

Quem sou eu?

Victor Osório

Software Development Specialist @Amdocs

twitter/vepo

github.com/vepo

- 5 anos com Apache Kafka
- 5 anos sem fazer CRUD
- + 17 anos com Java







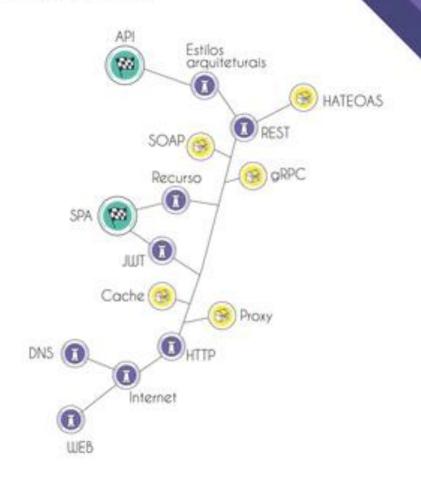




Moro em Campinas

Roadmap back-end

Conhecendo o protocolo HTTP e arquiteturas REST



Meu livro

Roadmap back-end

Conhecendo o protocolo HTTP e arquiteturas REST

https://www.casadocodigo.com.br/products/livro-roadmap-backend

Procure na Casa do Código

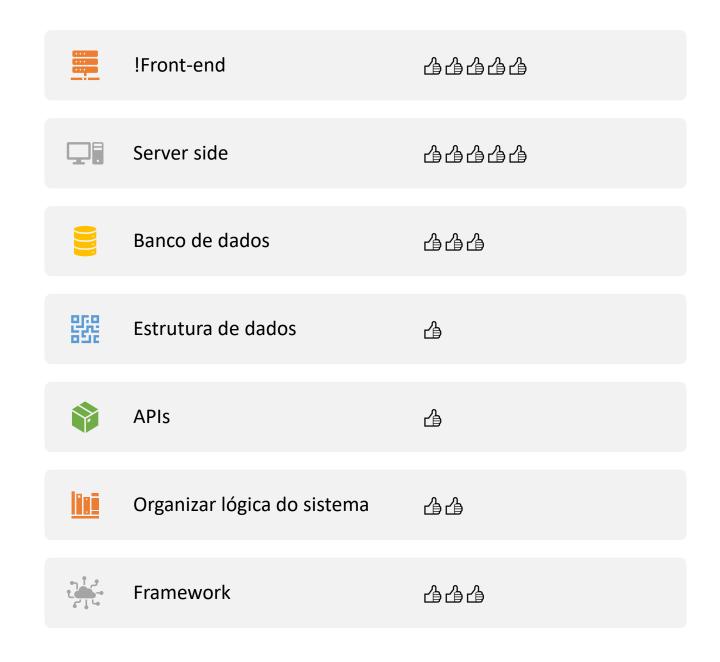
www.casadocodigo.com.br



O que é um desenvolvedor back-end?



O que é o desenvolvedor Back-end?



O que é o desenvolvedor Front-end?



Interface WEB

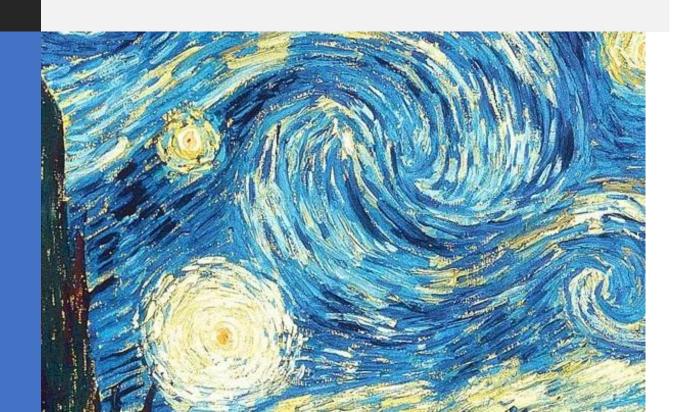
4444



Framework

44444

O que é o desenvolvedor Back-end?



- Aquele que n\u00e3o desenvolve
 - Mobile
 - Front-end
 - Machine Learning

Hipótese: Back-end, na verdade, é a área mãe de todo desenvolvimento de software.







