



INTEGRATION TESTING: POSTMAN

Discentes:

- Rosimere Martins
- Raimunda Daniela
- Sibele Almeida
- Rian Veigas

➤ **Docente:** Prof. Fabricio Farias

Baião

Agosto/2025

O que é o Postman?



É uma plataforma colaborativa de API líder mundial, utilizada por desenvolvedores para criar, testar, documentar e colaborar em APIs.

Entrega solicitações e recebe respostas, assim como um serviço postal.



Características Principais

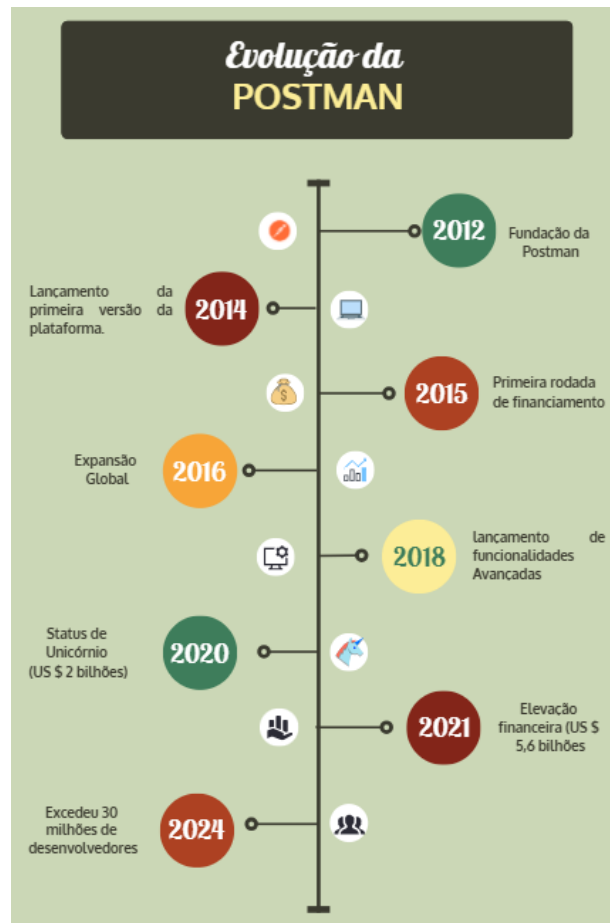


Criação e Gestão de Solicitações:

- **Editor Visual Poderoso:** Permite criar e configurar solicitações HTTP (REST e SOAP), definindo métodos, URLs, parâmetros, cabeçalhos, corpos de solicitação e configurações de autenticação.
- **Coleções:** Agrupamento de solicitações em coleções para organizar o trabalho e facilitar o compartilhamento e a reutilização.




Evolução da Plataforma Postman



Postman




1º Passo: Download

[Product](#) [Pricing](#) [Enterprise](#) [Resources](#) [Contact Sales](#) [Launch Postman](#)

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

The Postman app

Download the app to get started with the Postman API Platform.

 **Windows 64-bit**

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

[Release Notes](#) →

Not your OS? Download for Mac ([Intel Chip](#), [Apple Chip](#)) or Linux

Home Workspaces API Network Search

Notion's Public Workspace New Import Overview GET Retrieve a database

Collection

Environment

History

+

Notion API

> Users

> Databases

> GET Retrieve a database

> 200 Success- Retrie...

> POST Query a database

> POST Create a database

> PATCH Update a database

> Pages

> Users

> Blocks

HTTP

Notion API / Databases / Retrieve a database

GET https://api.notion.com/v1/databases/id

Params Auth Headers(10) Body Scripts

Welcome to Postman!

How can we help you today?

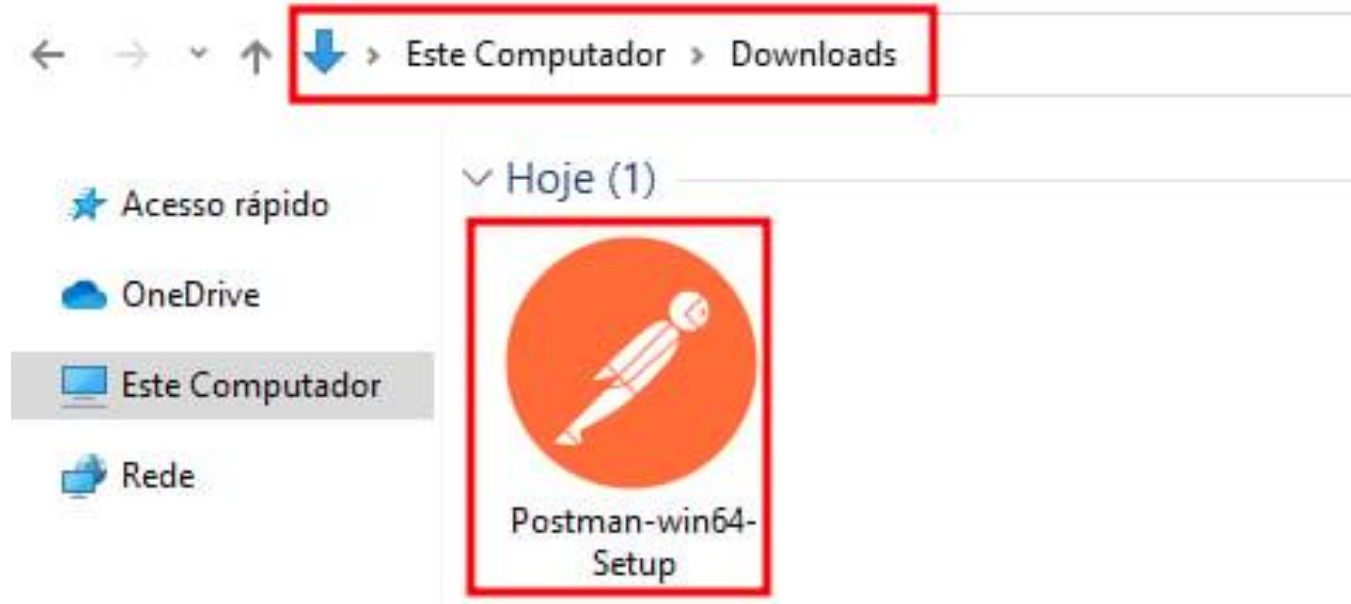
This chat may be monitored and recorded in accordance with our [Privacy Policy](#).

