

A expressão `parseFloat(url.searchParams.get('num1'))`; no código faz duas coisas principais:

Obtém o valor do parâmetro 'num1' da URL:

- `url.searchParams.get('num1')` procura o valor do parâmetro `num1` na URL.
- Por exemplo, se a URL for `http://localhost:3000/?num1=5&num2=3`, essa linha encontrará o valor "5" para `num1` como uma *string*.

Converte a string para um número decimal:

- `parseFloat()` pega a string encontrada e a converte para um número decimal (ou número de ponto flutuante).
- Por exemplo, `parseFloat("5")` resulta em 5 como um número.

Portanto, `parseFloat(url.searchParams.get('num1'))`; significa: "pegue o valor do parâmetro `num1` na URL e o transforme em um número".

```
// Importa o módulo 'http' do Node.js para criar um servidor web
const http = require('http');

// Cria o servidor HTTP que irá responder a requisições
const server = http.createServer((req, res) => {
  // Cria um objeto URL com a URL da requisição (req.url) e o host do servidor (req.headers.host)
  const url = new URL(req.url, `http://${req.headers.host}`);

  // Obtém o valor do parâmetro 'num1' na URL, converte para número com parseFloat
  const num1 = parseFloat(url.searchParams.get('num1'));
  // Obtém o valor do parâmetro 'num2' na URL, converte para número com parseFloat
  const num2 = parseFloat(url.searchParams.get('num2'));

  // Verifica se 'num1' ou 'num2' não são números válidos (NaN)
  if (isNaN(num1) || isNaN(num2)) {
    // Define o código de status como 400 (Bad Request) indicando erro de cliente
    res.statusCode = 400;
    // Envia a mensagem de erro indicando que são necessários números válidos
    res.end('Por favor, forneça dois números válidos nos parâmetros "num1" e "num2".');
    return; // Sai da função se houve erro, para não continuar com a soma
  }

  // Realiza a soma dos dois números
  const soma = num1 + num2;
  // Define o cabeçalho da resposta como texto simples
  res.setHeader('Content-Type', 'text/plain');
  // Envia a resposta com o resultado da soma
  res.end(`A soma é: ${soma}`);
});

// Configura o servidor para escutar na porta 3000 e imprime uma mensagem no console
server.listen(3000, () => {
  console.log('Servidor rodando na porta 3000...');
});
```

A expressão `isNaN(num1) || isNaN(num2)` está verificando se algum dos dois valores, `num1` ou `num2`, **não é um número**.

Vamos dividir em partes para entender melhor:

1.isNaN: Esse é um comando que significa "é um valor que **não é número**?". Ele devolve `true` se o valor não for um número e `false` se for um número.

2.isNaN(num1): Aqui, o código está perguntando: "**num1 não é um número**?". Se `num1` não for um número (como uma letra, por exemplo), isso vai ser `true`.

3.||: Esse símbolo significa "ou". Quer dizer que o código vai dar `true` se **qualquer uma das perguntas for verdadeira**.

4.isNaN(num1) || isNaN(num2): No final, essa linha está perguntando:

- "num1 não é um número?"
- "OU num2 não é um número?"

Se qualquer uma dessas respostas for `true` (ou seja, se um deles não for um número), o resultado final será `true`.