

Poc backend - parte I

Novidades

JAVA 8

- Default Methods
- Functional Interfaces

JAVA 9

- List, Set, Map
Methodos
Imutáveis: `.of()`;
- Reactive
Streams:
Publisher;

JAVA 10

- List, Set, Map:
`.copyOf()`;
- Local-Variable;

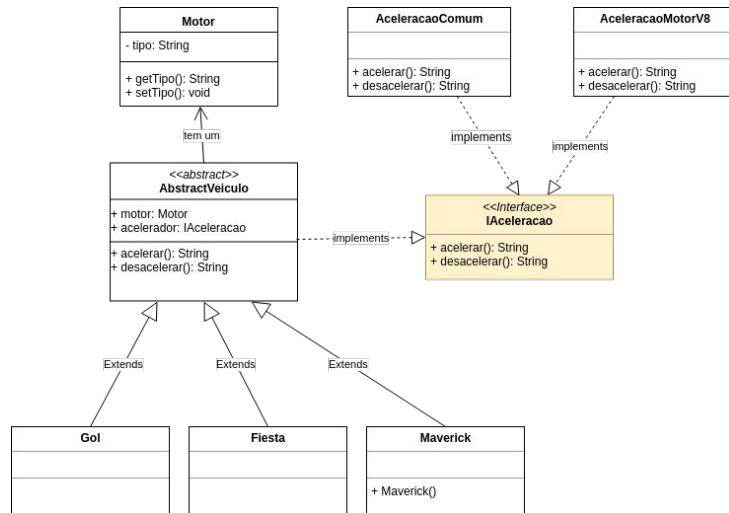
JAVA 11

- Local-Variable
para parâmetros
lambda;
- `readString()/writeString()` de/para
arquivos;

Novidades java 8

DEFAULT METHOD

- Adiciona novos métodos sem quebrar o código existente que implementa a interface.



Novidades java 8

FUNCTIONS

→ `java.util.function ;`

Functional Interfaces	Resulta
Predicate	recebe um valor e retorna um booleano
Function	se precisarmos receber algo como argumento e produzir um resultado.
Consumer	receber um argumento sem a necessidade de retornarmos nada

Novidades java 9

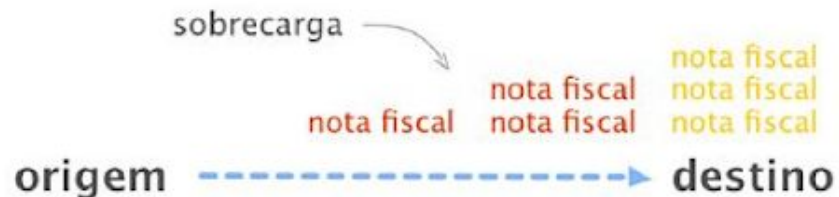
MÉTODOS IMUTÁVEIS: .of()

- As interfaces List e Set possuem métodos “of ()” para criar uma lista imutável vazia ou não;

Novidades java 9

SUBMISSION PUBLISHER

→ `java.util.concurrent.*`



Novidades java 10

MÉTODO ESTÁTICO COPYOF();

- Mudanças na API collection;
- Retorna uma List, Map e Set imutáveis;
- List não recebe modificações subsequentes;

Novidades java 10

LOCAL-VARIABLE TYPE INFERENCE;

- Variável local inicializada;
- Índice de loops;
- Local em loops;

Novidades java 11

LOCAL-VARIABLE PARA
PARÂMETROS LAMBDA;

- var usado para declarar os parâmetros formais implicitamente tipada;
- Porque usar: aplicar notações (@Nullable);

`(var s1, s2) -> s1 + s2` //não é permitido pular

`(var s1, String y) -> s1 + y` //não é permitido “mistura”

`var s1 -> s1` //precisa de “()” se usar var

Novidades java 11

READING/WRITING STRING DE/PARA ARQUIVOS;

- `java.nio.file.Files`;
- decodificando de bytes para caracteres;
- não utilizar para leitura de arquivos muito grandes (2GB);