KEY	VALUE
id	{{DATA}}

Postman



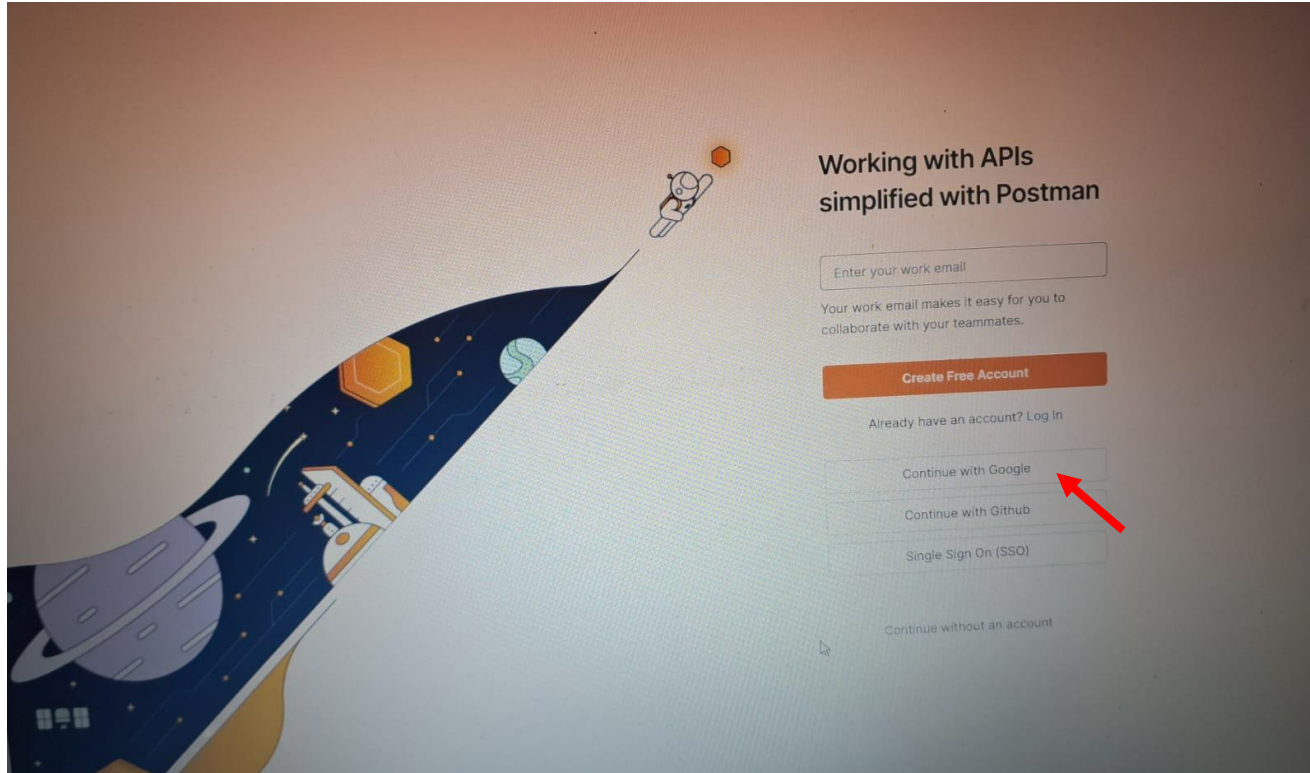
2º Passo: Acessar a pasta → Instalação



Postman



3º Passo: Login do programa



Postman



4º Passo: Login do programa

The image shows the Postman login interface on the left and a preview of the workspace on the right. The login screen has a dark background with an orange Postman logo at the top left. The text 'Bem-vindo ao Postman! Conte-nos um pouco sobre você.' is displayed. Below it, there is a form with two sections: 'Qual o seu nome?' with a text input field containing 'Kai Tocantins', and 'O que melhor descreve sua função?' with a grid of role buttons. A red arrow points to the 'Estudante / Educador' button. At the bottom right of the form is a 'Continuar' button. The right side shows a preview of the 'Kai's Workspace' interface, which has a dark theme. It features a 'GET' method selector, a URL bar, and a 'Kai' user profile icon.

Bem-vindo ao Postman! Conte-nos um pouco sobre você.

Qual o seu nome?

Kai Tocantins

O que melhor descreve sua função?

Desenvolvedor Backend Engenheiro de Dados

Engenheiro DevOps Desenvolvedor Frontend

Desenvolvedor Fullstack Estudante / Educador

Engenheiro de Qualidade / Testador Outro

Continuar

Kai's Workspace


GET

Kai

Postman



5º Passo: Login do programa



← Monte sua equipe

Comece a colaborar em APIs agora mesmo convidando seus colegas de trabalho para sua equipe.

Nome da equipe

Kai Tocantins's Team

Convide seus companheiros de equipe

Endereço de email

Endereço de email

+ Adicionar mais

[Copiar link de convite](#)

Começar

Kai Tocantins's Workspace

PM

GET

Back-end


QA

Front-end

Postman



6º Passo: Login do programa

 [Continuar com o plano gratuito →](#)

← **Você está no plano gratuito. Atualize para atender às suas necessidades de API.**

RECOMENDADO

Básico

Se você precisa de acesso centralizado à API, compartilhamento fácil e testes locais.

\$ 14 por assento/mês, cobrado anualmente

Selecione o plano Experimente grátis por 30 dias

Não é necessário cartão de crédito para o teste.

Principais características

- ✓ Companheiros de equipe ilimitados
- ✓ Espaços de trabalho internos e públicos ilimitados
- ✓ 10.000 solicitações de servidor simulado

Profissional

Se você precisa de gerenciamento seguro de funções e deseja conectar diferentes equipes.

\$ 29 por assento/mês, cobrado anualmente

Selecione o plano Experimente grátis por 30 dias

Não é necessário cartão de crédito para o teste.

Principais características

- ✓ Tudo no Básico
- ✓ RBAC (Controle de acesso baseado em função)
- ✓ Espaços de trabalho internos (privados)
- ✓ Espaços de trabalho para parceiros individuais e múltiplos

Empresa

Se você precisa de segurança, conformidade, planejamento e funções em toda a organização.

\$ 49 por assento/mês, cobrado anualmente

Selecione o plano Entre em contato com vendas

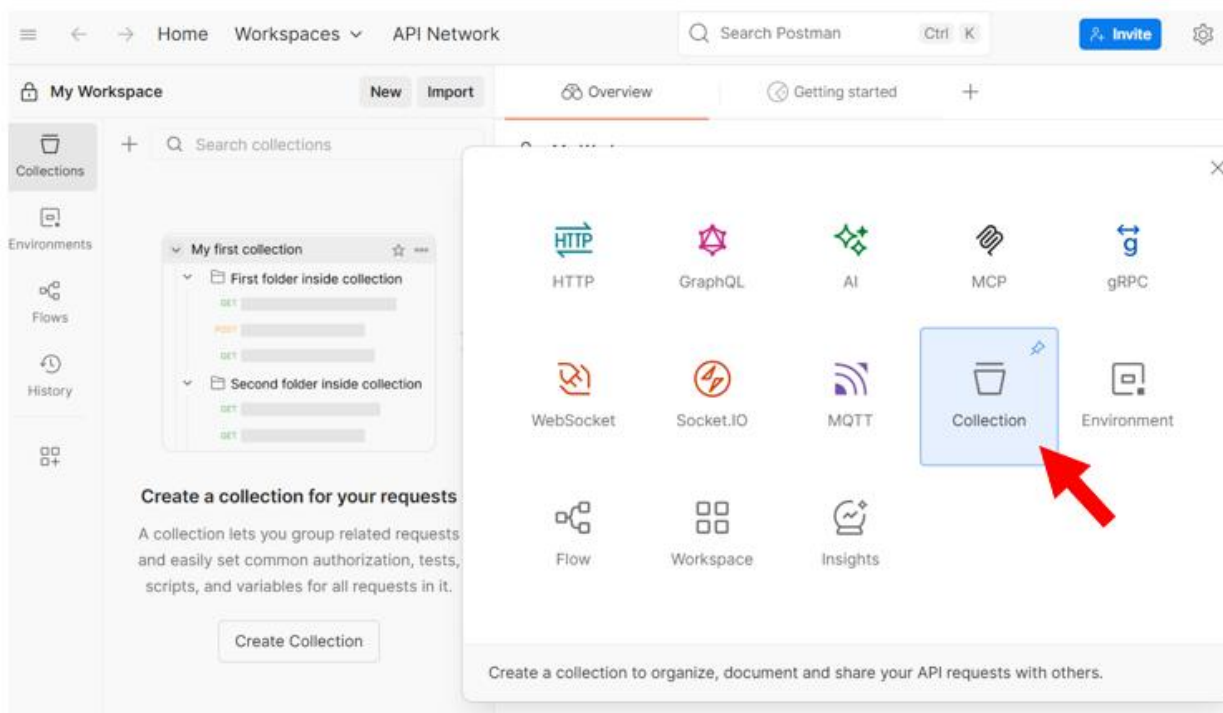
Principais características

- ✓ Tudo em Profissional
- ✓ SSO, SCIM, SAML
- ✓ RBAC avançado (controle de acesso baseado em função)
- ✓ Relatórios e análises avançadas



