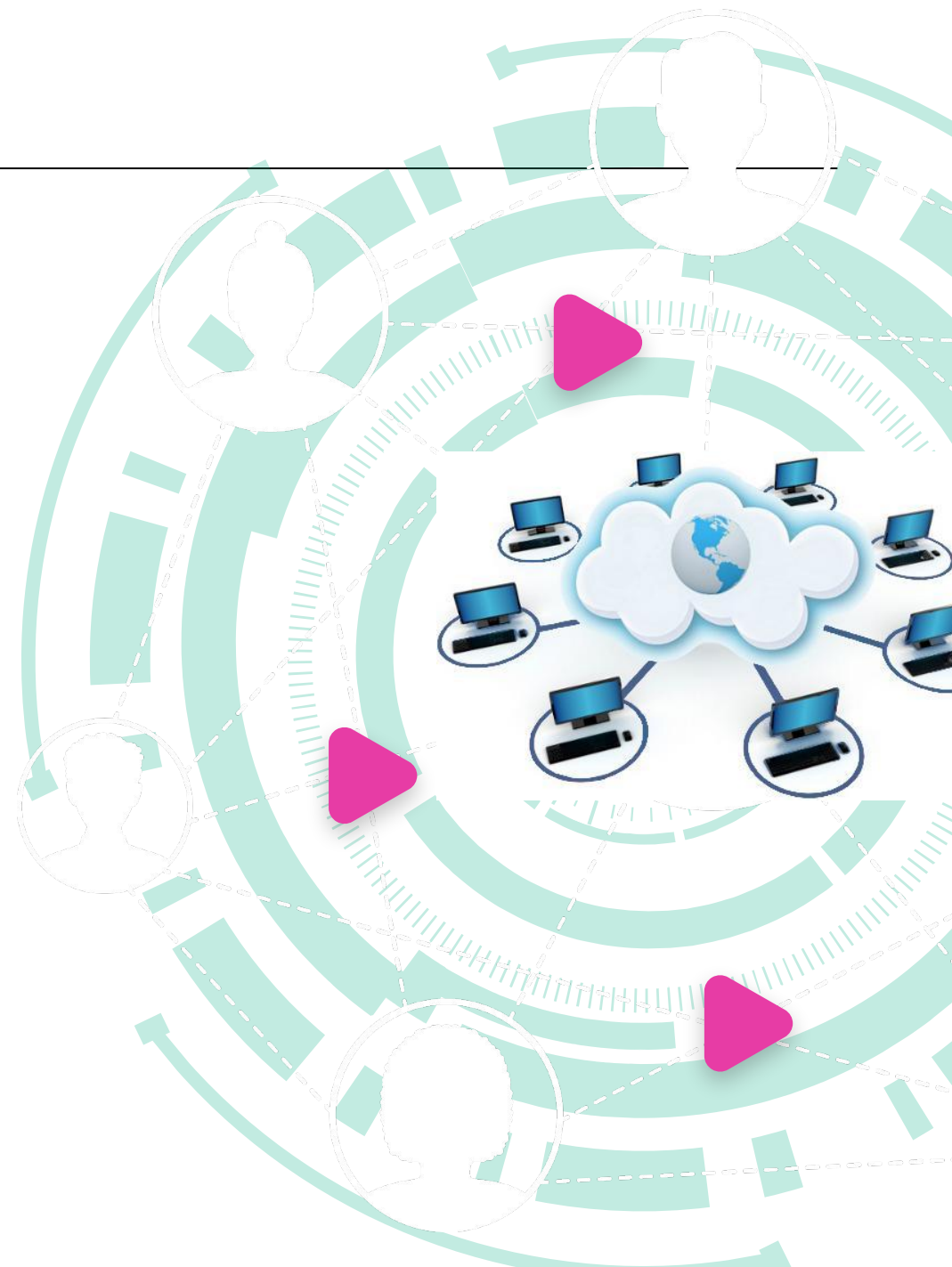
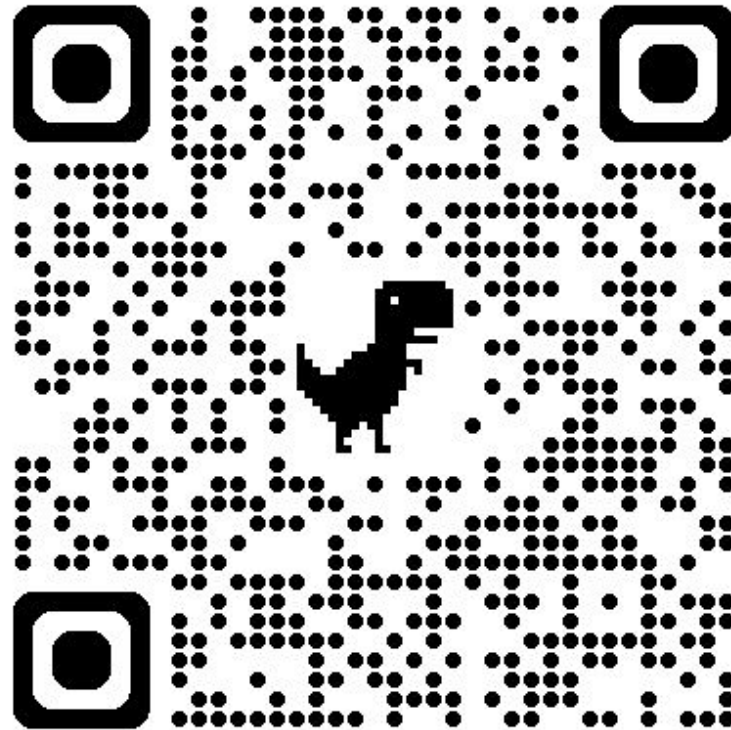


