

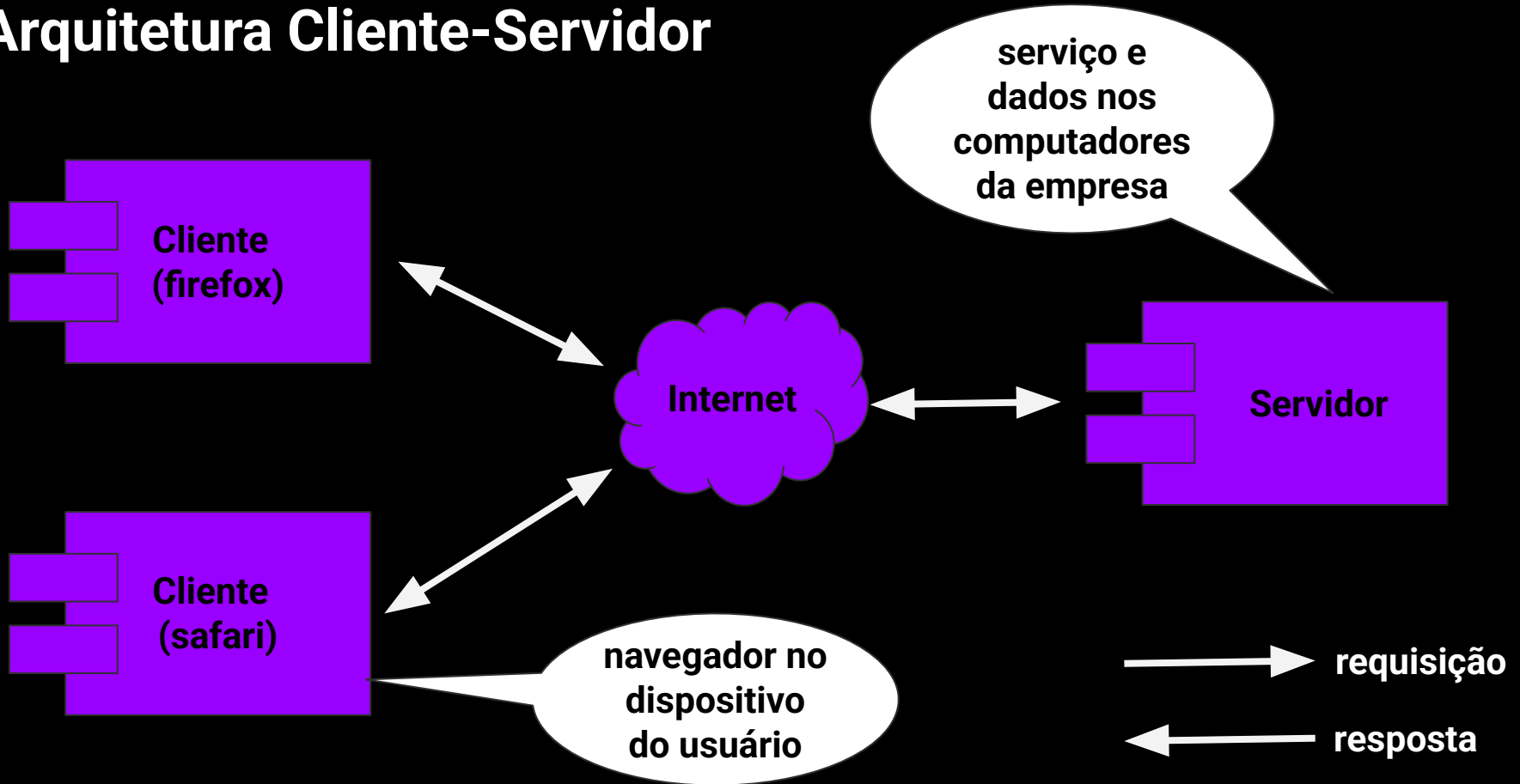
Arquitetura de Software Moderna

Matheus Barbosa
matheus.barbosa@dcx.ufpb.br

Software como Serviço (SaaS)