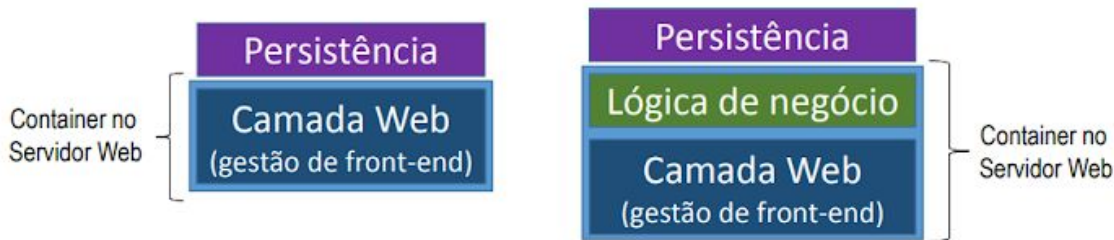
`readString(path, StandardCharsets.UTF_8)`

SOA & Microservice

SOA: É um tipo de arquitetura que usa serviços como blocos de construção de maneira a facilitar a integração empresarial e o reuso de componentes através de acoplamento fraco.
[JOSUTTIS 2007]

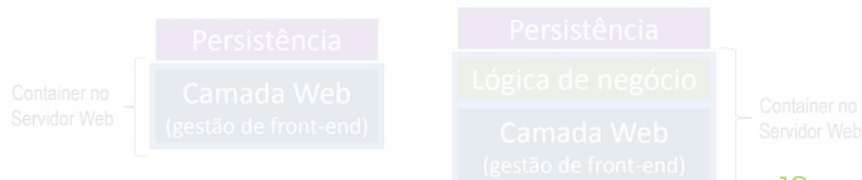
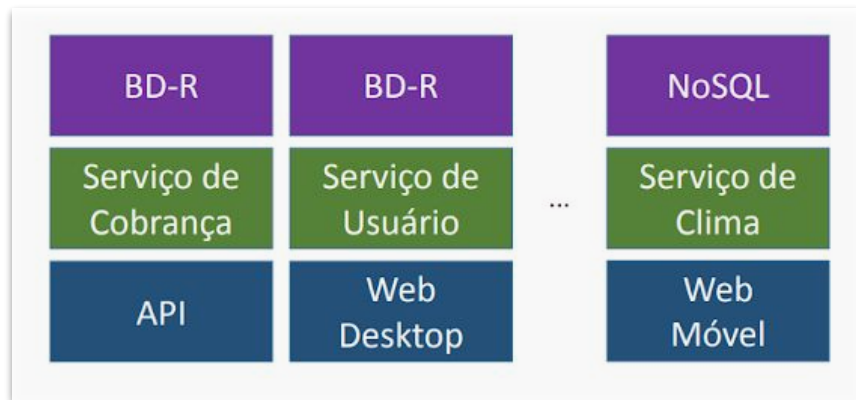
Motivação

- Arquitetura tradicional: n camadas monolíticas
- Camada grande.
- Elasticidade/escalabilidade complexa
- Implantação e manutenção dificultadas.
- Sistemas online - transações instantaneas.



SOA Arquitetura

- Diferentes camadas “paralelas” para o mesmo nível.
- Múltiplos serviços.
- Lógica de negócio e camadas específicas.
- SOA não é só serviços, é também uma arquitetura de integração (e envolve expor a lógica de negócios)!



SOA princípios

Acoplamento Fraco

mudanças não afetam outros componentes relacionados, falhas isoladas

Capacidade de descoberta

O contrato de um serviço devem ser identificados e vinculados durante a execução.

Interoperabilidade

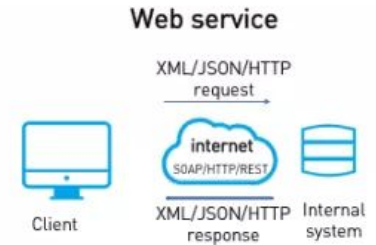
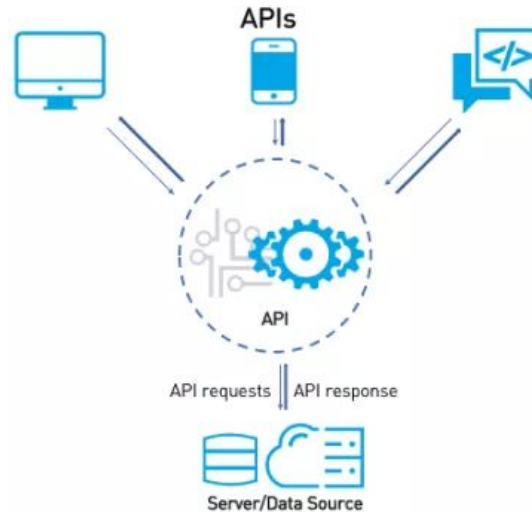
troca de mensagens deve ser independente de plataforma

Reuso

Mais de uma aplicação pode ser cliente de um serviço.