Anos 1990

- Apache HTTPd
- C/C++
- Java
- Sem frameworks
- Server Side Rendering



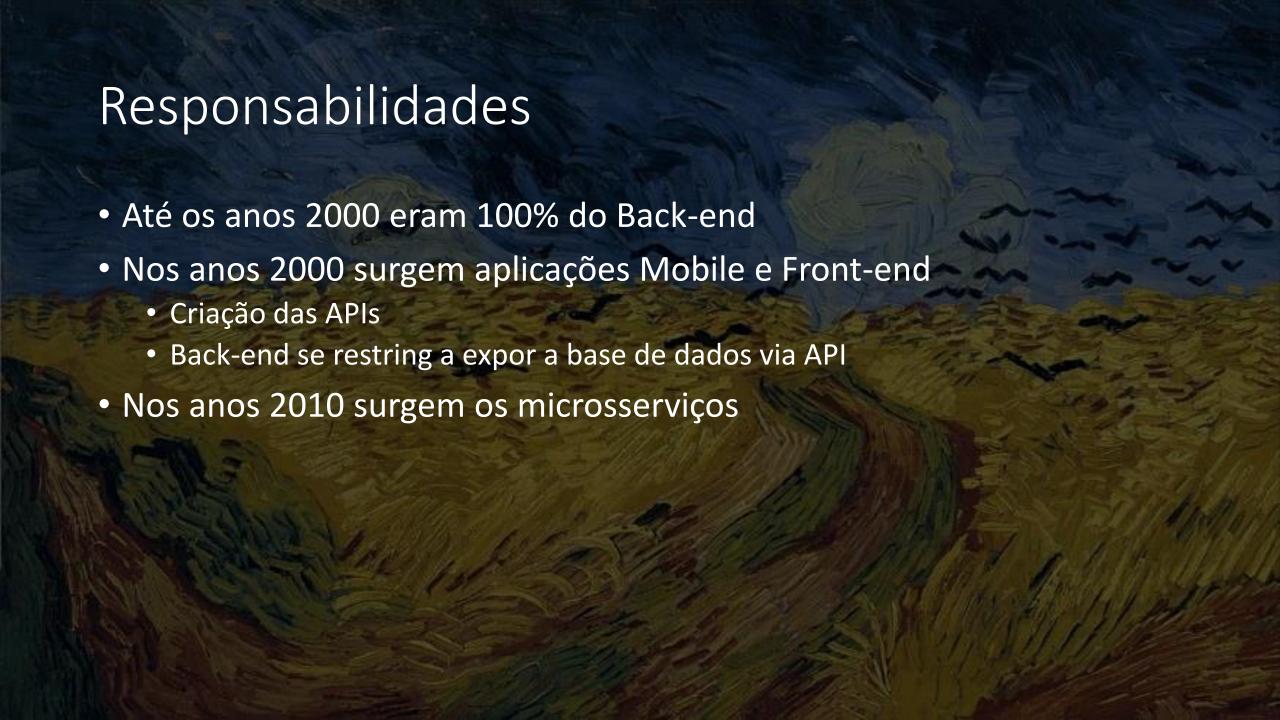




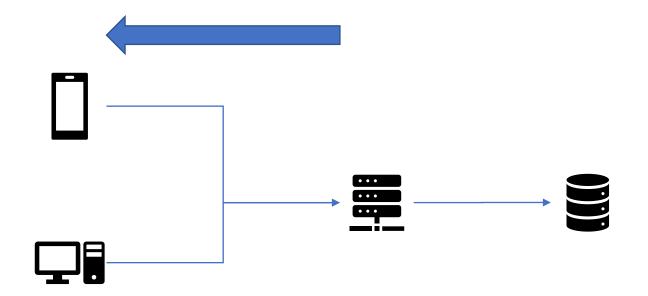








Migração de responsabilidades







Vamos falar de REST!?

