

Conceitos de API

primeiro tomo.



Prólogo

Thomás Daniel Vieira

Engenheiro de Software

linkedin.com/in/thomasvieira/

Autor do livro *Robot Framework: automação versátil e consistente para testes* (Casa do Código, 2023).



Cronograma de hoje

1. O que é uma API?
2. Modelo cliente – servidor
3. Arquitetura orientada à serviços
4. O protocolo HTTP
5. Modelos de arquitetura HTTP: REST e SOAP
6. URL
7. Métodos
8. Códigos de status de resposta



O que é uma API?

Do inglês *application programming interface*, é um conjunto de recursos disponibilizados de forma padronizada para **integração** entre aplicações ou componentes em um sistema.

Existem diferentes tipos de API's: serviços, filas, etc.



a ver um exemplo prático...

[ver instruções.](#)



Preparando o ambiente de desenvolvimento e estudo

Pré-requisitos: docker ou nodejs.

Instalação do NodeJS: <https://www.alura.com.br/artigos/como-instalar-node-js-windows-linux-macos/>.

A API de demonstração está em <https://github.com/thvieira/bookstore-api-demo.git>.

01. Baixar o projeto:

```
$ git clone https://github.com/thvieira/bookstore-api-demo.git
```

02. Entrar no diretório do projeto:

```
$ cd api-alurabooks
```

03. Instalar as dependências do projeto:

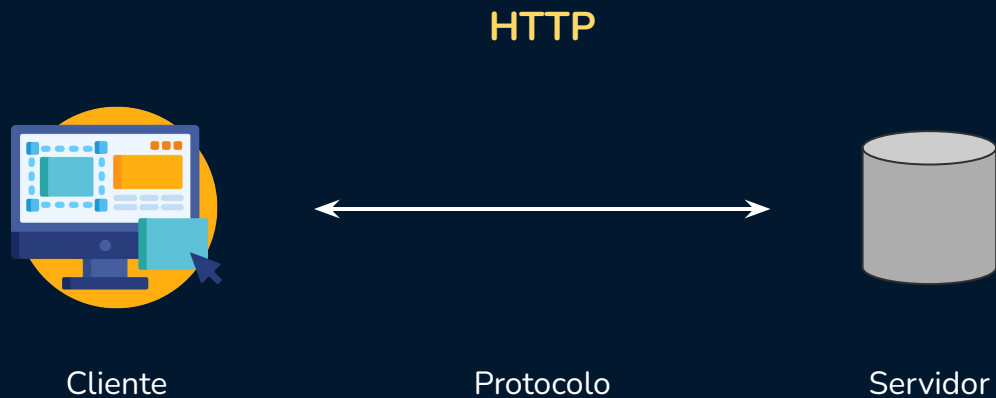
```
$ npm install
```

04. Executar o programa e disponibilizar a API em http://localhost:8000:

```
$ npm run app
```

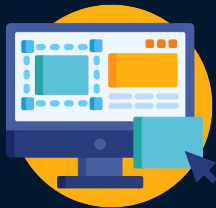


Modelo cliente – servidor



Arquitetura orientada à serviços

Navegador, aplicativo
mobile, caixa eletrônico...



Cliente

HTTP



Protocolo

front-end



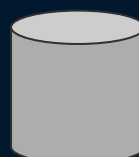
Servidor

HTTP



Protocolo

back-end



Servidor



O protocolo HTTP

Hypertext Transfer Protocol é um protocolo da camada de aplicação.

Protocolo é um conjunto de regras e definições para comunicação entre duas entidades.

Quem inicia a comunicação é sempre o cliente.



Troca de mensagens



Cliente

e aí, bora se comunicar?

oi, bora!

GET /times

Ok

Internacional, Grêmio, Guarany...



Servidor



Troca de mensagens



Cliente

e aí, bora se comunicar?

oi, bora!

GET /times

404 Not found



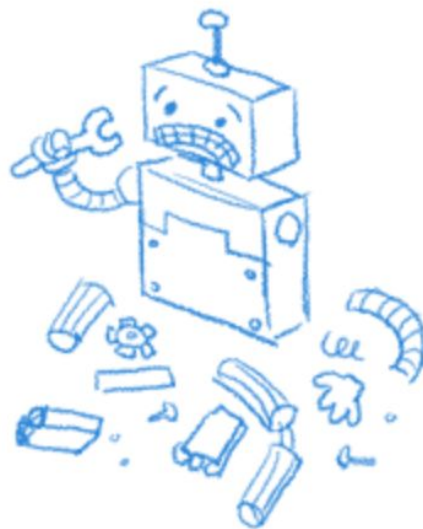
Servidor





404. That's an error.

References not found. That's all we know.