Postman

7º Passo: Criando uma Coleção de Testes



Uma coleção é essencial para a organização. Em projetos reais, você teria coleções separadas para diferentes módulos da sua API (Coleção de Autenticação, Coleção de Usuários, Coleção de Produtos, etc.). Isso facilita a execução de testes específicos e a manutenção do projeto.

Postman



8º Passo: Nomear a Coleção

A imagem é uma captura de tela da interface do usuário do Postman. No topo, há uma barra de navegação com links para Home, Workspaces e API Network, além de uma barra de busca e botões para Invite, Upgrade e outros. Abaixo, a seção "My Workspace" contém abas para "New" e "Import". No lado esquerdo, há uma barra lateral com ícones para Collections, Environments, Flows e History. A coleção "Teste de Integração" está selecionada e expandida, mostrando o texto "This collection is empty. Add a request to start working." No topo da coleção, há uma barra de ferramentas com ícones para Save, Run, Fork e Share. Abaixo disso, há uma barra de navegação para a coleção com abas para Overview, Auth, Scripts, Tests, Variables e Runs. A aba "Overview" está selecionada, e o título "Teste de Integração" é exibido em grande. Uma seta vermelha aponta para o ícone de nomeação (um ícone de documento com uma seta) na barra de ferramentas da coleção. À direita da coleção, há uma seção de estatísticas com os seguintes dados: 0 requests, 1 view, 0 forks, 0 watchers, e "Created by: You". Abaixo disso, há uma seção de "Pinned Environments" com o texto "Pin an environment to".

My Workspace

New Import

Overview Getting started Teste de Integração

Search collections

Teste de Integração

This collection is empty. Add a request to start working.

Overview Auth Scripts Tests Variables Runs

Teste de Integração

Make things easier for your teammates with a complete collection description.

View complete documentation →

0 requests

1 view

0 forks

0 watchers

Created by: You

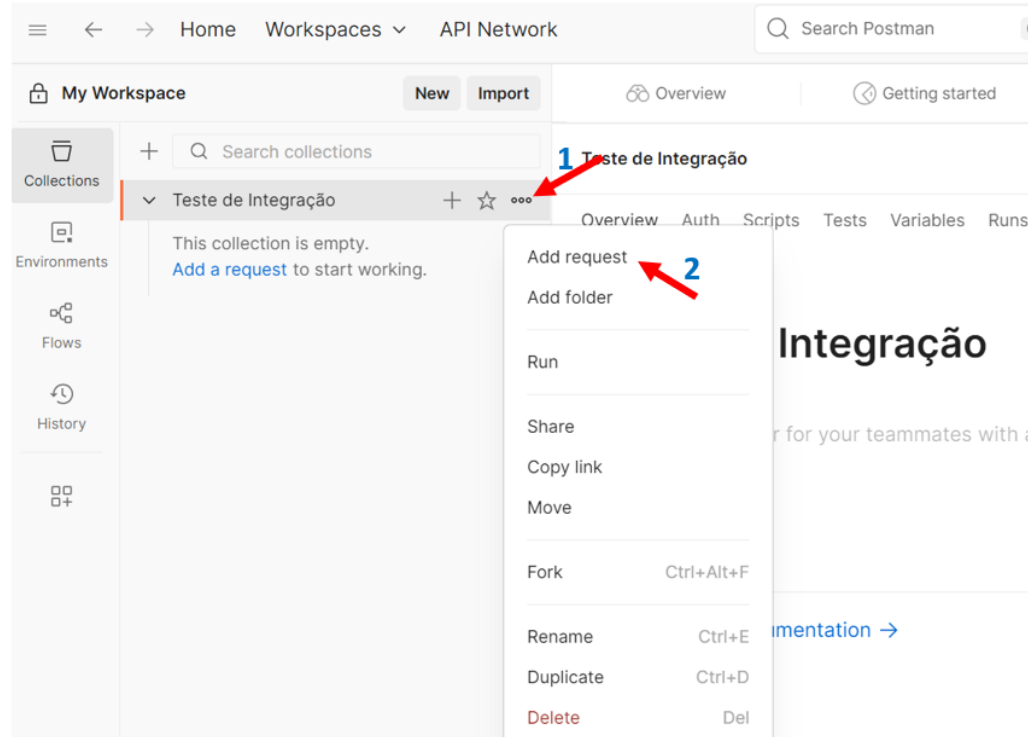
Pinned Environments

Pin an environment to

Postman



9º Passo: Adicionar primeira requisição “Listar post do usuário”(GET)



Postman



10º Passo: Adicionar primeira requisição “Listar post do usuário”(GET)

A imagem mostra a interface do Postman com o seguinte conteúdo:

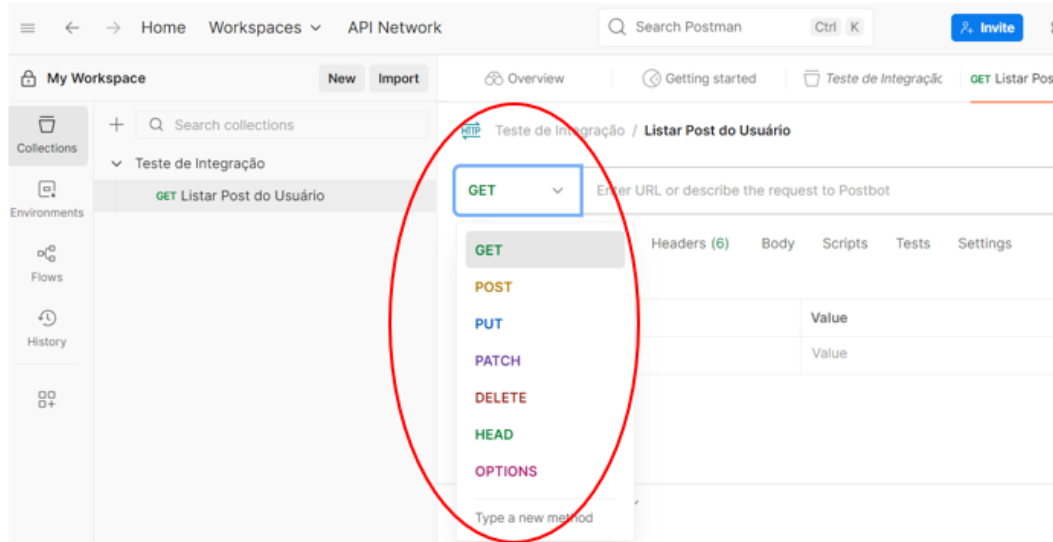
- Top Bar:** Home, Workspaces, API Network, Search Postman, Ctrl K, Invite button.
- Left Sidebar:** My Workspace, Collections (Search collections), Environments, Flows, History.
- Workspace:** Overview, Getting started, Teste de Integração, GET Listar Post do Usuar.
- Request Editor:** GET method selected, URL field with placeholder "Enter URL or describe the request in Postbot". A red arrow points to the URL field.
- Tabs:** Params, Authorization, Headers (6), Body, Scripts, Tests, Settings.
- Query Params Table:**

Key	Value	Description
Key	Value	Description

Postman



11º Passo: Escolher o método



Método GET: É o método mais seguro e idempotente, ou seja, você pode executá-lo quantas vezes quiser que ele não altera o estado do servidor.

