



Trabalho Serviços REST e Representação de Dados

Leonardo Quezado
MDCC – UFC | 2025.

Repositório do Código

Descrição da Implementação:

- Cliente REST desenvolvido em Python
- Biblioteca utilizada: requests
- API consumida: calculadora-fxpc.onrender.com
- Funcionalidades: listagem e execução de 4 operações matemáticas

Github: <https://github.com/LeonardoQuezado/trabalho-sistemas-distribuidos/tree/main/parte1-calculadora>

Análise com Postman 2.1. GET /operations - Listar Operações Disponíveis

Análise:

- Status: 200 OK
- Content-Type: application/json
- Headers importantes: Date, Content-Encoding, CF-RAY

The screenshot shows the Postman interface with a successful GET request to `https://calculadora-fxpc.onrender.com/operations`. The Headers tab displays the following key-value pairs:

Key	Value
Date	Sat, 06 Dec 2025 23:24:31 GMT
Content-Type	application/json
Content-Length	121
Connection	keep-alive
CF-RAY	9a9f8885db836f91-JDO
access-control-allow-origin	*
Content-Encoding	br
rndr-id	cc870ad4-9d63-4a1a
vary	Accept-Encoding
x-render-origin-server	gunicorn
cf-cache-status	DYNAMIC
Server	cloudflare
alt-svc	h3=":443"; ma=86400

The Response tab shows the following details:

- Status: 200 OK
- Time: 41.48 s
- Size: 500 B
- Save Response

At the bottom left, there is a note: "Create collections in Postman" and "Use collections to save your requests and share them with others."

GET /operations - Resposta (Body)

Payload retornado:

- 4 operações disponíveis
- Cada operação tem: method, name, path
- Formato: JSON estruturado

The screenshot shows a Postman interface with the following details:

- Request Method:** GET
- URL:** https://calculadora-fxpc.onrender.com/operations
- Headers:** 6 (shown in the Headers tab)
- Body:** JSON response (Pretty tab selected)
- Response Status:** 200 OK
- Response Time:** 41.48 s
- Response Size:** 500 B
- Response Content:** (JSON object with 4 operations)

```
1  [
2   "operations": [
3     {
4       "method": "POST",
5       "name": "Soma",
6       "path": "/operation/soma/param1/param2"
7     },
8     {
9       "method": "POST",
10      "name": "Subtracao",
11      "path": "/operation/subtracao/param1/param2"
12    },
13    {
14      "method": "POST",
15      "name": "Multiplicacao",
16      "path": "/operation/multiplicacao/param1/param2"
17    },
18    {
19      "method": "POST",
20      "name": "Divisao",
21      "path": "/operation/divisao/param1/param2"
22    }
23  ]
24 ]
```

Create collections in Postman
Use collections to save your requests and share them with others.

POST /operation/soma/10/5

Análise dos Headers da Requisição:

- Método: POST
- URL: /operation/soma/10/5
- Parâmetros enviados via path (na URL)
- Content-Length: 0 (sem body na requisição)

Query Params			
	Key	Value	Bulk Edit
	Key	Value	
Body	Cookies	Headers (13)	Test Results
	Key	Value	
	Date ⓘ	Sun, 07 Dec 2025 13:14:11 GMT	
	Content-Type ⓘ	application/json	
	Content-Length ⓘ	20	
	Connection ⓘ	keep-alive	
	CF-RAY ⓘ	9aa448dd0cc51dd3-JDO	
	access-control-allow-origin ⓘ	*	
	Content-Encoding ⓘ	br	
	rndr-id ⓘ	dbd7bf79-9735-4012	
	vary ⓘ	Accept-Encoding	
	x-render-origin-server ⓘ	gunicorn	
	cf-cache-status ⓘ	DYNAMIC	
	Server ⓘ	cloudflare	
	alt-svc ⓘ	h3=":443"; ma=86400	

POST /operation/soma/10/5 - Resultado

Análise do Payload:

- Operação: Soma ($10 + 5$)
- Resultado: 15.0
- Formato: JSON simples e direto
- Tipo de dado: float (número decimal)