Serviço

- Operações que realizam funções de negócio.
- Disponível em uma rede.
- Possibilidades: Web Service, Serviço REST, Microserviços.
- Web Service REST: expõem uma API para manipular operações de CRUD em dados, baseado no protocolo HTTP padrão.



Serviço: tipos

- De Entidade.
- Funcional.
- De Processo.

Microserviços

Padrão arquitetural

Para desenvolvimento de aplicações server-side

.

Independentes

Desenvolvido e instanciados de maneira independentes um do outro.

Conjunto Serviços

Aplicações decompostas que se colaboram, comunicam-se por protocolo

.

Persistência

Persistência própria.

Implementações

Conjunto restrito de funções.

Só lógica

Preocupação com a lógica de negócio (sem apresentações).

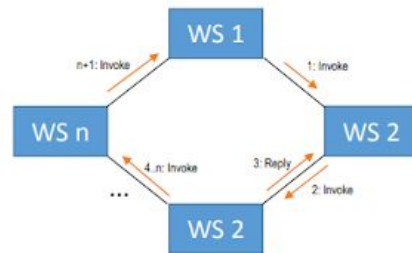
Serviço: composição

- SOA é uma ideia geral.
- Microserviços são uma maneira específica de implementá-los.
- Princípios de SOA se aplicam para Microserviços.
- Diferença implementação:
 - ◆ SOA: favorece a orquestração.
 - ◆ Micro: favorece a coreografia distribuída (REST + HTTP/S).

Orquestração



Coreografia



[JURIC 2013]

Serviço: contrato

- “contrato” entre o requisitante e o provedor deste serviço.
- Contrato especifica o serviço.

The diagram illustrates the components of a REST API URL: `http://api-int.grupodimedservices.com.br/tst/item/v3/itens/base/autocomplete?nome=parace&codigofilial=101&maxResult=200&ordenarRentabilidade=true&ordenarPreco=false`. The components are grouped as follows:

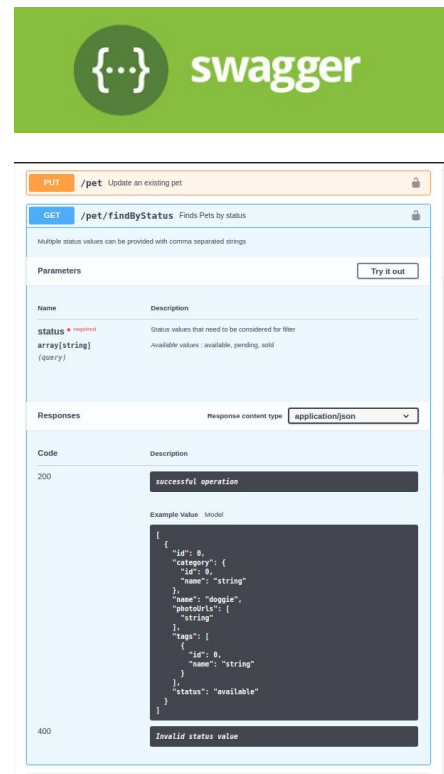
- protocolo**: `http`
- hostname**: `api-int.grupodimedservices.com.br`
- serviço**: `item`
- versão**: `v3`
- recurso a ser acessado**: `itens/base/autocomplete`

The following table shows the details of the request:

Field	Value
Request URL	<code>http://api-int.grupodimedservices.com.br/tst/item/v3/itens/base/autocomplete?nome=parace&codigofilial=101&maxResult=200&ordenarRentabilidade=true&ordenarPreco=false</code>
Request Method	<code>GET</code>
Status Code	<code>401 Unauthorized</code>
Remote Address	<code>10.11.100.200:80</code>
Referrer Policy	<code>no-referrer-when-downgrade</code>