- Fornece software e dados como um serviço pela Internet
- Não há necessidade de instalar aplicativos nem fazer backup dos dados no dispositivo do usuário
- Mais fácil de melhorar o serviço

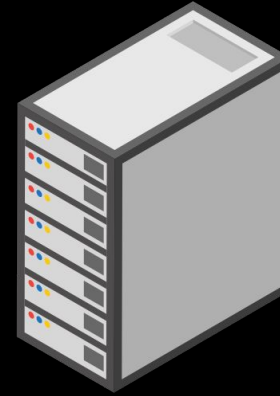
Arquitetura Cliente-Servidor



Cliente



Servidor



Requisição HTTP



Espera

Resposta HTTP



Providencia o serviço

Hypertext Transfer Protocol (HTTP)

É um protocolo de camada de aplicação para transmissão de documentos hipermídia, como o HTML.



Métodos HTTP

GET

Solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.

POST

É utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

PUT

Substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.

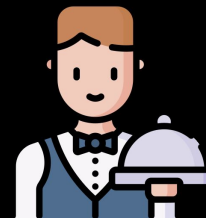
DELETE

Remove um recurso específico.

HTTP GET

PEDIR CARDÁPIO

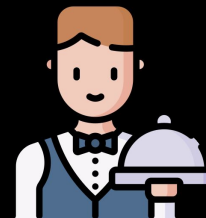
RETORNAR DADOS



HTTP POST

FAZER O PEDIDO

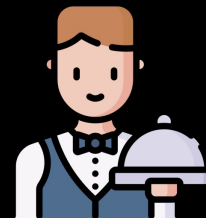
ADICIONAR DADOS



HTTP PUT

ADICIONAR UM ITEM

ALTERAR DADOS



HTTP DELETE

CANCELAR O PEDIDO

REMOVER DADOS



Códigos de status de respostas HTTP

- Respostas Informativas (100 – 199)
- Respostas bem-sucedidas (200 – 299)
- Mensagens de redirecionamento (300 – 399)
- Respostas de erro do cliente (400 – 499)
- Respostas de erro do servidor (500 – 599)

Principais status HTTP

200 OK

A solicitação foi bem-sucedida. Este é o código de status mais comum e indica que o servidor processou a requisição e retornou os dados esperados.

400 Bad Request

O servidor não pode ou não processará a solicitação devido a algo percebido como um erro do cliente

Principais status HTTP

401 Unauthorized

Significa "não autenticado". Ou seja, o cliente deve se autenticar para obter a resposta solicitada.

403 Forbidden

O cliente não tem direitos de acesso ao conteúdo; ou seja, não é autorizado, portanto o servidor está se recusando a fornecer o recurso solicitado. Ao contrário do 401 Unauthorized, a identidade do cliente é conhecida pelo servidor.

Principais status HTTP

404 Not Found

O servidor não pode encontrar o recurso solicitado.

500 Internal Server Error

O servidor encontrou uma situação com a qual não sabe lidar.

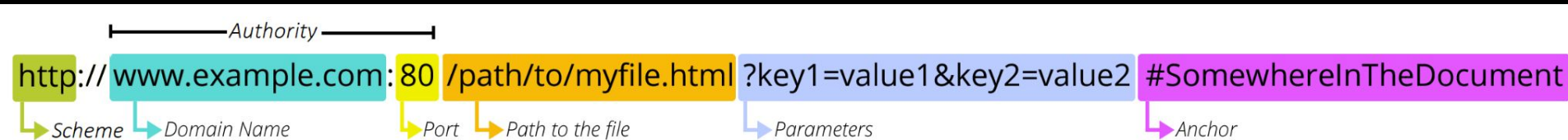
Cabeçalhos HTTP

Os cabeçalhos HTTP permitem que o cliente e o servidor passem informações adicionais com a solicitação ou a resposta HTTP.