Sistemas Distribuidos e Mobile

Modelo de Maturidade de Richardson para APIs REST



<https://github.com/otavianosilverio/Sistemas-Distribuidos>

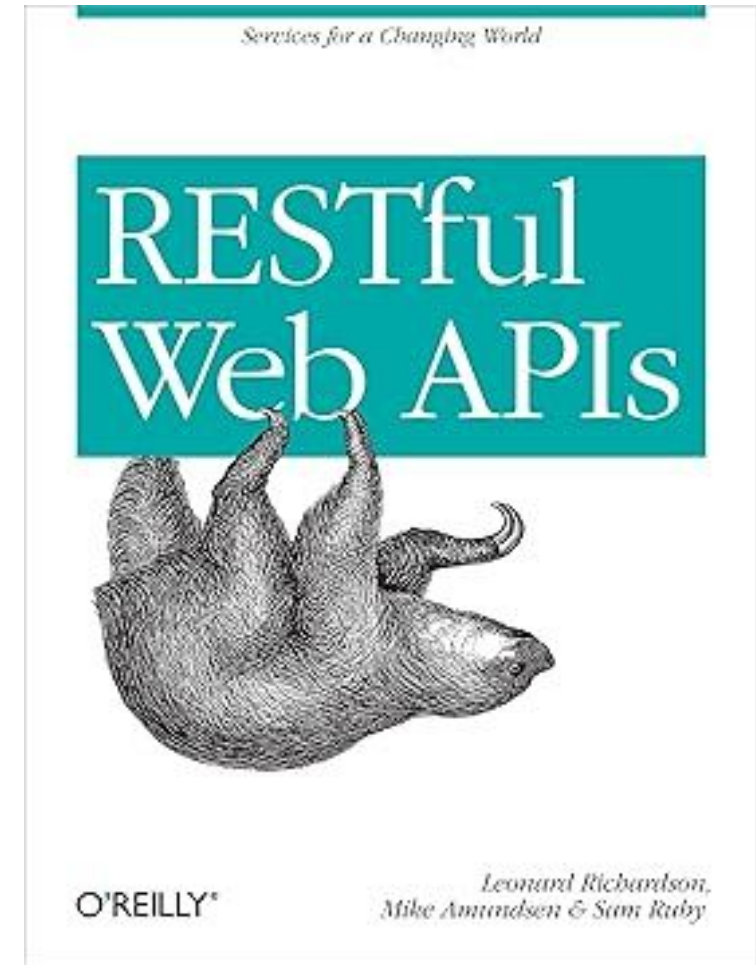


API REST

No início dos anos 2000 Roy Fielding apresentou em sua tese de doutorado o padrão REST, o principal objetivo desse padrão era tentar resolver problemas do protocolo SOAP, como alta complexidade de implementação e lentidão na comunicação. Embora não seja um padrão novo, REST tem ganhado bastante destaque nos últimos anos, sobretudo depois da popularização da arquitetura de microserviços. O problema é que nem sempre implementamos o padrão REST da forma que foi originalmente especificada por Roy Fielding, já que para ele uma API só pode ser considerada REST se de fato implementar um conjunto de constraints.