Postman



API JSONPlaceholder

É uma API REST falsa e gratuita que serve dados em formato JSON. Criada para desenvolvedores que precisam de um backend para testar e prototipar sem a necessidade de criar um.

Endpoint: É a URL que representa um recurso no servidor. Por exemplo, <https://jsonplaceholder.typicode.com/posts> é o endpoint para acessar os posts.

Recursos: São os dados que a API gerencia, como posts, usuários, comentários, etc. Cada recurso tem um ID único.

Postman



12º Passo: Integração da API publica

URL

<https://jsonplaceholder.typicode.com/posts?userId=1>: A URL é dividida em:

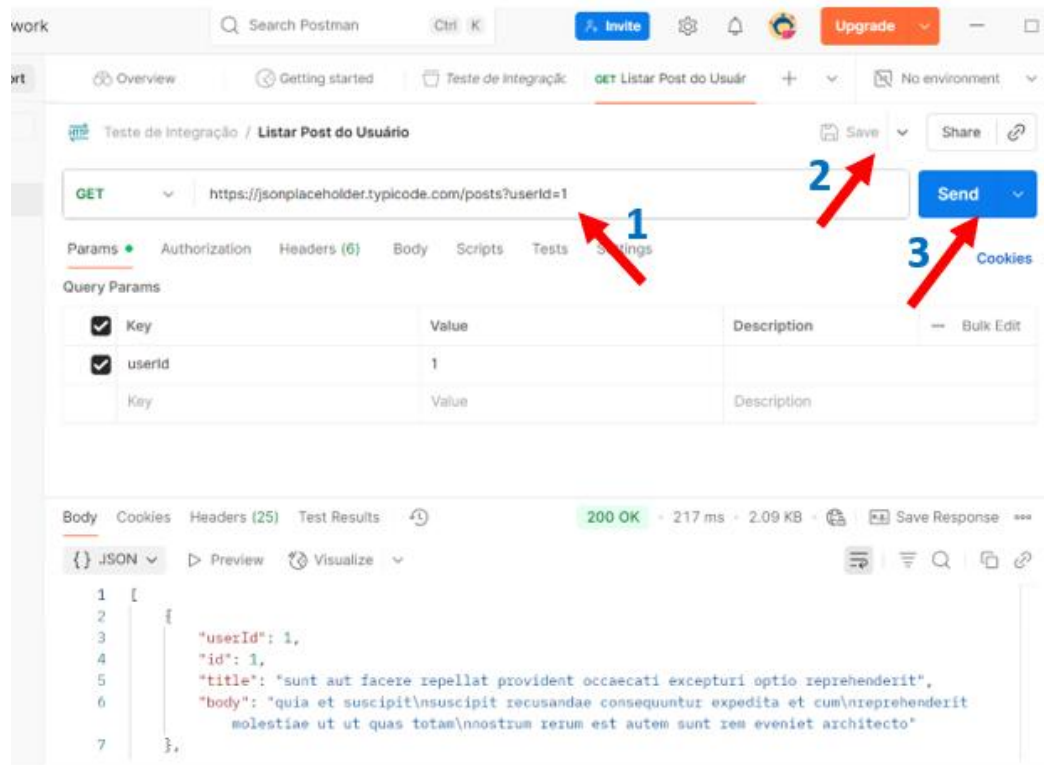
Base

<https://jsonplaceholder.typicode.com>

URL:

Endpoint: /posts.

Query Parameter: ?userId=1. O ? indica o início de um parâmetro, e o userId=1 é o filtro. Neste caso, estamos pedindo à API que retorne apenas os posts onde o userId é igual a 1.



Postman



A JSONPlaceholder usa os métodos HTTP padrão para interagir com seus recursos:

GET: Para buscar/ler dados.

POST: Para criar um novo dado.

PUT / PATCH: Para atualizar dados.

DELETE: Para apagar dados.

Quando você envia uma requisição para a JSONPlaceholder, ela simula como um servidor de verdade responderia, mas os dados não são realmente persistidos. Cada requisição POST ou PUT retorna uma resposta de sucesso, mas não altera o "banco de dados" da API. Isso a torna perfeita para aprendizado e testes.

Postman



Teste de validação

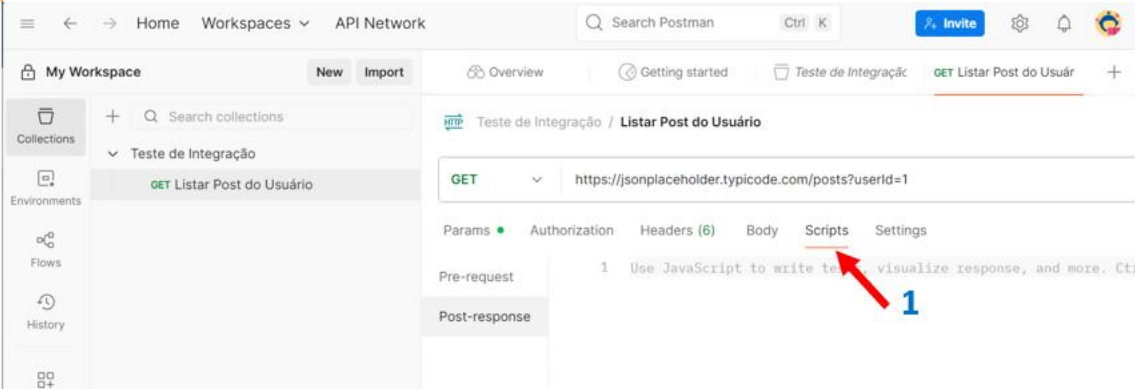
Esta é a parte crucial do teste de integração no Postman. A aba Tests permite escrever scripts em JavaScript para validar a resposta.

- O primeiro teste verifica se o código de status HTTP é 200 (OK), indicando sucesso.
- O segundo teste verifica se o número de itens na resposta é 10, como esperado para este usuário.
- O terceiro teste verifica se o primeiro item da lista tem a propriedade `userId`.

Postman



13º Passo: Adicionando Testes Scripts → código



Inserir código

2

```
pm.test("Status da resposta é 200 OK", function () {  
  pm.response.to.have.status(200);});
```

```
pm.test("Resposta contém 10 posts para o userId 1", function () {  
  const responseData =  
    pm.response.json();  
  pm.expect(responseData.length).to.  
    eql(10);});
```

```
pm.test("O primeiro post tem a propriedade 'userId'", function () {  
  const responseData =  
    pm.response.json();  
  pm.expect(responseData[0]).to.hav  
    e.property('userId');});
```

Postman



14º Passo: Resultado do Teste

A interface do Postman mostra o teste de integração "Listar Post do Usuário" executado com sucesso. O resultado é "200 OK" com uma duração de 222 ms e um tamanho de 2.09 KB. O código de teste está visível na aba "Scripts".