Contrato: documentação

- Documentação da API;
- Exemplo Swagger



HTTP

Hypertext Transfer Protocol é um protocolo de camada de aplicação para transmissão de documentos hipermídia, como o HTML. Foi desenvolvido para comunicação entre navegadores web e servidores web, porém pode ser utilizado para outros propósitos também. [MDN web docs]

HTTP

- Protocolo que permite servidores e navegadores da Web troquem dados.
- Usa conexões TCP confiáveis.
- Stateless.

request

POST <http://api-int.grupodimedservices.com.br/tst/mostruario/v3/itens/detalhe>

```
JSON ▾ Basic ▾ Query
1 {
2   "filial": "101",
3   "perfil": 1,
4   "itens": [
5     {
6       "codigo": 896760,
7       "quantidade": 1
8     }
9   ],
10  "consultaRegrasFiscais": {
11    "uf": "RS",
12    "pais": "BR",
13    "ufDestino": "RS",
14    "paisDestino": "BR"
15  }
16 }
```

response

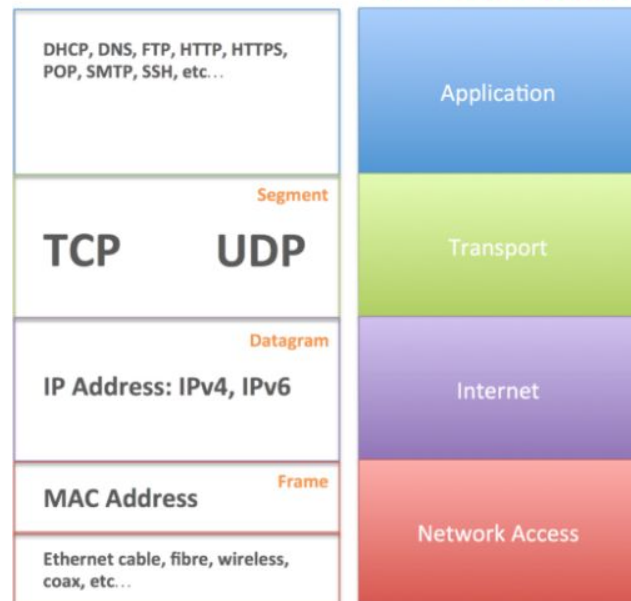
[illegible]

TCP/IP

- Transmission Control Protocol/Internet Protocol.
- Dividido em 4 camadas(layers).
- Regras e/ou protocolos entre hardwares e a internet.

- ◆ SMTP: Simple Mail Transfer Protocol
- ◆ SSH: Secure Shell
- ◆ FTP: File Transfer Protocol
- ◆ SFTP: Secure File Transfer protocol
- ◆ SCP: Secure Copy Protocol
- ◆ DHCP: Dynamic Host Configuration protocol
- ◆ HTTP: Hyper Text Transfer Protocol
- ◆ POP: Post Office Protocol (the version currently in use is 3, POP3)
- ◆ IMAP: Internet Message Access Protocol