- Informações do tipo de conteúdo (JSON, XML, HTML...)
- Informações de autenticação (Basic, Bearer Token...)

Anatomia de uma URL

Uma URL é composta por diferentes partes, algumas obrigatórias e outras opcionais. As partes mais importantes estão destacadas na URL abaixo



API REST

API REST (Representational State Transfer) é um estilo arquitetural para a construção de serviços web que utilizam o protocolo HTTP para interações entre sistemas. Em termos simples, uma API REST permite que diferentes aplicações se comuniquem entre si, utilizando os métodos HTTP para acessar e manipular recursos representados por URLs.

Estilos de formatação

```
{
  "login": "barbosamaatheus",
  "id": 18462250,
  "node_id": "MDQ6VXNlcjE4NDYyMjUw",
  "avatar_url": "https://avatars.githubusercontent.com/u/18462250?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/barbosamaatheus",
  "html_url": "https://github.com/barbosamaatheus",
  "followers_url": "https://api.github.com/users/barbosamaatheus/followers",
  "following_url": "https://api.github.com/users/barbosamaatheus/following{/other_user}",
  "gists_url": "https://api.github.com/users/barbosamaatheus/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/barbosamaatheus/starred{/owner}/{/repo}",
  "subscriptions_url": "https://api.github.com/users/barbosamaatheus/subscriptions",
  "organizations_url": "https://api.github.com/users/barbosamaatheus/orgs",
  "repos_url": "https://api.github.com/users/barbosamaatheus/repos",
  "events_url": "https://api.github.com/users/barbosamaatheus/events{/privacy}",
  "received_events_url": "https://api.github.com/users/barbosamaatheus/received_events",
  "type": "User",
  "user_view_type": "public",
  "site_admin": false,
  "name": "Matheus Barbosa",
  "company": "UFPE",
  "blog": "https://barbosamaatheus.github.io/",
  "location": "João Pessoa - PB",
  "email": null,
  "hireable": true,
  "bio": "Professor at DCX UFPE || Phd student in Software Engineering and Programming language at UFPE || Passionate about Development",
  "twitter_username": "barbosamaatheus",
  "public_repos": 45,
  "public_gists": 15,
  "followers": 64,
  "following": 10,
  "created_at": "2016-04-14T11:23:22Z",
  "updated_at": "2025-07-29T17:29:24Z"
}
```