Troca de mensagens



Cliente

e aí, bora se comunicar?

oi, bora!

GET /times

503 Serviço indisponível



Servidor

Estrutura da mensagem HTTP

Request Line

Composto pelo método, URI e a versão do HTTP.

```
GET /index.html HTTP/1.1  
Host: exemplo.com  
Accept-Language:pt
```

Header

Cabeçalhos com informações complementares.

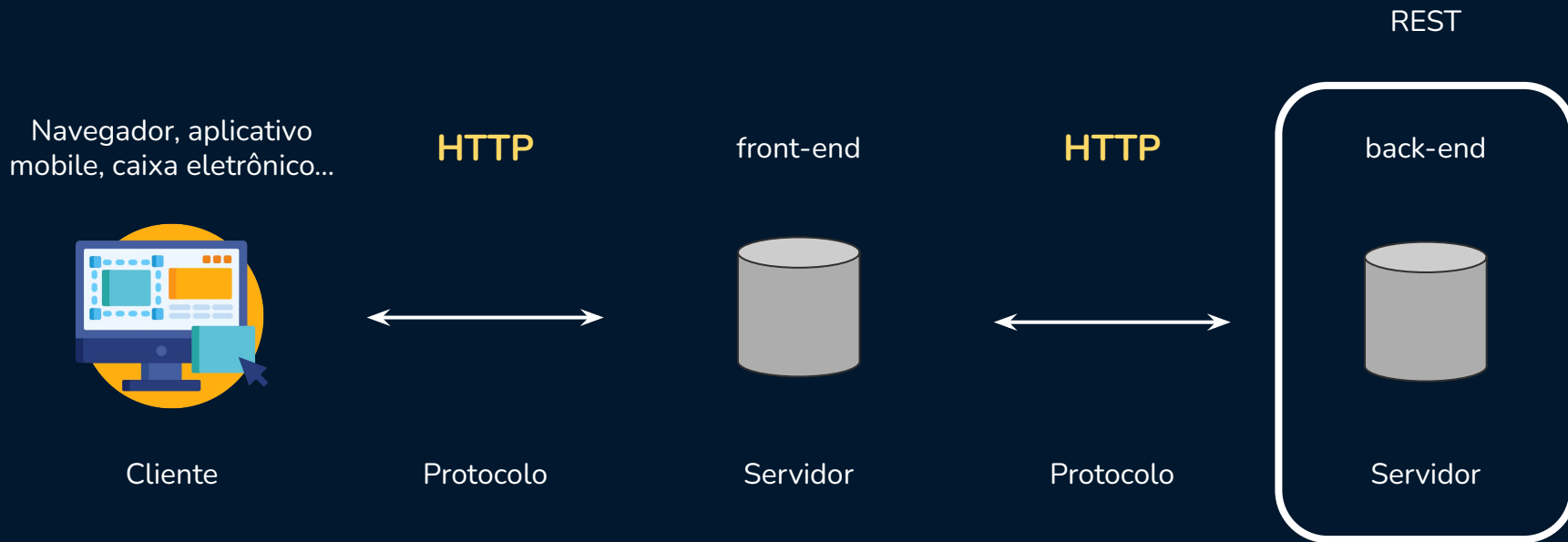
```
HTTP/1.1 200 OK  
Data: Sat. 09 Oct 2020 14:28:02  
GMT  
Last-Modified: Tue, 01 Nov 2019  
20:18:22 GMT  
Accept-Ranges: bytes  
Content-Length: 29769  
Content-Type: text/html  
<DOCTYPE html...
```

Body

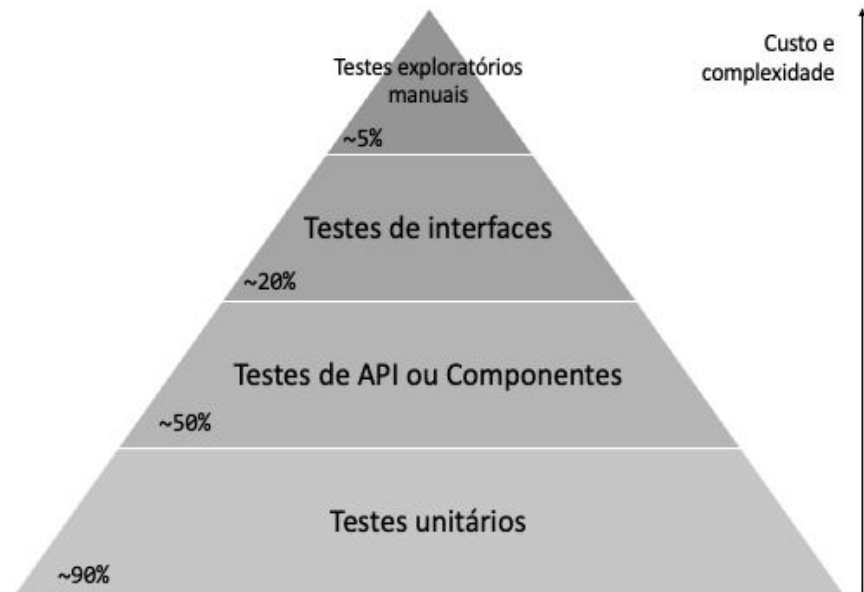
O corpo de uma resposta contém os dados em si.