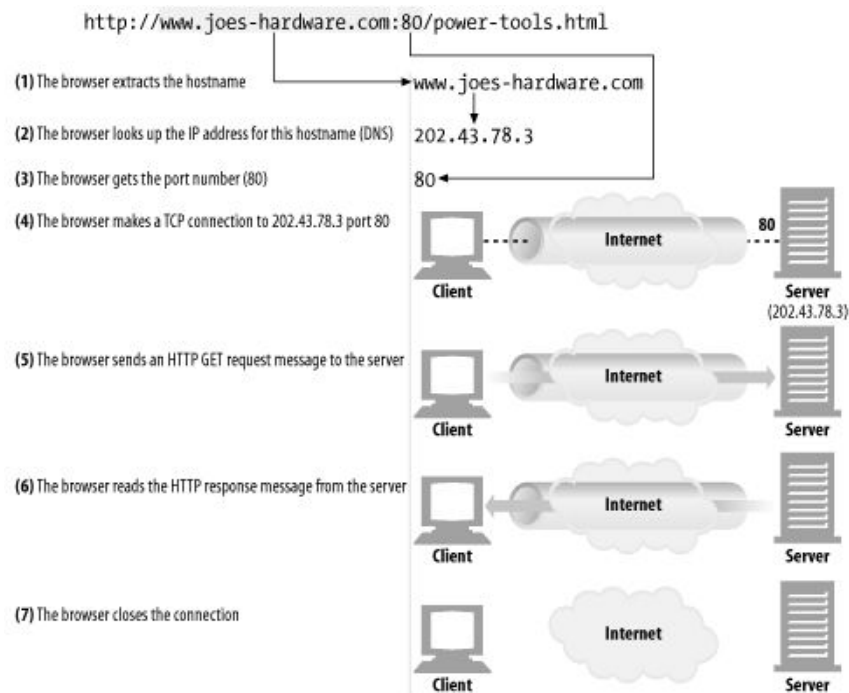
The TCP/IP Model



TCP/IP

- Transmission Control Protocol/Internet Protocol.
- Dividido em 4 camadas(layers).
- Regras e/ou protocolos entre hardwares e a internet.

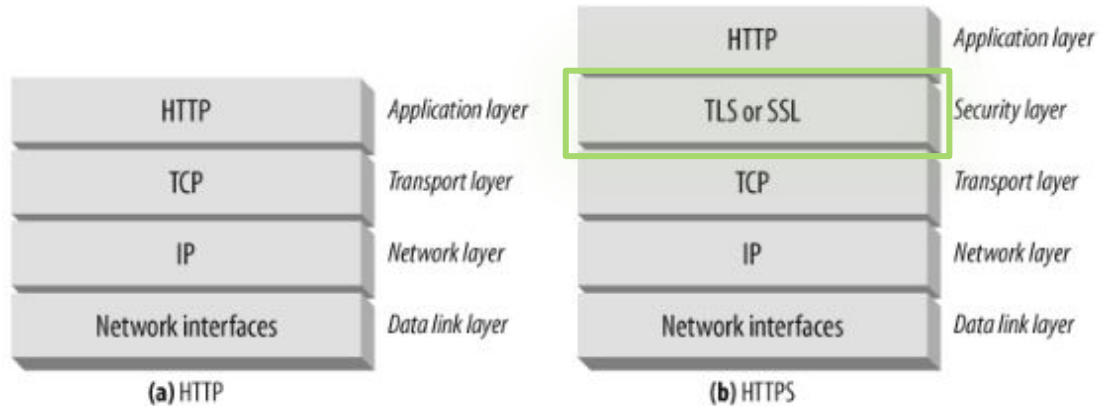
- ◆ SMTP: Simple Mail Transfer Protocol
- ◆ SSH: Secure Shell
- ◆ FTP: File Transfer Protocol
- ◆ SFTP: Secure File Transfer protocol
- ◆ SCP: Secure Copy Protocol
- ◆ DHCP: Dynamic Host Configuration protocol
- ◆ HTTP: Hyper Text Transfer Protocol
- ◆ POP: Post Office Protocol (the version currently in use is 3, POP3)
- ◆ IMAP: Internet Message Access Protocol



<https://www.oreilly.com/library/view/http-the-definitive/1565925092/ch04s01.html>

HTTP e HTTPS

- HTTP na camada superior do TCP.
- TCP na camada superior IP.
- No HTTPS, inserido em uma camada de criptografia (chamada TLS ou SSL) entre HTTP e TCP.