The screenshot shows a dark-themed REST API testing interface. At the top, there's a header bar with tabs for "Body", "Cookies", "Headers (13)", and "Test Results". On the right side of the header, it displays "Status: 200 OK", "Time: 296 ms", and "Size: 398 B", along with a "Save Response" button. Below the header, there are two sections: "Query Params" and "Body". The "Query Params" section is currently empty. The "Body" section has a "Pretty" tab selected, showing the following JSON response:

```
1  {
2      "result": 15.0
3 }
```

POST /operation/subtracao/10/5

Análise dos Headers:

- Status Code: 200 OK
- Content-Type: application/json
- Mesmo padrão da operação anterior
- Requisição stateless (sem estado)

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (13) Test Results Status: 200 OK Time: 284 ms Size: 397 B Save Response ▾

Key	Value
Date ⓘ	Sun, 07 Dec 2025 13:17:15 GMT
Content-Type ⓘ	application/json
Content-Length ⓘ	19
Connection ⓘ	keep-alive
CF-RAY ⓘ	9aa44d5bb98412fd-JDO
access-control-allow-origin ⓘ	*
Content-Encoding ⓘ	br
rnldr-id ⓘ	3b51e881-6f13-41f0
vary ⓘ	Accept-Encoding
x-render-origin-server ⓘ	gunicorn
cf-cache-status ⓘ	DYNAMIC
Server ⓘ	cloudflare
alt-svc ⓘ	h3=":443"; ma=86400

POST /operation/subtracao/10/5 - Resultado

Análise do Payload:

- Operação: Subtração ($10 - 5$)
- Resultado: 5.0
- Formato consistente com outras operações

The screenshot shows a dark-themed interface of a REST API testing tool. At the top, there's a header bar with tabs for "Body", "Cookies", "Headers (13)", and "Test Results". The "Body" tab is selected, indicated by an underline. To the right of the tabs, status information is displayed: "Status: 200 OK", "Time: 284 ms", and "Size: 397 B". There's also a "Save Response" button.

Below the header, there's a table with three columns: "Key", "Value", and "Bulk Edit". The "Key" column contains the word "Key" repeated twice, and the "Value" column contains the word "Value" repeated twice. A "Bulk Edit" button is located in the top right corner of the table area.

Underneath the table, there's a toolbar with buttons for "Pretty", "Raw", "Preview", "Visualize", and "JSON". The "Pretty" button is currently selected, highlighted with a grey background. To the right of the toolbar, there are icons for copy, search, and refresh.

The main content area displays the JSON response in a pretty-printed format. The response consists of three numbered lines:

```
1  {
2   "result": 5.0
3 }
```

POST /operation/multiplicacao/10/5

Análise dos Headers:

- Status Code: 200 OK
- Content-Type: application/json
- API mantém padrão de resposta

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (13) Test Results

Status: 200 OK Time: 282 ms Size: 398 B Save Response ▾

Key	Value
Date ⓘ	Sun, 07 Dec 2025 13:18:01 GMT
Content-Type ⓘ	application/json
Content-Length ⓘ	20
Connection ⓘ	keep-alive
CF-RAY ⓘ	9aa44e79fded12fd-JDO
access-control-allow-origin ⓘ	*
Content-Encoding ⓘ	br
rndr-id ⓘ	f75aa0a8-1662-451f
vary ⓘ	Accept-Encoding
x-render-origin-server ⓘ	gunicorn
cf-cache-status ⓘ	DYNAMIC
Server ⓘ	cloudflare
alt-svc ⓘ	h3=":443"; ma=86400

POST /operation/multiplicacao/10/5 - Resultado

Análise do Payload:

- Operação: Multiplicação (10×5)
- Resultado: 50.0
- Resposta JSON consistente

The screenshot shows a REST API testing interface with the following details:

- Body:** The response body is displayed in a JSON editor. It contains one key-value pair: "result": 50.0.
- Headers:** There are 13 headers present in the response.
- Status:** The status is 200 OK.
- Time:** The response time is 282 ms.
- Size:** The size of the response is 398 B.

