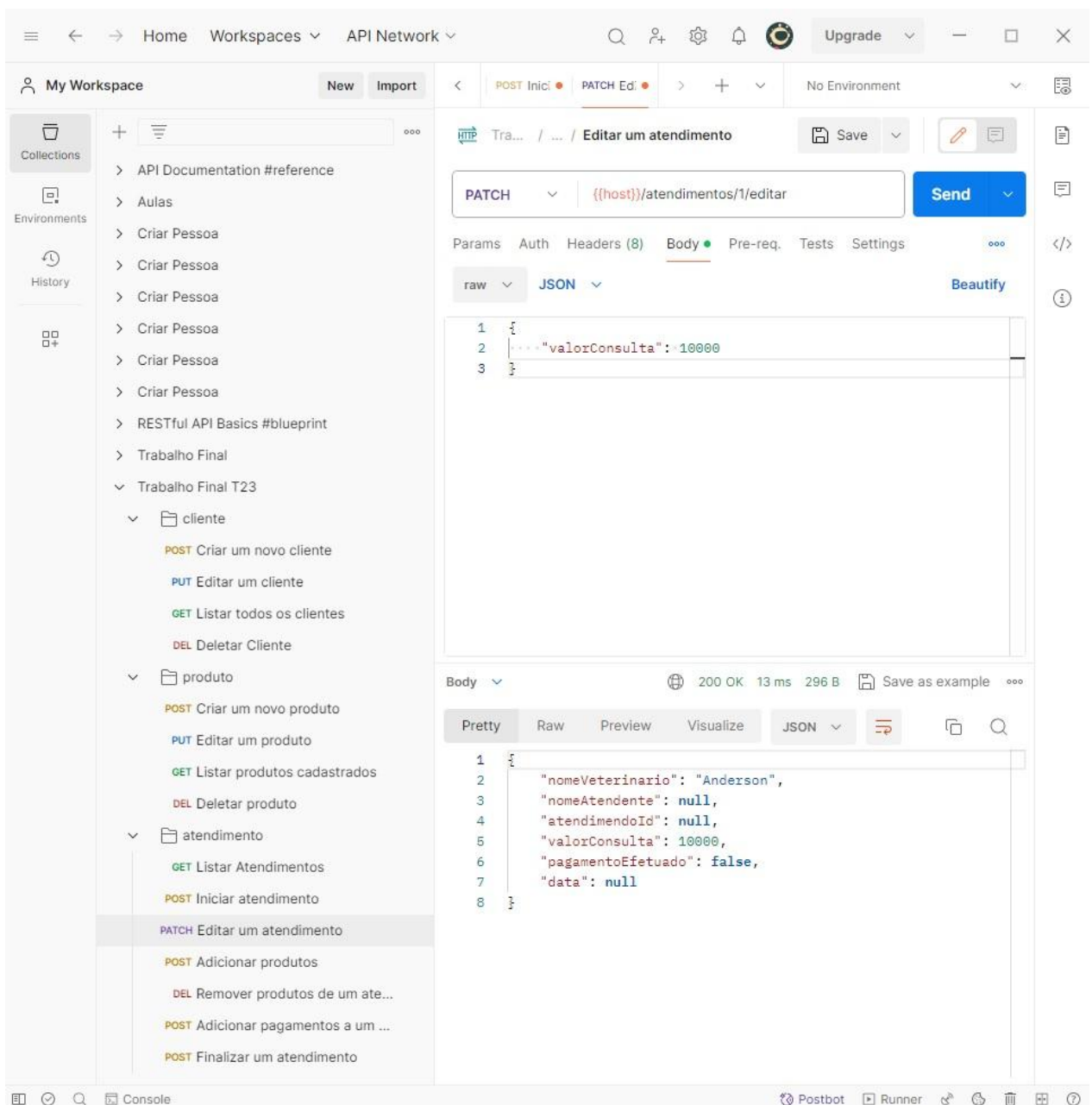
- Método: **PUT**
- Endpoint: **/atendimentos/{id}/editar**
- Corpo da requisição (JSON):

```
{  
  "valorConsulta": 10000  
}
```

**Modelo (print screen):**

"200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action. "



**Listar Atendimentos:** Obtenha uma lista de todos os atendimentos fazendo uma requisição **GET** para **/atendimentos**. Isso retornará uma lista de objetos Atendimento.

- Método: **GET**
- Endpoint: **/atendimentos**

*Modelo (print screen):*

"200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action."

The screenshot shows the Postman API client interface. On the left, the 'My Workspace' sidebar lists various API collections, with 'Trabalho Final T23' expanded to show a list of endpoints. The 'GET Listar Atendimentos' endpoint is selected. The main panel shows the request configuration: Method is 'GET', URL is 'http://localhost:8080/atendimentos'. Below the URL bar, the 'Query Params' table is empty. The 'Body' tab is selected, showing a '200 OK' status, '10 ms' time, and '295 B' size. The response body is displayed in 'Pretty' format, showing a JSON array with one object.

Key	Value	Des...	Bulk Edit
Key	Value	Description	

```
1 [
2   {
3     "nomeVeterinario": "Anderson",
4     "nomeAtendente": null,
5     "atendimentoId": 1,
6     "valorConsulta": 10000,
7     "pagamentoEfetuado": false,
8     "data": null
9   }
10 ]
```

**Remover Produto do Atendimento:** Remova um produto de um atendimento específico enviando uma requisição **DELETE** para **/atendimentos/atendimentos/{id}**, onde **{id}** é o identificador único do atendimento e **{produtoId}** é o identificador único do produto a ser removido.

- Método: **DELETE**
- Endpoint: **/atendimentos/{id}**

**Finalizar Atendimento:** Finalize um atendimento enviando uma requisição **PUT** para **/atendimentos/finalizar/{id}**, onde **{id}** é o identificador único do atendimento.

- Método: **PUT**
- Endpoint: **/atendimentos/finalizar/{id}**

**Modelo (print screen):**

"200 OK

Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action. "

The screenshot displays the Postman API client interface. On the left, the 'My Workspace' sidebar shows a collection of API endpoints under the 'Trabalho Final T23' folder, including 'cliente', 'produto', and 'atendimento'. The main panel shows a PUT request to the endpoint `{{host}}/atendimentos/1`. The 'Query Params' table is empty. The 'Body' tab is selected, showing a JSON payload in 'Pretty' format:

```
1 {
2   "nomeVeterinario": "Anderson",
3   "nomeAtendente": null,
4   "atendimentoId": 1,
5   "valorConsulta": 10000,
6   "pagamentoEfetuado": false,
7   "data": null
8 }
```

The status bar at the bottom indicates a successful response: `200 OK`, `9 ms`, and `293 B`.