

Fundamentos de REST e Arquitetura de Microserviços

Entendendo os conceitos através de uma API
desenvolvida em Java e Spring Framework





Emerson Bezerra

Estudante do 6º período de Ciência da Computação na Unipê.
Em sua jornada acadêmica nesta instituição já passou pela
Escola de Computação Solidária (ECS) e Fabrica de Software.
Atualmente é desenvolvedor back-end na Zup Innovation.

...

Sumário

01

O que são microsserviços?

Significado, vantagens e desafios dessa arquitetura

02

O conceito de uma API REST

Permitindo a comunicação entre microsserviços de forma eficiente

03

As boas práticas de uma aplicação RESTful

Entendendo a semântica desse estilo arquitetural

04

Alguns exemplos práticos

Um caso hipotético de aplicação usando Java e Spring





01

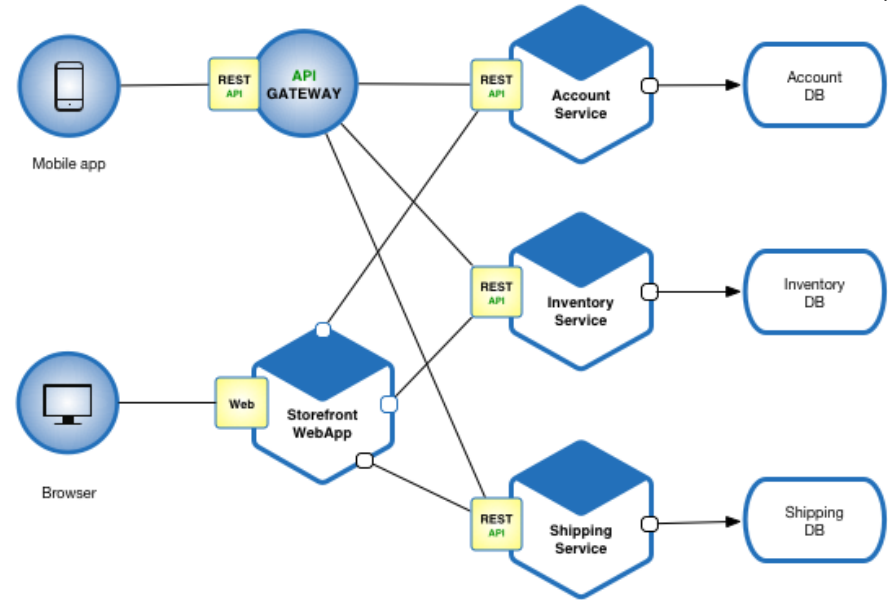
O que são microserviços?



Definição

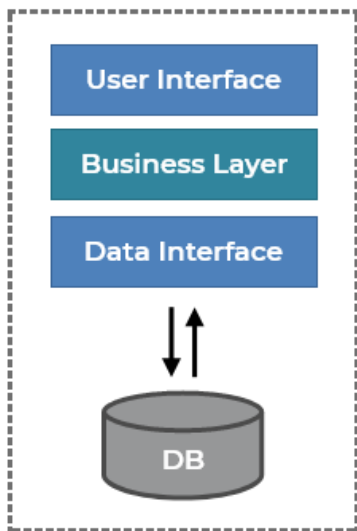
É um estilo arquitetural que estrutura uma aplicação como uma coleção de serviços que são:

- Altamente manutenível e testável
- Fracamente acoplado
- Independentemente implantável
- Organizado em torno de recursos de negócios
- Gerenciado por uma pequena equipe



Contraste com uma aplicação monolítica

Monolithic Architecture



Monolitos ainda estão presentes em vários 'cases'