Código inserido:

```
1 pm.test("Status da resposta é 200 OK", function () {  
2   pm.response.to.have.status(200);  
3 });  
4  
5 pm.test("Resposta contém 10 posts para o userId 1", function () {  
6   const responseData = pm.response.json();  
7   pm.expect(responseData.length).to.eql(10);  
8 });
```

Resultados do Teste:

- PASSED** Status da resposta é 200 OK
- PASSED** Resposta contém 10 posts para o userId 1
- PASSED** O primeiro post tem a propriedade 'userid'

As setas numeradas indicam as seguintes ações:

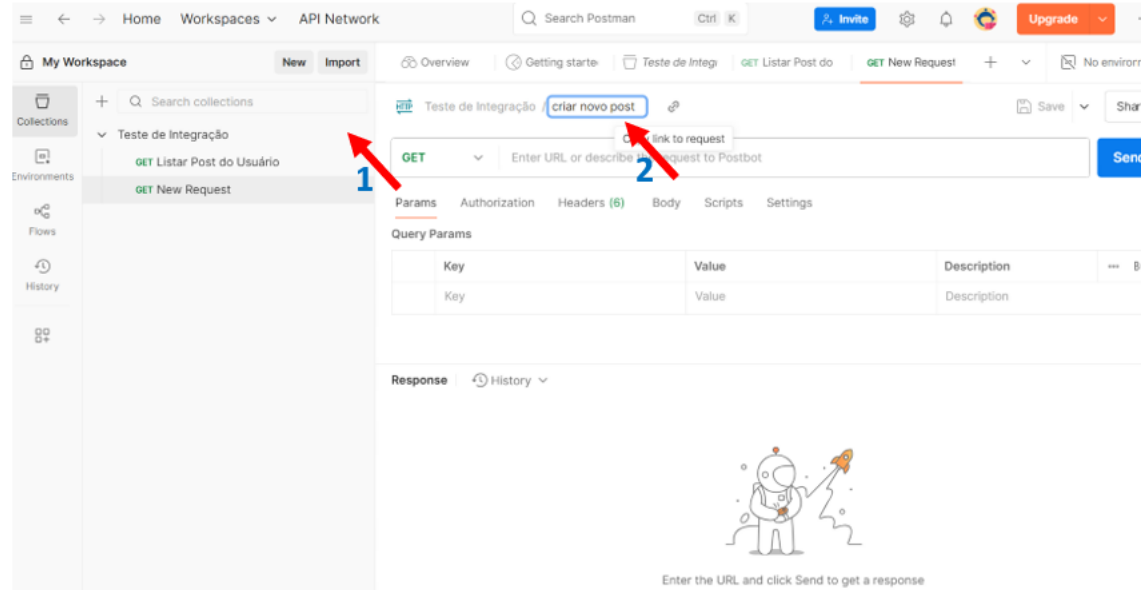
- 3: Botão "Save" (Salvar)
- 4: Botão "Send" (Enviar)
- 5: Aba "Test Results" (Resultados do Teste)

Postman



15º Passo: Adicionar segunda requisição “Criar novo post” (POST)

1: Clicar “...” e
selecionar
“Add request”



Postman



Requisição POST para Criar Dados

Método POST: Usado para enviar dados a um servidor para criar um novo recurso.

URL <https://jsonplaceholder.typicode.com/posts>: Note que aqui não usamos o ID, pois estamos criando um recurso novo, não acessando um existente.

Body (corpo da requisição): O body carrega os dados que você quer enviar para o servidor. Selecionar raw e JSON informa ao Postman que você está enviando dados brutos no formato JSON.

O Postman envia essa requisição com o body para o servidor.

Postman



16º Passo: Integração da API publica

<https://jsonplaceholder.typicode.com/posts>

My Workspace

Teste de Integração / criar novo post

POST `https://jsonplaceholder.typicode.com/posts`

Params Authorization Headers (7) **Body** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

1 Ctrl+Alt+P for Postbot

Response History

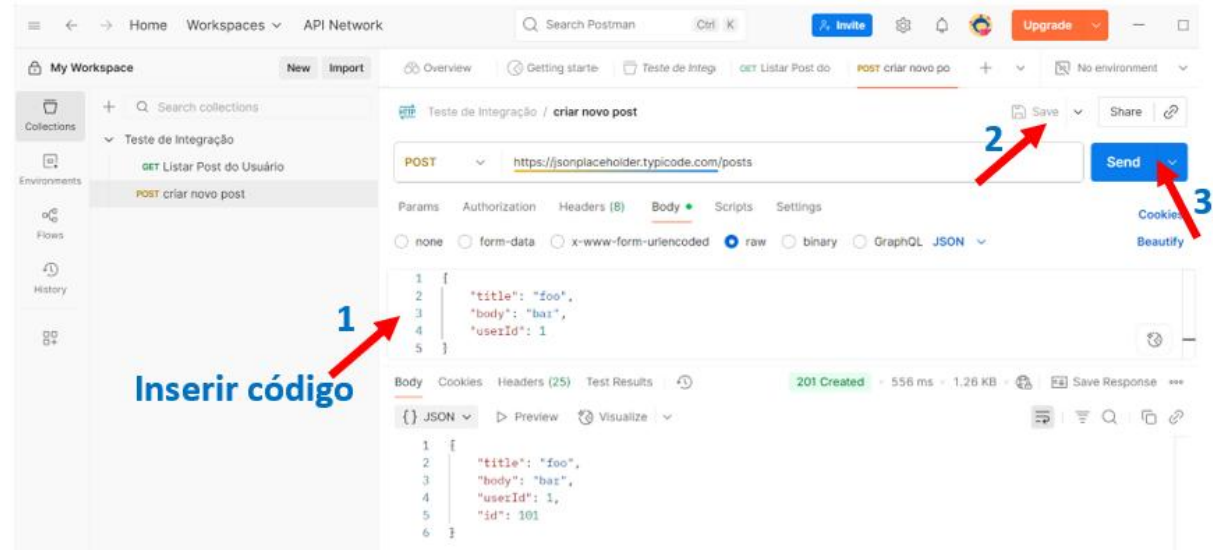
Click Send to get a response

Postman



17º Passo: código

```
{  "title": "foo",  
  "body": "bar",  
  "userId": 1}
```