- Não é só HTTP + JSON
 - Use os verbos HTTP
- Pense nos Recursos
 - Design da API
- Como expor operações fora do CRUD?
 - Crie nós de ações



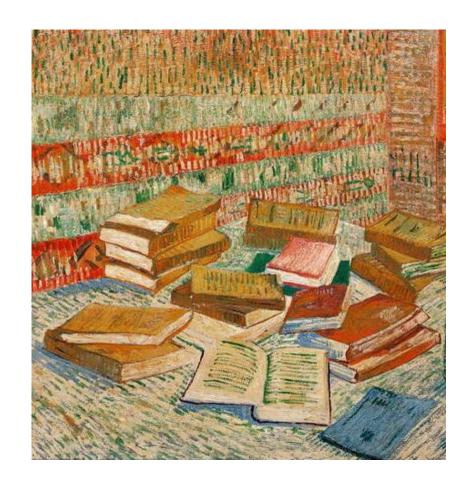


Vamos falar de REST!

- Como adicionar relacionamentos?
- Como sair do paradigma sincrono?

API da Biblioteca da Esquina

- **GET** /livros
 - Retorna lista de livros
- POST /livros
 - Cria novo livro
- **POST** /livros/{id}/emprestimo
 - Cria emprestimo do livro
- **GET** /usuario/{id}/emprestimos
 - Lista de emprestimos do usuário



Dicas para melhorar sua API

- Sua API não precisa somente mapear a base de dados
- Você pode ter dois modelos
 - API mapeia o caso de uso (DDD!!!!)
 - Base de dados é otimizada para o armazenamento
- Operações e Validações devem ser feitas no back-end
- Front-end deve fazer validações apenas para garantir a integridade do dado e não impactar na usuabilidade.





Vamos falar de ORM!?

- Quando usar?
 - SEMPRE!
- Quando não usar?
 - Só se você quiser otimizar na unha!
- Mas é lento?
 - Ou seu ORM é muito ruim
 - Ou você não está usando direito
 - PROCURE → Catálogo de Code Smells ORM
 - https://github.com/spgroup/ORM-Smells-Catalog

Síncrono e Assíncrono?!!?

- Nem tudo é HTTP
- Modele processos assícronos
- #ApacheKafka?
 - Provavelmente não!
- Pub/Sub
- Use e abuse da base de dados
- Cron Jobs



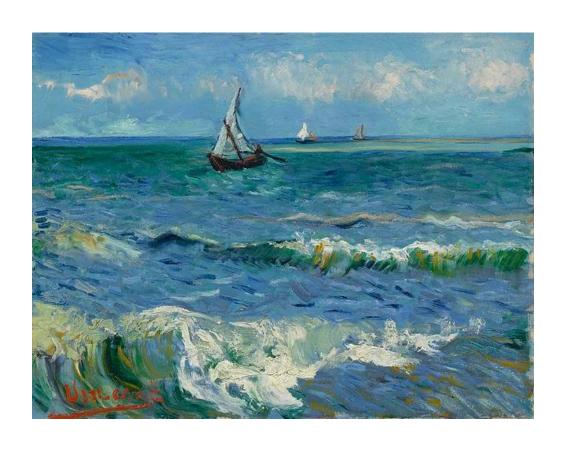
Cloud Native?!?!

- Cloud Native não é usar provedor cloud
- Tente
 - Container
 - Log no console
 - Não depender de arquivo
 - Configuração via variável de ambiente
 - Use uma build para todos ambientes



Navegação

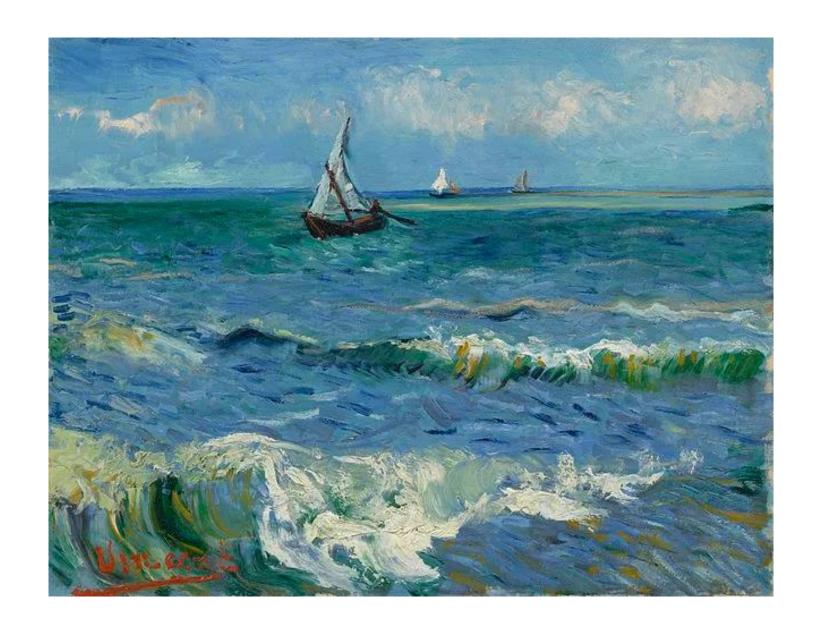
- Codificar no Front-end?!? NÃO!
- Conheça o HATEOAS



```
"id": 1,
"firstname": "Sergio",
"lastname": "Leone",
"year": 1929,
"_links": {
  "self": {
    "href": "http://localhost:8080/directors/1"
  "directors": {
    "href": "http://localhost:8080/directors"
  "directorMovies": {
    "href": "http://localhost:8080/director/1/movies"
```