Arquitetura orientada à serviços



Estratégia de testes



Modelos de arquitetura HTTP

REST

Representational State Transfer

JSON, texto puro, outros

É um conjunto de padrões

Não mantém informações sobre o estado da comunicação

Mais flexível

Expõe dados

Utiliza sempre HTTPS como protocolo

Possibilidade de armazenamento em cache

SOAP

Simple Object Access Protocol

XML

É um protocolo

Mantém informações sobre o estado da comunicação

Mais rígido

Expõe operações

Funciona com qualquer protocolo

Costuma ser mais lento pois as mensagens são maiores



Referências:

Amazon AWS: <https://aws.amazon.com/pt/compare/the-difference-between-soap-rest/>

RedHat: <https://www.redhat.com/pt-br/topics/integration/whats-the-difference-between-soap-rest>

URL

Uniform Resource Locator, ou seja, “Localizador de Recursos Universais”.

<http://localhost:8080>, <https://www.google.com>, <https://db.tec.br>, etc.



URL

Localizador?

É como um endereço através do qual é possível acessar os recursos.



URL

Recursos?

Pode ser uma página HTML, um arquivo de imagem, um arquivo de áudio, dentre outros conteúdos digitais.



URL

Universal?

Toda a rede mundial de computadores utiliza este padrão.



URL

http://localhost:3000/

protocolo://domínio:porta/caminho/recurso?query_string#fragmento



HTTP,
HTTPS...



URL

http://localhost:3000/

protocolo://domínio:porta/caminho/recurso?query_string#fragmento



**nome ou endereço de um
servidor que disponibilizará
os recursos.**

**Pode ser até mesmo um
endereço IP.**



URL

http://localhost:3000/

protocolo://domínio:porta/caminho/recurso?query_string#fragmento



um servidor possui várias
portas para conexão.

O padrão do HTTP é a
porta 80 ou a porta 8080.

No caso do HTTPS o padrão
é a porta 443.



URL

http://localhost:3000/

protocolo://domínio:porta/caminho/recurso?query_string#fragmento



**caminho até o recurso
desejado.**



URL

http://localhost:3000/

protocolo://domínio:porta/caminho/recurso?query_string#fragmento



nome do recurso desejado.



URL

`http://localhost:3000/`

`protocolo://domínio:porta/caminho/recurso?query_string#fragmento`



**Lista de parâmetros da
requisição, que devem
seguir o padrão de
chave=valor, chave=valor, ...**



URL

`http://localhost:3000/`

`protocolo://domínio:porta/caminho/recurso?query_string#fragmento`



**Parte ou posição específica
dentro do recurso.**



Postman ou Insomnia

Explorando os recursos de uma API.

postman.com/downloads/

insomnia.rest/download/



Métodos

GET	Acessar dados (<i>read</i>)
POST	Criar dados (<i>create</i>)
PUT	Substituir dados (<i>update</i>)
PATCH	Alterar dados (<i>update</i>)
DELETE	Apagar dados (<i>delete</i>)



Referências:

RFC 7231 – Method definitions: <https://www.rfc-editor.org/rfc/rfc7231#section-4.3>.

Códigos de status de resposta

100	Informacional
200	Sucesso
300	Redirecionamento
400	Erro do cliente
500	Erro do servidor



Referências:

RFC 7231 – Response Status Codes: <https://www.rfc-editor.org/rfc/rfc7231#section-6>.

Códigos de Status HTTP: <https://www.httpstatus.com.br/>.

Perguntas? Observações?
Dúvidas? Comentários?



Exercícios

Utilize o Postman para realizar requisições para:

1. Listar todos os livros;
2. Listar todos os livros com filtro de busca por autor;
3. Ler informações de um livro específico;
4. Criar um novo livro;
5. Alterar informações de um livro;
6. Remover um livro;



Referências

The Practical Test Pyramid:

<https://martinfowler.com/articles/practical-test-pyramid.html>

RFC 7231: <https://www.rfc-editor.org/rfc/rfc7231>.

VIEIRA, Thomás. **Robot Framework**: Automação versátil e consistente para testes. Casa do Código, São Paulo, 2023.



Obrigado!

educacao@db.tec.br