POST /operation/divisao/10/5

PAnálise dos Headers:

- Status Code: 200 OK
- Content-Type: application/json
- Padrão mantido em todas operações

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (13) Test Results Status: 200 OK Time: 263 ms Size: 397 B Save Response ▾

Key	Value
Date ⓘ	Sun, 07 Dec 2025 13:19:02 GMT
Content-Type ⓘ	application/json
Content-Length ⓘ	19
Connection ⓘ	keep-alive
CF-RAY ⓘ	9aa44ff72d3b280c-JDO
access-control-allow-origin ⓘ	*
Content-Encoding ⓘ	br
rndr-id ⓘ	59e3efb0-e36d-49c2
vary ⓘ	Accept-Encoding
x-render-origin-server ⓘ	gunicorn
cf-cache-status ⓘ	DYNAMIC
Server ⓘ	cloudflare
alt-svc ⓘ	h3=":443"; ma=86400

POST /operation/divisao/10/5 - Resultado

Análise do Payload:

- Operação: Divisão ($10 \div 5$)
- Resultado: 2.0
- API retorna float mesmo para divisão exata

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

- Body:** The response body is displayed in JSON format, showing a single key-value pair: "result": 2.0.
- Headers:** There are 13 headers present in the response.
- Status:** The status is 200 OK.
- Time:** The response time is 263 ms.
- Size:** The size of the response is 397 B.

Análise Detalhada via Postman

Cabeçalhos e Payloads

REQUISIÇÃO GET /operations:

Headers enviados:

- User-Agent: PostmanRuntime/7.x.x (identifica o cliente)
- Accept: */* (aceita qualquer tipo de resposta)
 - Accept-Encoding: gzip, deflate (aceita compressão)
- Connection: keep-alive (mantém conexão aberta)

Payload: Nenhum (GET não envia body)

RESPOSTA GET /operations:

Headers recebidos:

- Content-Type: application/json (servidor retorna JSON)
- Content-Encoding: gzip (resposta comprimida)
 - Content-Length: 121 bytes (tamanho do payload)
- access-control-allow-origin: * (permite qualquer origem)

Payload: JSON com array de 4 operações disponíveis

Análise POST: REST vs Formulário Web

POST REST vs Formulário HTML

Diferenças e Similaridades

Identificadas SIMILARIDADES:

- ✓ Ambos usam método POST do protocolo HTTP
- ✓ Ambos enviam dados ao servidor para processamento
- ✓ Ambos recebem resposta com status code
- ✓ Estrutura similar: Headers + Body (quando presente)

DIFERENÇAS OBSERVADA S NO POSTMAN:

- HEADERS DE REQUISIÇÃO:
 - API REST: Accept header espera application/json ou */*
 - Formulário Web: Accept header espera text/html
- CONTENT-TYPE DA REQUISIÇÃO:
 - API REST: application/json OU parâmetros diretos na URL (path params)
 - Formulário Web: application/x-www-form-urlencoded ou multipart/form-data
- PAYLOAD ENVIADO:
 - API REST: JSON estruturado {"num1": 10, "num2": 5} OU vazio (quando parâmetros estão na URL)
 - Formulário Web: Formato URL encoded operacao=soma&num1=10&num2=5
- CONTENT-TYPE DA RESPOSTA:
 - API REST: application/json (retorna apenas dados estruturados)
 - Formulário Web: text/html (retorna página HTML completa)
- PAYLOAD RECEBIDO:
 - API REST: {"result": 15.0} - somente o dado necessário
 - Formulário Web: <html><body>Resultado: 15</body></html> - página inteira
- COMPORTAMENTO PÓS-REQUISIÇÃO:
 - API REST: Responde direto com status 200 OK, sem redirecionamento
 - Formulário Web: Comum redirecionar (302/303) para outra página
- GERENCIAMENTO DE ESTADO:
 - API REST: Stateless - não utiliza cookies ou sessões
 - Formulário Web: Stateful - geralmente usa Set-Cookie para manter sessão