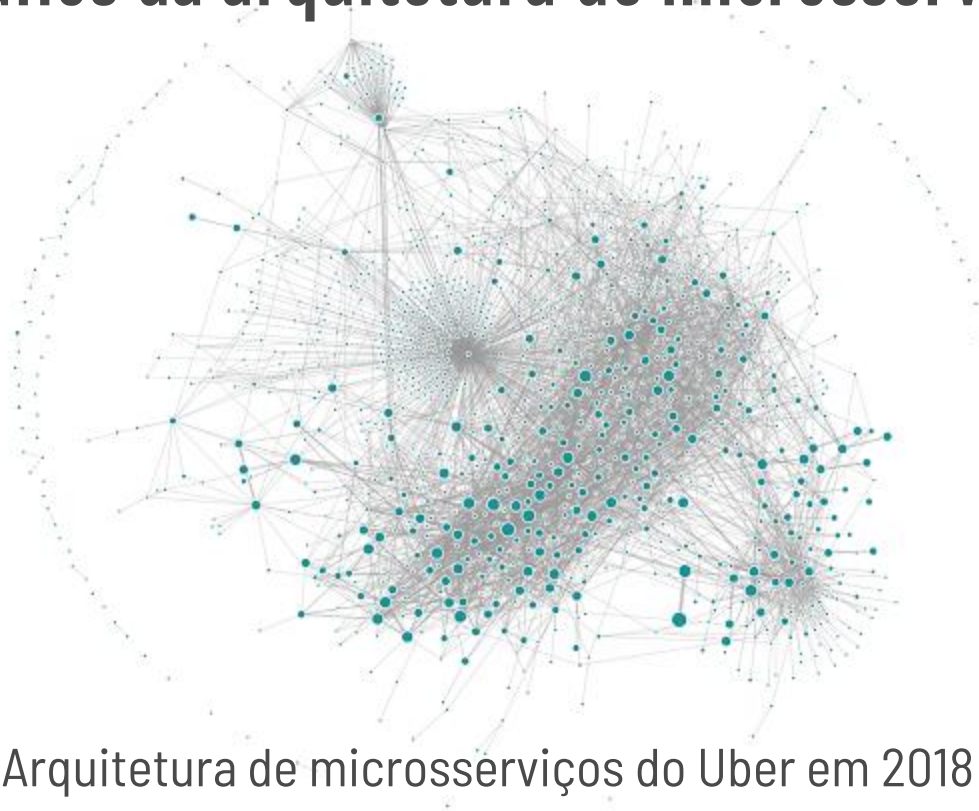
Vantagens:

- Simplicidade
- Facilidade de deploy
- Bem conhecido
- Fácil de debugar
- Fácil de testar
- Fácil de monitorar

Desvantagens:

- Difícil de manter quando cresce
- Indisponibilidade durante deploy
- Trabalho mais difícil com grandes times
- Escalabilidade não flexível
- Tecnologia restrita

Desafios da arquitetura de microsserviços



Arquitetura de microsserviços do Uber em 2018



02

O conceito de uma API REST



Entendendo o que é uma API

É um conjunto de definições e protocolos usado no desenvolvimento e na integração de software de aplicações. API é um acrônimo em inglês que significa ***interface de programação de aplicações***.



API Privada



API de Parceiros

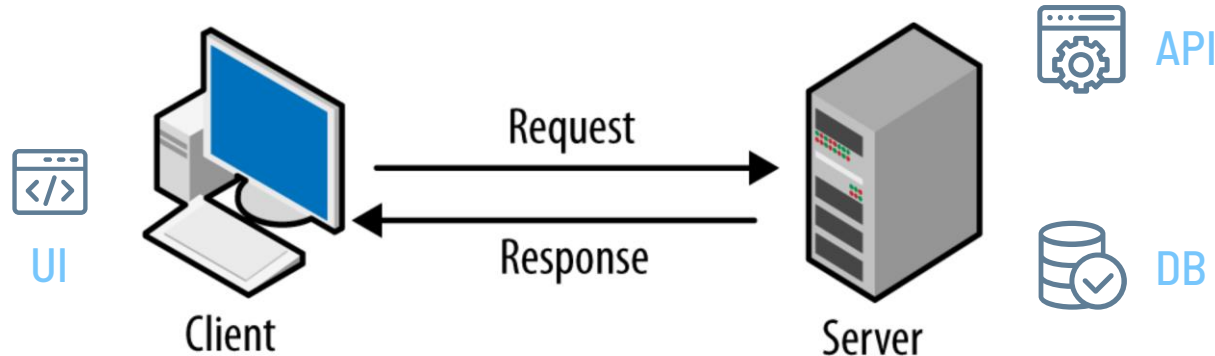


API Pública

Abordagens para políticas de lançamento de APIs.

Mas o que significa ser REST?

Significa que é adotado um estilo baseado no protocolo HTTP. Basicamente preconiza o uso adequado desse protocolo. REST é um acrônimo para ***Representational State Transfer***.