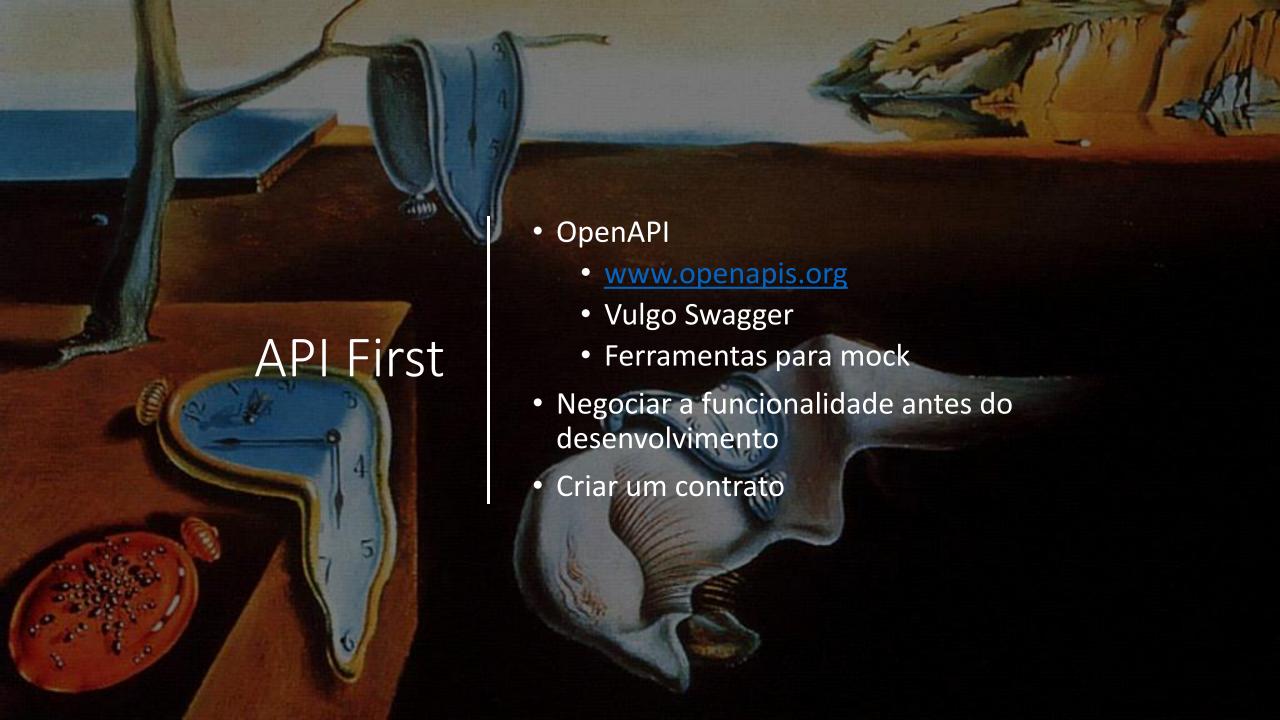
Trade offs do HATEOAS

- - Dificil de implementar
- + Reduz erros no cliente
- + Reduz acoplamento





Como envolver toda equipe?!?!



Considerações finais

- Back-end é o ponto central de qualquer arquitetura
- Também é o serviço que precisa escalar
- Requer mais Design e Conceitos
- Conheça os componentes e evite adicionar complexidade