DevTools is now available in Portuguese

Don't show again Always match Chrome's language Switch DevTools to Portuguese

Elements Console Sources Network Performance Memory >>

⌵ Filter Invert More filters

All Fetch/XHR Doc CSS JS Font Img Media Manifest Socket Wasm Other

50 ms 100 ms 150 ms 200 ms 250 ms 300 ms 350 ms 400 ms

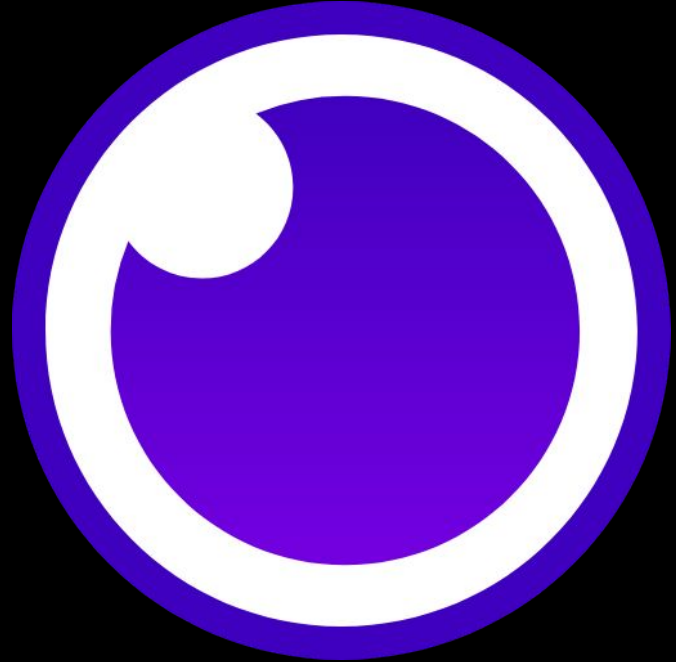
Name	Headers	Preview	Response	Initiator	Timing
barbosamaatheus	General				
content.js	Request URL		https://api.github.com/users/barbosamaatheus		
executor.js	Request Method		GET		
data:image/svg+xml;...	Status Code		200 OK (from disk cache)		
data:image/svg+xml;...	Remote Address		20.201.28.148:443		
data:image/svg+xml;...	Referrer Policy		strict-origin-when-cross-origin		
	Response Headers				
	Accept-Ranges		bytes		
	Access-Control-Allow-Origin		*		
	Access-Control-Expose-Headers		ETag, Link, Location, Retry-After, X-GitHub-OTP, X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Used, X-RateLimit-Resource, X-RateLimit-Reset, X-OAuth-Scopes, X-Accepted-OAuth-Scopes, X-Poll-Interval, X-GitHub-Media-Type, X-GitHub-SSO, X-GitHub-Request-Id, Deprecation, Sunset		
	Cache-Control		public, max-age=60, s-maxage=60		
	Content-Encoding		gzip		
	Content-Length		625		
	Content-Security-Policy		default-src 'none'		
	Content-Type		application/json; charset=utf-8		
	Date		Thu, 07 Aug 2025 03:24:52 GMT		

6 requests 1.6 MB transferred

Ferramentas de teste de API



<https://www.postman.com/>



<https://insomnia.rest/>

Personal Workspace

NewImport

Collections

Search collections

academic-stack-spark

github

GET https://api.github.com/users/barbo...

Environments

Flows

History

GET https://api.github.com/uacademic-stack-spark

github / https://api.github.com/users/barbosamaatheus

SaveShare

GEThttps://api.github.com/users/barbosamaatheus

Send

ParamsAuthorizationHeaders (6)BodyScriptsSettings

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

BodyCookiesHeaders (26)Test Results

200 OK363 ms1.87 KBSave Response

JSONPreviewVisualize

```
1 {
2   "login": "barbosamaatheus",
3   "id": 18462250,
4   "node_id": "MDQ6VXNlcjE4NDYyMjUw",
5   "avatar_url": "https://avatars.githubusercontent.com/u/18462250?v=4",
6   "gravatar_id": "",
7   "url": "https://api.github.com/users/barbosamaatheus",
8   "html_url": "https://github.com/barbosamaatheus",
9   "followers_url": "https://api.github.com/users/barbosamaatheus/followers",
10  "following_url": "https://api.github.com/users/barbosamaatheus/following{/other_user}",
11  "gists_url": "https://api.github.com/users/barbosamaatheus/gists{/gist_id}"
```

OnlineFind and replaceConsole

All LogsClear

GET https://api.github.com/users/barbosamaatheus

NetworkRequest HeadersResponse HeadersResponse Body

200363 msShow raw log

PostbotRunnerStart ProxyCookiesVaultTrash

Monolitos vs. Microsserviços

Monolito

Aplicação única, grande e coesa, onde todas as funcionalidades residem em um único bloco de código.

Vantagens: Desenvolvimento inicial mais simples e fácil depuração.

Desafios: Complexidade crescente, dificultando a manutenção, escalabilidade e o deploy (ex: build de 30+ minutos, deploy de 6+ horas em sistemas legados).

Microsserviços

Coleção de pequenos serviços, cada um independente e focado em uma funcionalidade específica.

Inspiração: Modelos adotados por empresas como Amazon (2002) e Netflix (2009).

Benefícios: Permite equipes menores e independentes, agilidade e resiliência.

Desafios: Complexidade em orquestrar todos os microsserviços.