Boas práticas do HTTP

Uso
adequado
dos métodos
HTTP

Uso
adequado de
URL's

Uso de
códigos de
status
padronizados

Uso
adequado de
cabeçalhos
HTTP

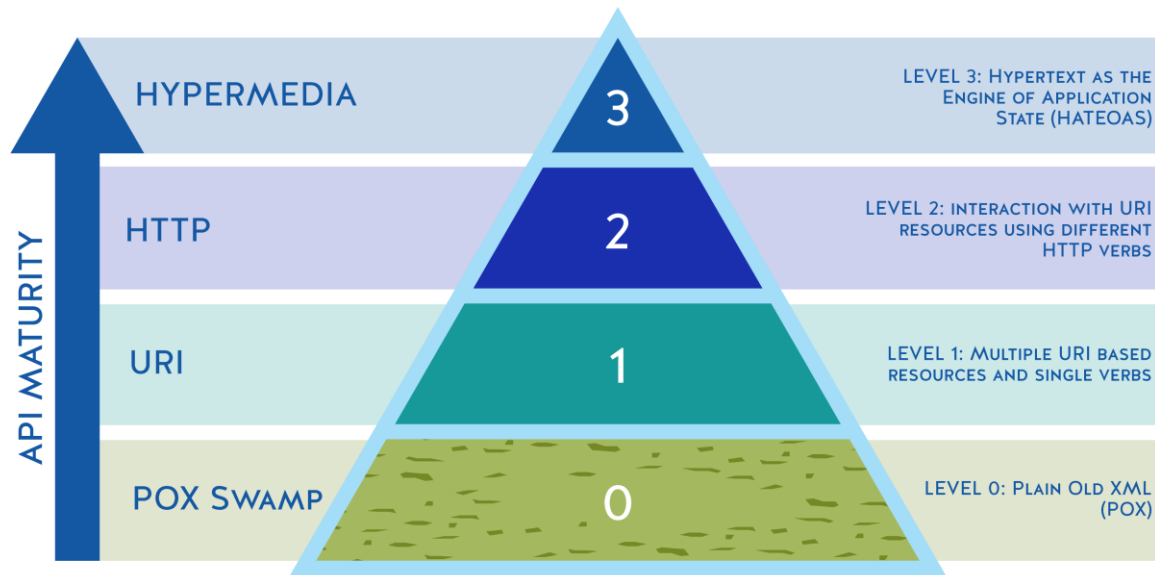
Interligações
entre vários
recursos
diferentes

03

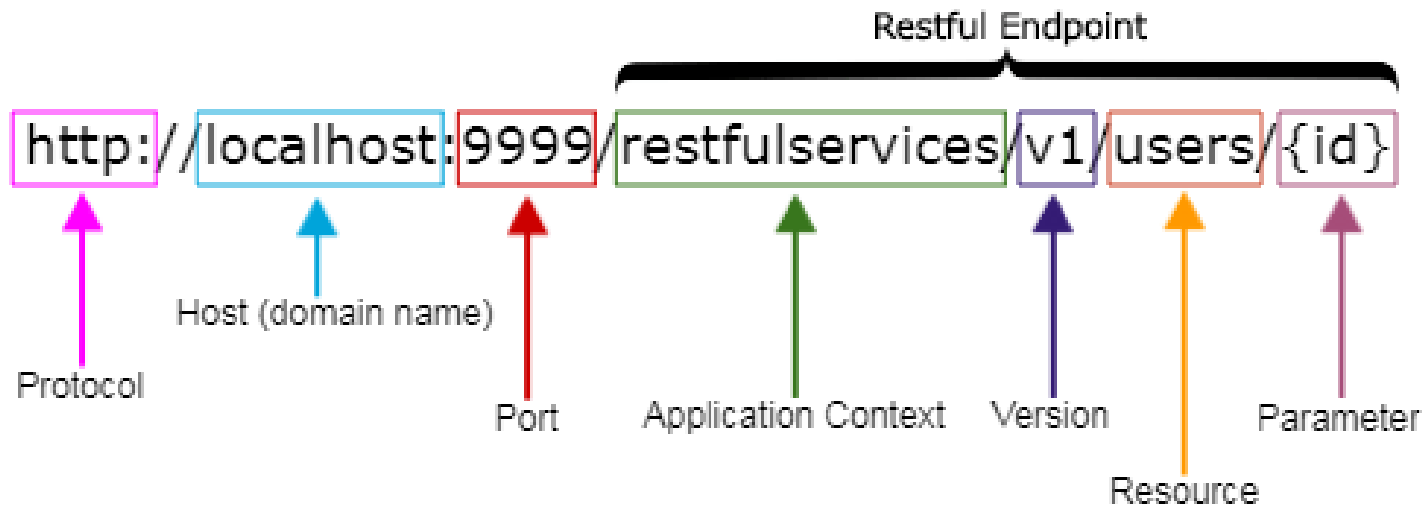
As boas práticas de uma aplicação RESTful