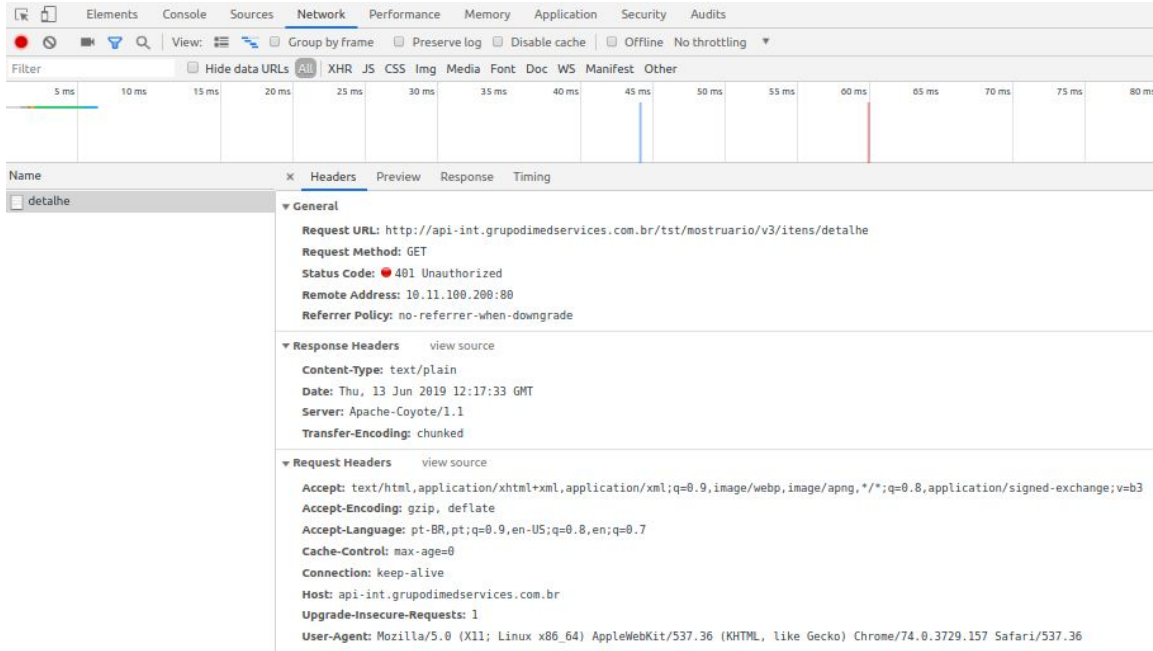


<https://www.oreilly.com/library/view/http-the-definitive/1565925092/ch04s01.html>

TLS ou SSL

- Transport Layer Security: TLS 1.3 (2018).
- Secure Sockets Layer: SSL 3.0 (1996 - depreciada).
- Subcamada no TCP/IP.
- Como trabalha:
 - ◆ Mistura ilegível de caracteres (encryption);
 - ◆ Autenticação dos dispositivos (authentication);
 - ◆ Assinatura digital (integrity)

HTTP: informação solicitação



The screenshot displays the Network tab of a web browser's developer tools. A request to `http://api-int.grupodimedsservices.com.br/tst/mostruario/v3/itens/detalhe` is selected. The request method is GET, and the status code is 401 Unauthorized. The response headers indicate the content type is text/plain, the date is Thu, 13 Jun 2019 12:17:33 GMT, and the server is Apache-Coyote/1.1. The request headers show the user agent as Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.157 Safari/537.36.

Name	Headers	Preview	Response	Timing
detalhe	<p>General</p> <p>Request URL: <code>http://api-int.grupodimedsservices.com.br/tst/mostruario/v3/itens/detalhe</code></p> <p>Request Method: GET</p> <p>Status Code: 401 Unauthorized</p> <p>Remote Address: 10.11.100.200:80</p> <p>Referrer Policy: no-referrer-when-downgrade</p> <p>Response Headers view source</p> <p>Content-Type: text/plain</p> <p>Date: Thu, 13 Jun 2019 12:17:33 GMT</p> <p>Server: Apache-Coyote/1.1</p> <p>Transfer-Encoding: chunked</p> <p>Request Headers view source</p> <p>Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3</p> <p>Accept-Encoding: gzip, deflate</p> <p>Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7</p> <p>Cache-Control: max-age=0</p> <p>Connection: keep-alive</p> <p>Host: api-int.grupodimedsservices.com.br</p> <p>Upgrade-Insecure-Requests: 1</p> <p>User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.157 Safari/537.36</p>			



HTTP: métodos

Método Http	Descrição
GET	Recuperar um recurso
POST	Criar um recurso ou para executar uma operação complexa em um recurso
PUT	Atualizar um recurso
DELETE	Excluir um recurso
PATCH	Executar uma atualização parcial em um recurso

HTTP: código de status

Alcance	Significado	Código do Status/Descrição
2xx	Execução bem sucedida	200 OK, 201 Created, 202 Accepted, 204 No Content,
4xx	Problemas com a solicitação.	400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 405 Method Not Allowed, 406 Not Acceptable, 415 Unsupported Media Type, 422 Unprocessable Entity, 429 Too Many Requests
5xx	Erro servidor.	500 Internal Server Error, 503 Service Unavailable

Fim!

