

1- Elementos de entrada:

São os dados que a aplicação recebe para realizar suas tarefas. Por exemplo:

POST /cliente: para criar um novo cliente.

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

- POST /clientes**: The method and endpoint.
- Parameters**: A section with "No parameters" and buttons for "Cancel" and "Reset".
- Request body**: Labeled as "required", with a dropdown menu set to "application/json". The body contains a JSON object:

```
{  "idCliente": 1,  "nome": "Raissa Bernardo",  "cpf": "51078965423",  "telefone": "1677865342",  "email": "raissaantonis6@gmail.com"}
```
- Execute** and **Clear** buttons: Located below the request body.
- Responses**: A section showing the response details:
 - Code**: 200
 - Details**:
 - Response body**: A dark box containing the text "Inserido".
 - Response headers**: A section below the response body.

GET /cliente: para buscar informações de um cliente.

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

- Curl**: A section showing the curl command:

```
curl -X 'GET' \  'http://localhost:8080/clientes' \  -H 'accept: */*'
```
- Request URL**: A section showing the URL `http://localhost:8080/clientes`.
- Server response**: A section showing the response details:
 - Code**: 200
 - Details**:
 - Response body**: A dark box containing a JSON object:

```
{  "idCliente": 1,  "nome": "Raissa Bernardo",  "cpf": "51078965423",  "telefone": "1677865342",  "email": "raissaantonis6@gmail.com"}
```

PUT /cliente/{id}: para atualizar as informações de um cliente.

PUT /clientes/{idCliente}

Parameters

Name	Description
idCliente * required	
integer(\$int32)	1
(path)	

Request body required

application/json

```
{  "idCliente": 1,  "nome": "Raissa Antonia",  "cpf": "51069897876",  "telefone": "1677865342",  "email": "raissaantoniao@gmail.com"}
```

Execute Clear

Code Details

200

Response body

Atualizado

Download

DELET /cliente/{id}: para apagar as informações que foram inputadas no cliente com aquele id.