Maturidade de uma API REST

RICHARDSON MATURITY MODEL



Anatomia de uma URL



Os métodos/verbos HTTP

Task	Method	Path
Create a new task	POST	/tasks
Delete an existing task	DELETE	/tasks/{id}
Get a specific task	GET	/tasks/{id}
Search for tasks	GET	/tasks
Update an existing task	PUT	/tasks/{id}

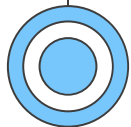
Código de status



Lista completa em:

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status>

Fonte: <https://itnext.io/api-calls-and-http-status-codes-e0240f78f585>



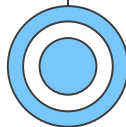
O corpo das requisições e respostas em APIs REST



```
<cep>01001-000</cep>  
<logradouro>Praça da Sé</logradouro>  
<complemento>lado ímpar</complemento>  
<bairro>Sé</bairro>  
<localidade>São Paulo</localidade>  
<uf>SP</uf>  
<ibge>3550308</ibge>  
<gia>1004</gia>  
<ddd>11</ddd>  
<siafi>7107</siafi>
```

{JSON}

```
{  
  "cep": "01001-000",  
  "logradouro": "Praça da Sé",  
  "complemento": "lado ímpar",  
  "bairro": "Sé",  
  "localidade": "São Paulo",  
  "uf": "SP",  
  "ibge": "3550308",  
  "gia": "1004",  
  "ddd": "11",  
  "siafi": "7107"  
}
```





04

Alguns exemplos práticos



Implementado uma API REST com Spring Framework



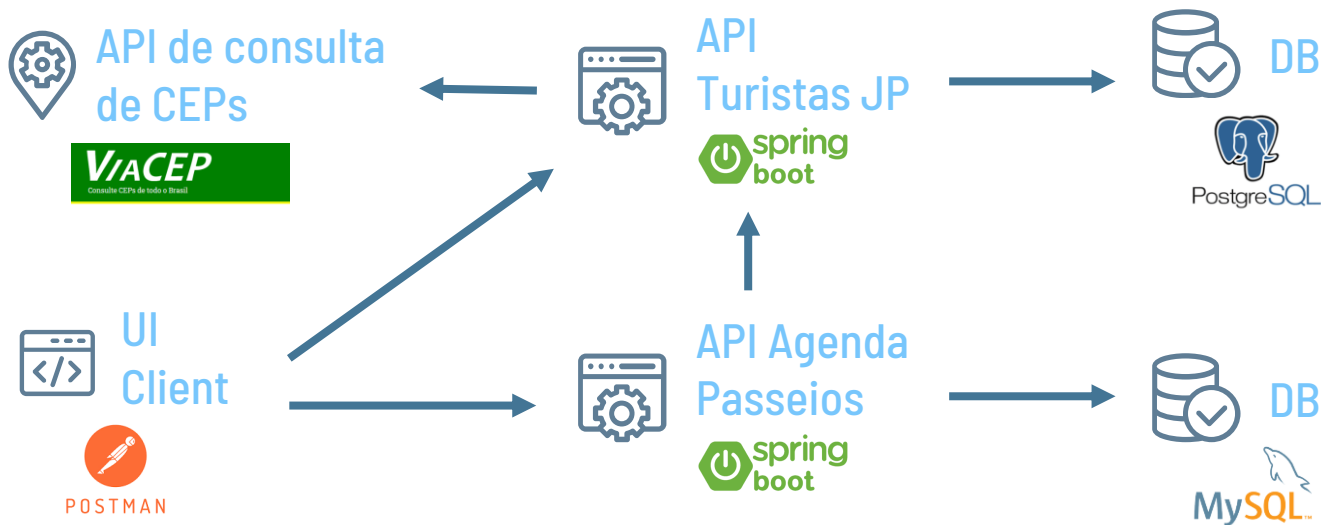
<https://start.spring.io>



Um caso hipotético de integração de microsserviços



Uma empresa de turismo de João Pessoa está precisando de um sistema que cadastre turistas e realize agendamento de passeios para esses turistas previamente cadastrados.



Obrigado!

Você tem alguma pergunta?
emersondevelops@gmail.com



emersondevelops



projeto e apresentação



CRÉDITOS: O template desta apresentação foi criado por [Slidesgo](#), incluindo ícones por [Flaticon](#), infográficos e imagens por [Freepik](#) e ilustrações por [Stories](#)