Inserir código

Postman

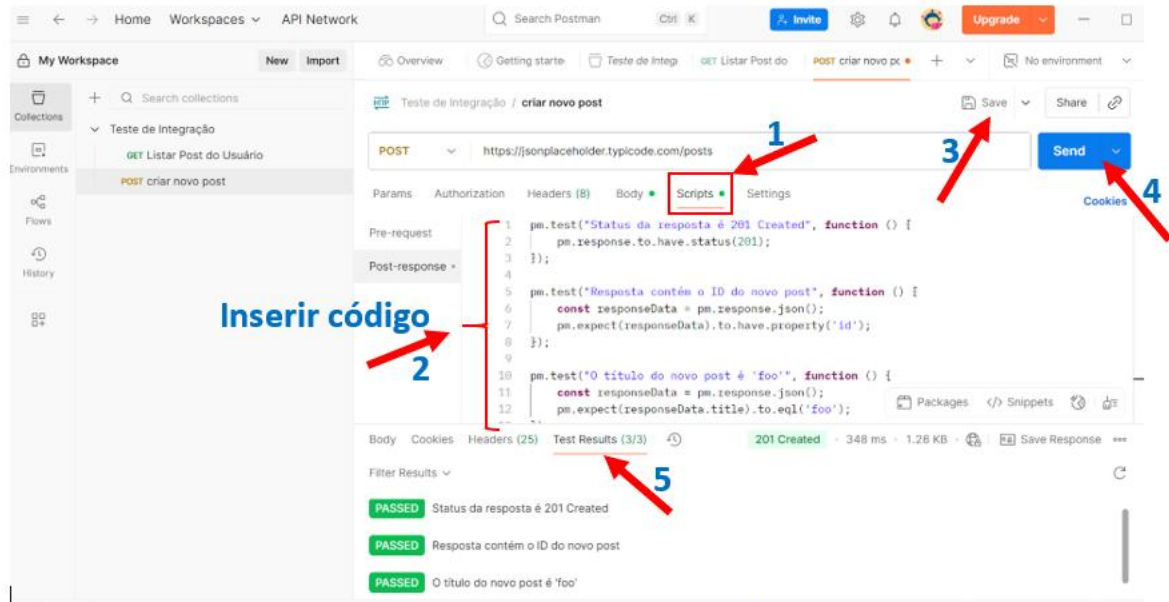


18º Passo: Teste à Requisição POST

```
pm.test("Status da resposta é  
201 Created", function () {  
  pm.response.to.have.status(2  
01);});
```

```
pm.test("Resposta contém o  
ID do novo post", function () {  
  const responseData =  
  pm.response.json();  
  pm.expect(responseData).to.  
  have.property('id');});
```

```
pm.test("O título do novo post  
é 'foo'", function () {  const  
  responseData =  
  pm.response.json();  
  pm.expect(responseData.title  
).to.eql('foo');});
```



Postman



Teste de validação da criação do POST

- O primeiro teste verifica o código de status 201 (Created), que é a resposta padrão para a criação de um recurso.
- O segundo e terceiro testes validam se a resposta contém o ID do novo post e se o título enviado está correto.

UNIVERSIDADE DO ESTADO DO PARÁ



PERFORMANCE TESTING: Apache Jmeter

Baião

Julho/2025

O que é o Apache Jmeter?



É uma ferramenta gratuita, utilizada principalmente para realizar testes de desempenho, carga e estresse em aplicações web, bancos de dados, APIs e outros serviços de rede.

Simula o comportamento de múltiplos usuários para analisar como uma aplicação se comporta sob alta demanda.

Características



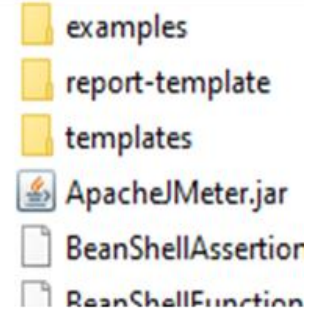
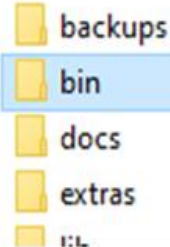
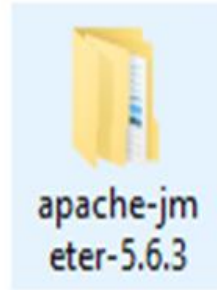
- É uma aplicação 100% Java e multiplataforma, o que permite sua execução em diferentes sistemas operacionais.
- Sendo um projeto de código aberto, a comunidade pode contribuir com melhorias, extensões (plugins) e relatórios adicionais.
- É capaz de testar diversos tipos de aplicações e serviços.