{ REST }

Para padronizar e facilitar o desenvolvimento de APIs REST, Leonard [Richardson](#) propôs um modelo de maturidade para esse tipo de API, definido em 4 níveis



O que é o Modelo de Maturidade de Richardson?

{ REST }

- Um framework que avalia o nível de aderência de uma API aos princípios REST.
- Propõe quatro níveis de maturidade: Nível 0, Nível 1, Nível 2, e Nível 3.

Nível 0 – The Swamp of POX

- Uso de uma única URI para todas as operações.
- APIs que seguem este nível utilizam métodos como POST para todas as requisições.
- Exemplo: Serviços SOAP encapsulados em uma API REST.

Conceito chave: Falta de uso adequado de recursos HTTP e verbos como GET, POST, PUT, DELETE.

Nível 0 – The Swamp of POX

Verbo HTTP	URI	Operação
GET	/getClient/1	Pesquisar
POST	/salvarCliente	Criar
POST	/alterarCliente/1	Alterar
GET/POST	/excluirCliente/1	Excluir

No modelo acima fazemos uso apenas dos métodos GET e POST e como já citado, os nomes dos métodos não seguem nenhum padrão.

Nível 1 – Recursos

- Introdução de recursos como entidades da API.
- URLs representam diferentes recursos (por exemplo, /usuarios, /pedidos).
- Ainda não há uso adequado dos verbos HTTP.

Conceito chave: APIs agora têm endereços específicos para recursos, mas o uso de verbos HTTP é limitado.

Nível 1 – Recursos

Verbo HTTP	URI	Operação
GET	/clientes/1	Pesquisar
POST	/clientes	Criar
PUT	/clientes/1	Alterar
DELETE	/clientes/1	Excluir

Um detalhe interessante é que no nível 1 já usamos os verbos HTTP de forma correta, já que se os verbos não fossem usados, as rotas de pesquisar, alterar e incluir ficariam ambíguas.

Nível 2 – Verbos HTTP

- Uso correto dos verbos HTTP: GET, POST, PUT, DELETE, etc.
- Cada verbo HTTP é mapeado para operações específicas (exemplo: GET para leitura, POST para criação).
- Exemplo: Requisições como GET /usuarios para buscar todos os usuários.

Conceito chave: APIs passam a respeitar o protocolo HTTP, melhorando a semântica das operações.

Verbo HTTP	Função
GET	Recuperar dados
POST	Gravar dados
PUT	Alterar dados
DELETE	Excluir dados

Nível 3 – HATEOAS

- As respostas das APIs contêm links para outras ações possíveis.
- Facilita a navegação e interação dos clientes com a API.
- Exemplo: GET /usuarios/123 inclui links como /usuarios/123/pedidos.

Conceito chave: APIs orientadas por hipermídia, tornando-as mais autodescritivas e interativas.

HATEOAS significa *Hypermedia as the Engine of Application State*.

Nível 3 – HATEOAS

```
{
  "id":1,
  "nome":"John",
  "sobrenome":"Doe",
  "links":[
    {
      "rel":"self",
      "href":"http://localhost:8080/clientes/1"
    },
    {
      "rel":"alterar",
      "href":"http://localhost:8080/clientes/1"
    },
    {
      "rel":"excluir",
      "href":"http://localhost:8080/clientes/1"
    }
  ]
}
```

HATEOAS significa *Hypermedia as the Engine of Application State*.

Comparação dos Níveis

- Nível 0: POX – Falta de recursos claros.
- Nível 1: Recursos – URLs para diferentes entidades.
- Nível 2: Verbos HTTP – Uso adequado de GET, POST, etc.
- Nível 3: HATEOAS – Navegação orientada por hipermídia.

Benefícios de APIs REST Maturidade

- Melhor uso dos padrões da web.
- Aumento da clareza e da interoperabilidade.
- Facilita a manutenção e o crescimento das APIs.

Glory of REST



Level 3: Hypermedia Controls

Level 2: HTTP Verbs

Level 1: Resources

Level 0: The Swamp of POX



Conclusão

- O Modelo de Maturidade de Richardson oferece um roadmap para o desenvolvimento de APIs RESTful.
- Seguir o modelo ajuda a garantir uma API escalável, compreensível e aderente aos princípios REST.

Referências

<https://martinfowler.com/articles/richardsonMaturityModel.html>

<https://rivaildojunior.medium.com/modelo-de-maturidade-de-richardson-para-apis-rest-8845f93b288>