Request body required

application/json

1

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \  'http://localhost:8080/clientes' \  -H 'accept: */*' \  -H 'Content-Type: application/json' \  -d '1'
```

Request URL

http://localhost:8080/clientes

Server response

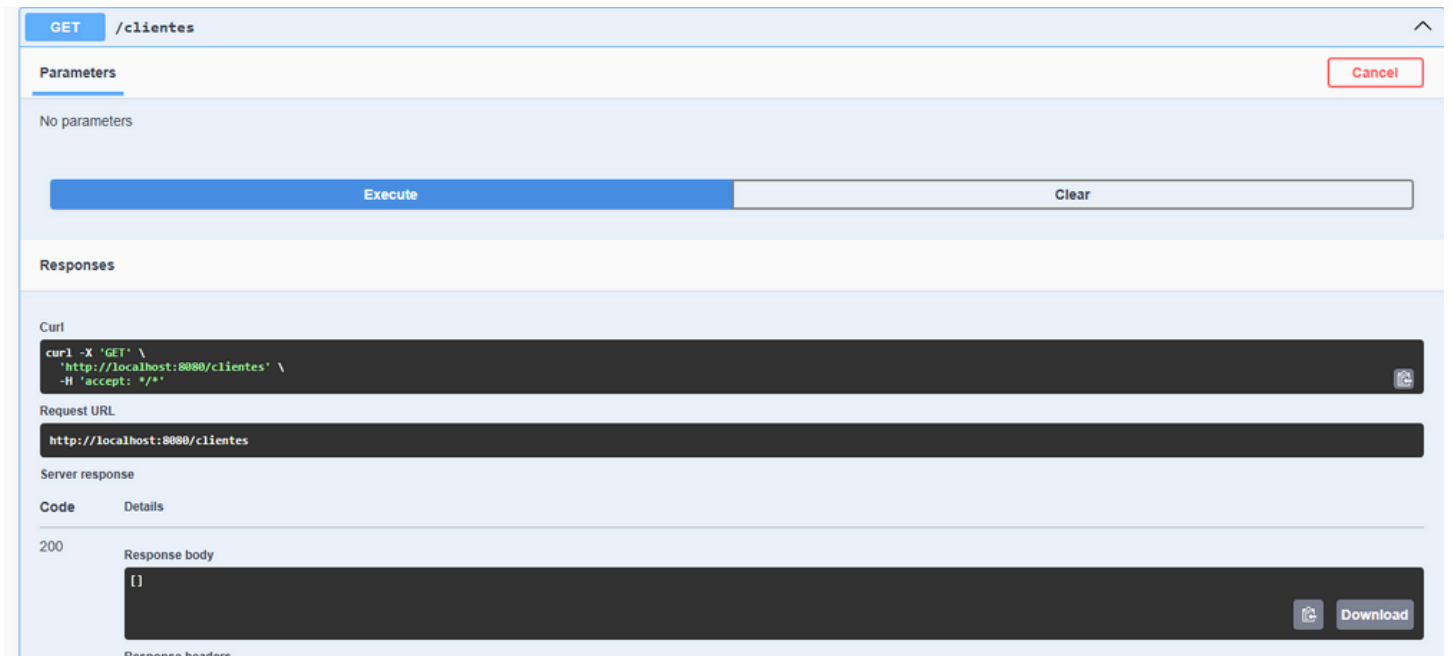
Code	Details
200	

Response body

Produto deletado com sucesso!

Download

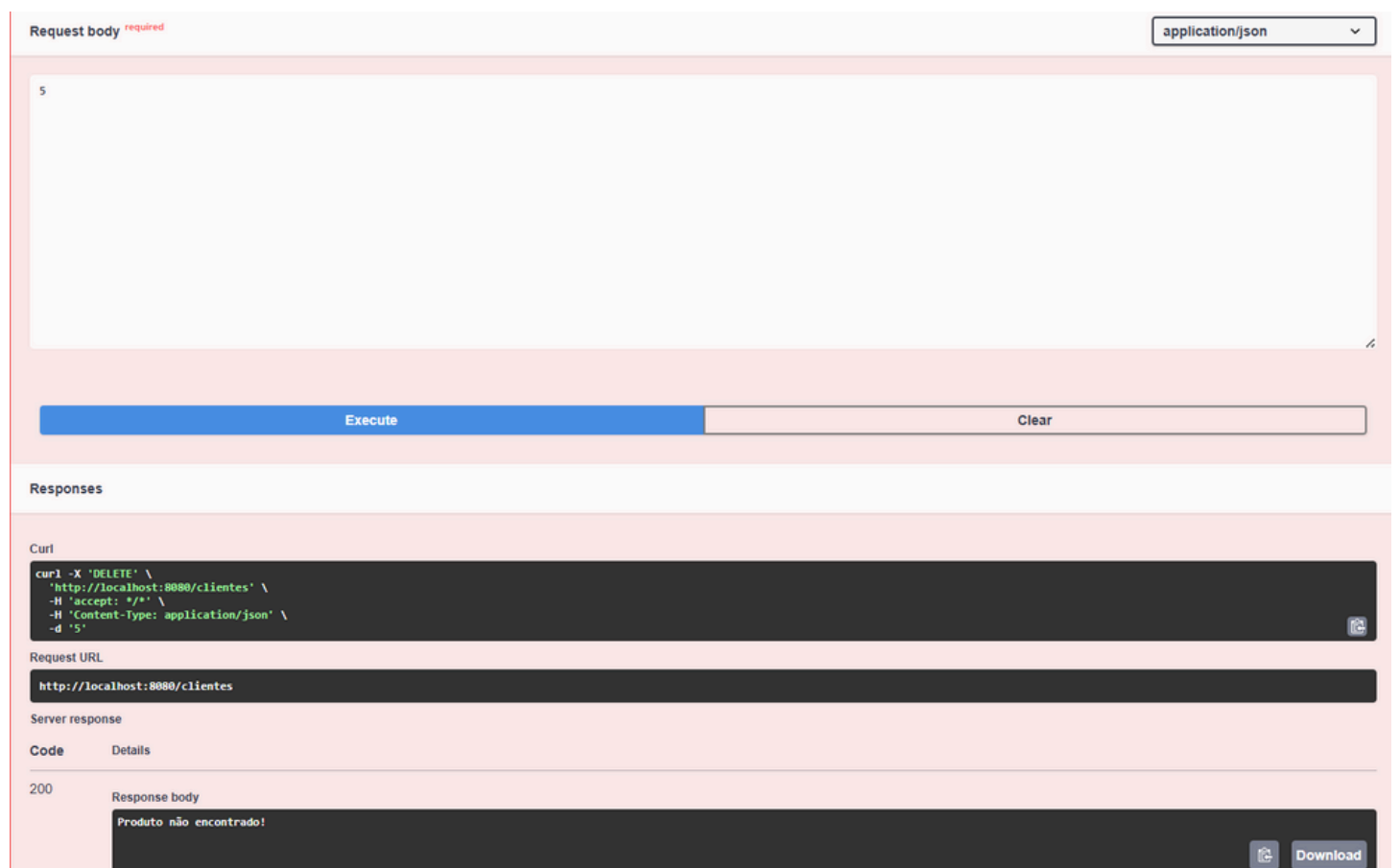
Depois, pode-se olhar ao get, que as informações deste cliente não estarão mais lá:



2- Processamento

Após receber os dados, o sistema faz o processamento para cumprir a tarefa solicitada.

Se o endpoint for DELETE/cliente/{id}, o sistema busca no banco de dados as informações do usuário com aquele id.



3- Saída

A saída é o que a aplicação retorna após processar a requisição.

Um bom exemplo é o get mostrado a cima:

Curl

```
curl -X 'GET' \
  'http://localhost:8080/clientes' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8080/clientes
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "idCliente": 1, "nome": "Raissa Bernardo", "cpf": "51078965423", "telefone": "1677865342", "email": "raissaantonio@gmail.com" }]</pre> <div> Download</div>