Apache Jmeter



Passo: Descompactar a pasta

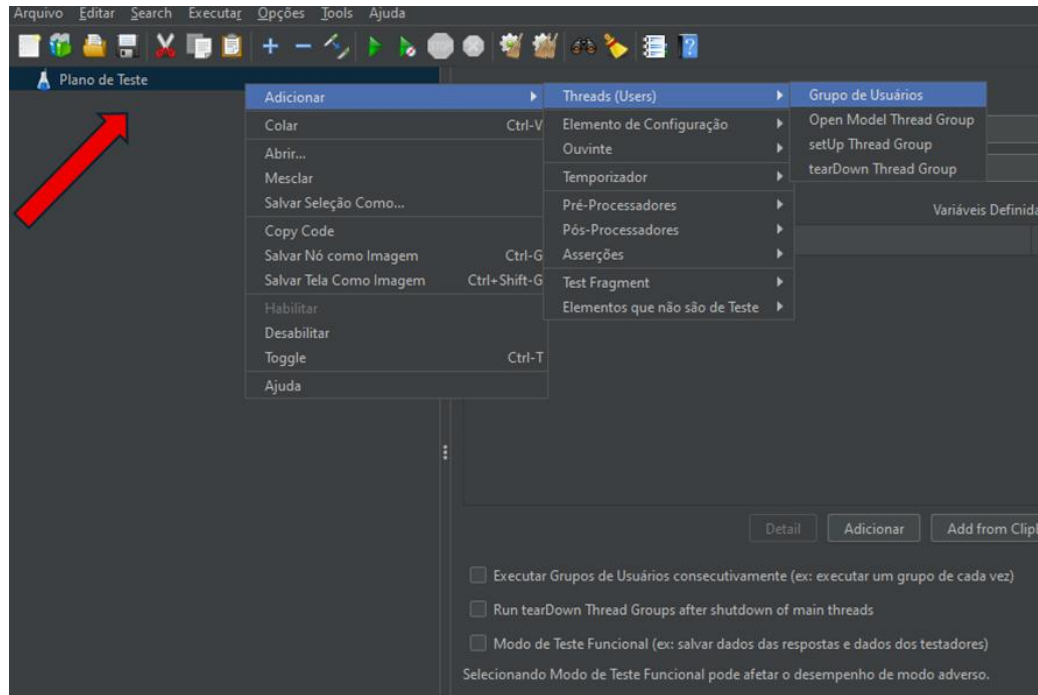


Apache Jmeter



Passo: Criar grupo de usuários

1: Clicar com botão direito em plano de teste



Apache Jmeter



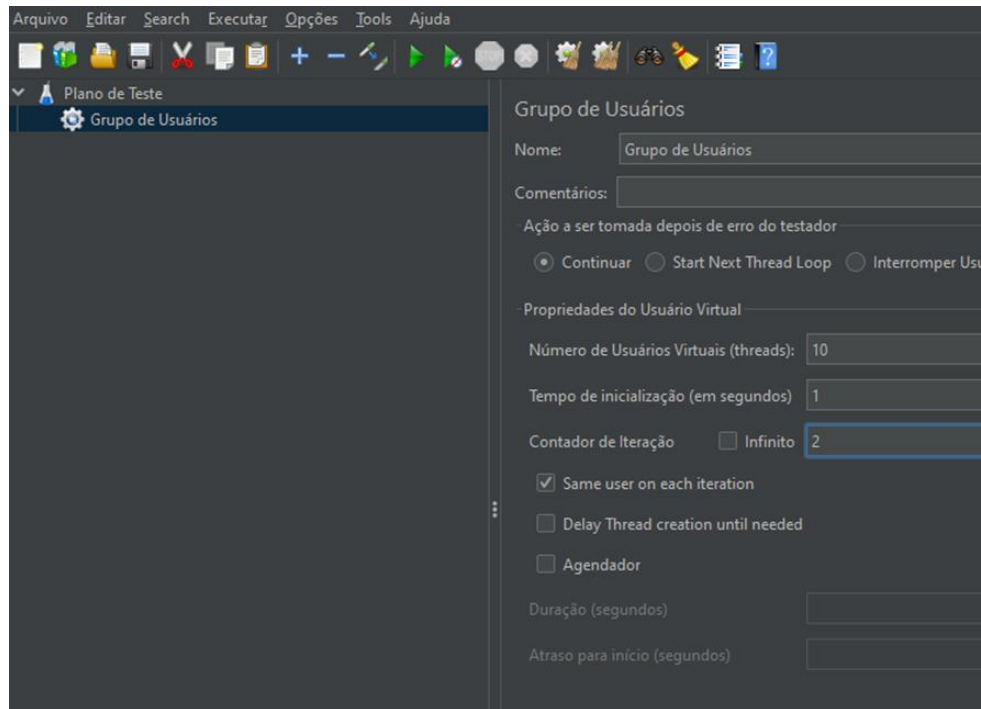
Passo: Adicionar usuários

2: Número de usuários(threads)

Tempo de inicialização = 1 segundo
o JMeter vai iniciar os 10 usuários dentro de 1 segundo.

Como são muitos usuários em pouco tempo, praticamente todos vão começar quase ao mesmo tempo.

Contador de iteração

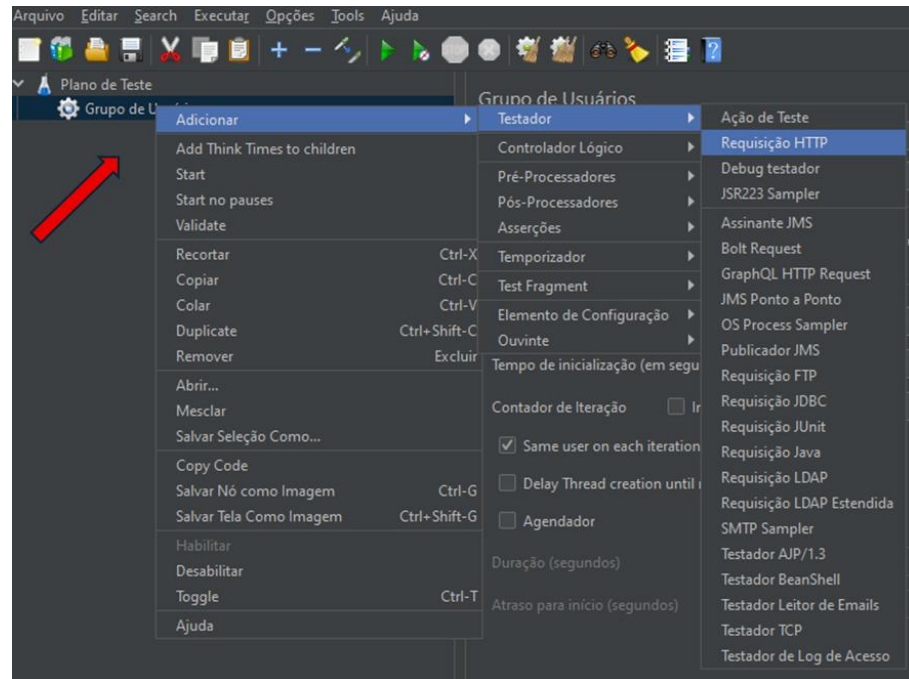


Apache Jmeter



Passo: Adicionar requisição

**3: Clicar em “Grupo de usuários
Adicionar
Testador
Requisição HTTP**



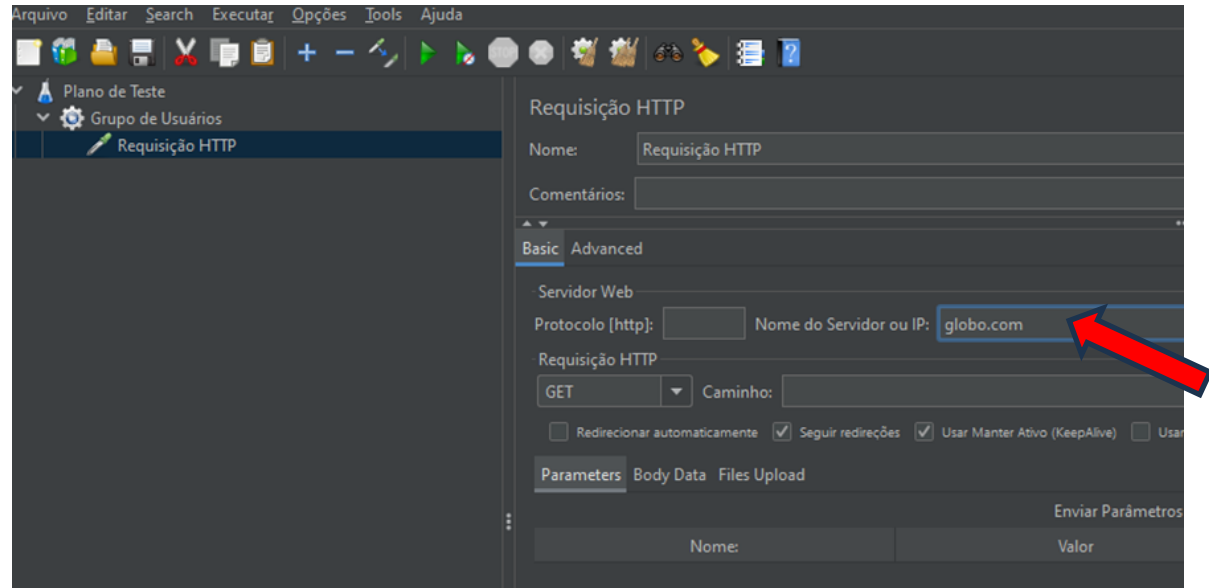
Apache Jmeter



Passo: Adicionar endereço

4: Adicionar um endereço no campo “Nome do servidor ou IP”

Exemplo: globo.com

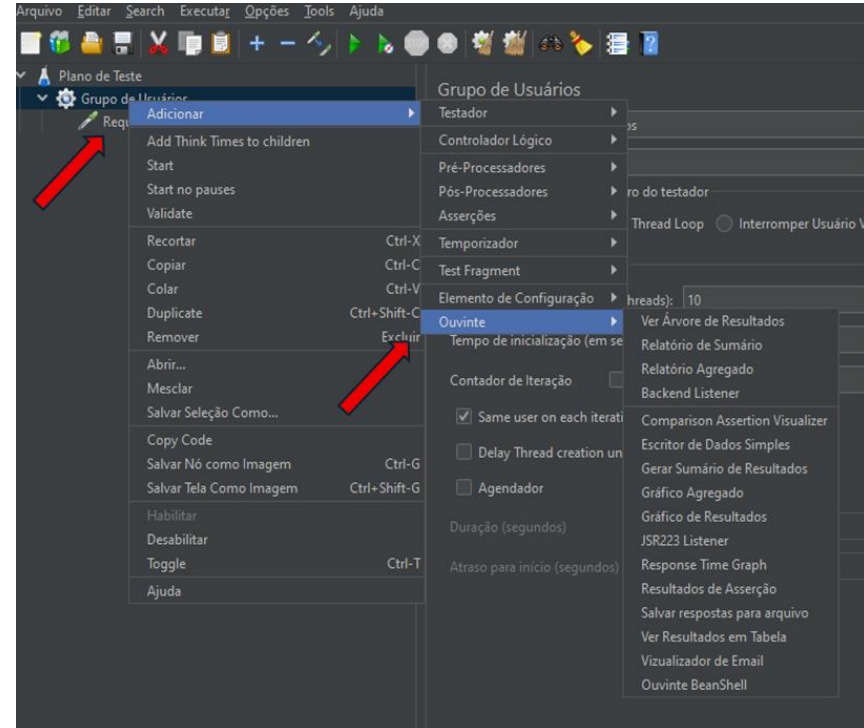


Apache Jmeter



Passo: Visualizar os resultados

5: Clicar com botão direito em grupo de usuários, em seguida adicionar, ouvinte e ver árvore de resultados.



Apache Jmeter



Passo: Resultado da requisição

Início da Amostra: 2025-08-28 21:01:07 GFT

→ Data e hora em que a requisição foi iniciada.

Tempo de Carga: 35794 ms (~35,8s)

→ Tempo total para completar a requisição, incluindo conexão, envio, resposta e fechamento.

Esse tempo está bem alto, pode indicar lentidão no servidor ou rede.

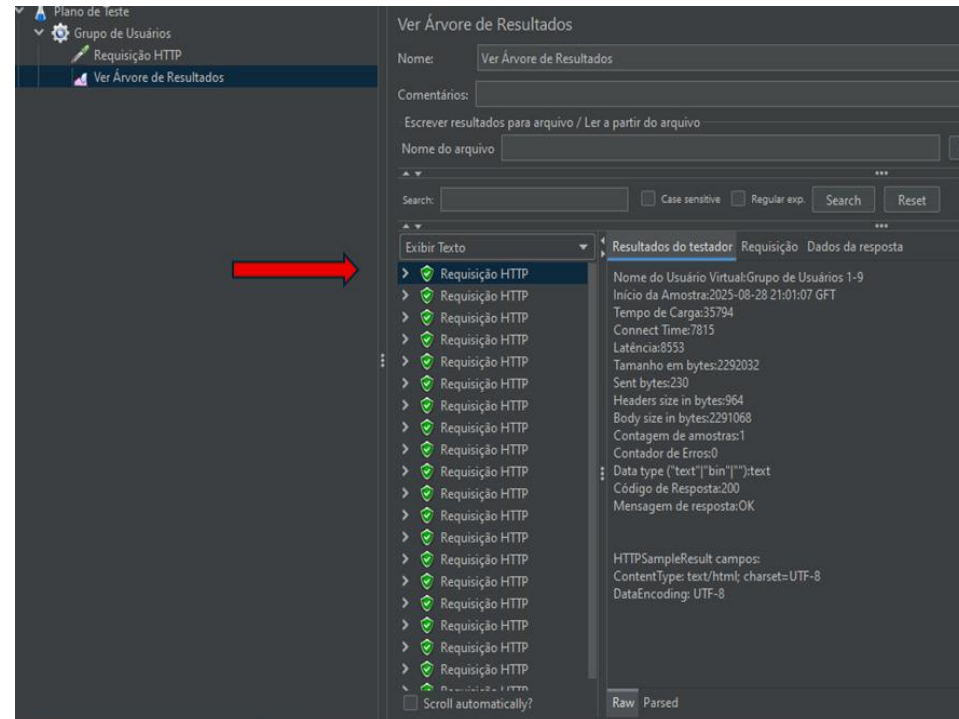
Connect Time: 7815 ms (~7,8s)

→ Tempo gasto apenas para **abrir a conexão TCP** com o servidor.

Latência: 8553 ms (~8,5s)

→ Tempo até o **primeiro byte da resposta** (server acknowledgment).

Ou seja, o servidor demorou esse tempo para começar a responder depois de receber a requisição.



Apache Jmeter



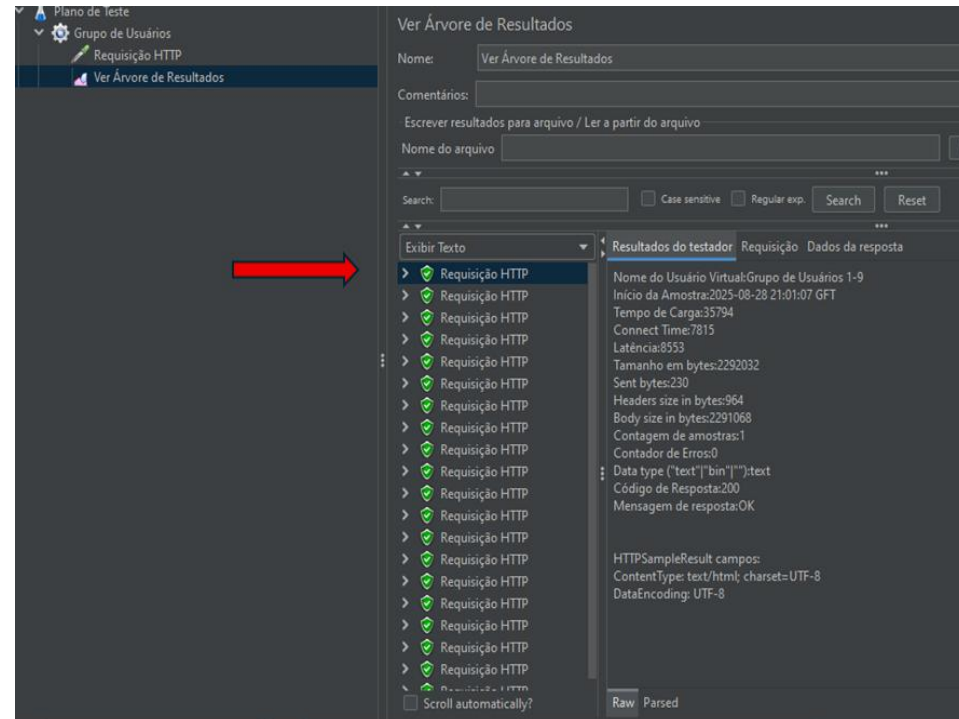
Passo: Resultado da requisição

Positivo: Não houve erro, resposta chegou corretamente .

Ponto de atenção: O tempo está **muito alto**: **Conexão (7,8s)** e **latência (8,5s)** são sinais de lentidão.

Tempo total (35s) é excessivo para a maioria dos sistemas web (o ideal é abaixo de 2-3 segundos em cenários normais).

Tamanho da resposta (2,3 MB): pode impactar performance, principalmente se muitos usuários acessarem ao mesmo tempo..



Apache Jmeter



Passo: Resultado da requisição

A requisição retornou com sucesso, porém o tempo de carga foi de 35,8 segundos, com alta latência e conexão demorada. Esse valor está acima do esperado para aplicações web responsivas e pode comprometer a experiência do usuário em cenários de carga maior.

Latência é o tempo entre o momento em que o cliente envia a requisição e o instante em que recebe o primeiro byte de resposta do servidor (ou seja, antes de baixar todo o conteúdo).

Para **APIs**, o ideal seria até **100–300 ms**.

