

Meetups
OF THE MINISTRY
OF TESTING



RECIFE

Testando sua API com RSpec e HTTParty

MoT - Recife

QUEM SOMOS



Kamila Carvalho
QA @ Thoughtworks



/kamcarvalho



/kamilancs

Deyvirson Mendonça
QA @ Concrete



/deyvirsonbm



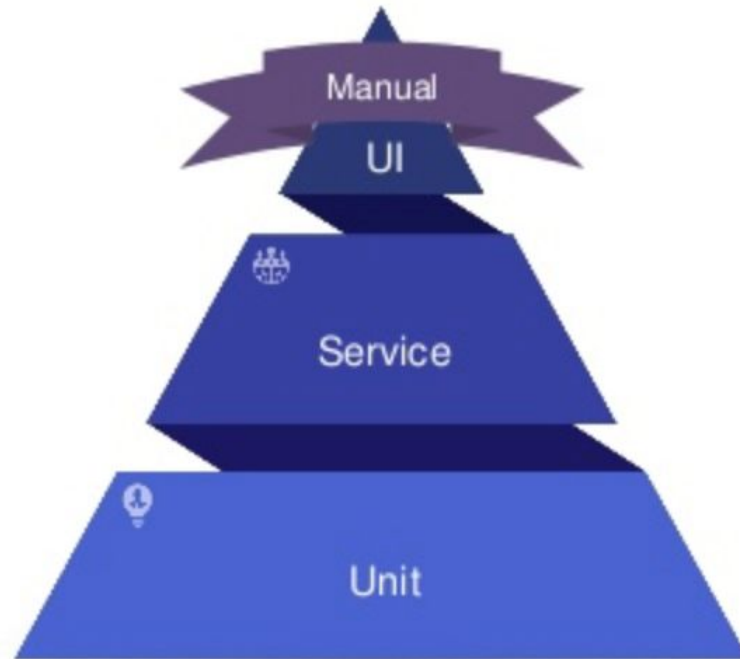
/deyvirsonmendonca



Vamos lá?

-
- Pirâmide de testes
- API
 - O que é API
 - Exemplos de API
 - Métodos HTTP
 - HTTP Status Code
 - Microserviços
 - HTTParty
 - RSpec
- Hands-on!

Pirâmide de Testes

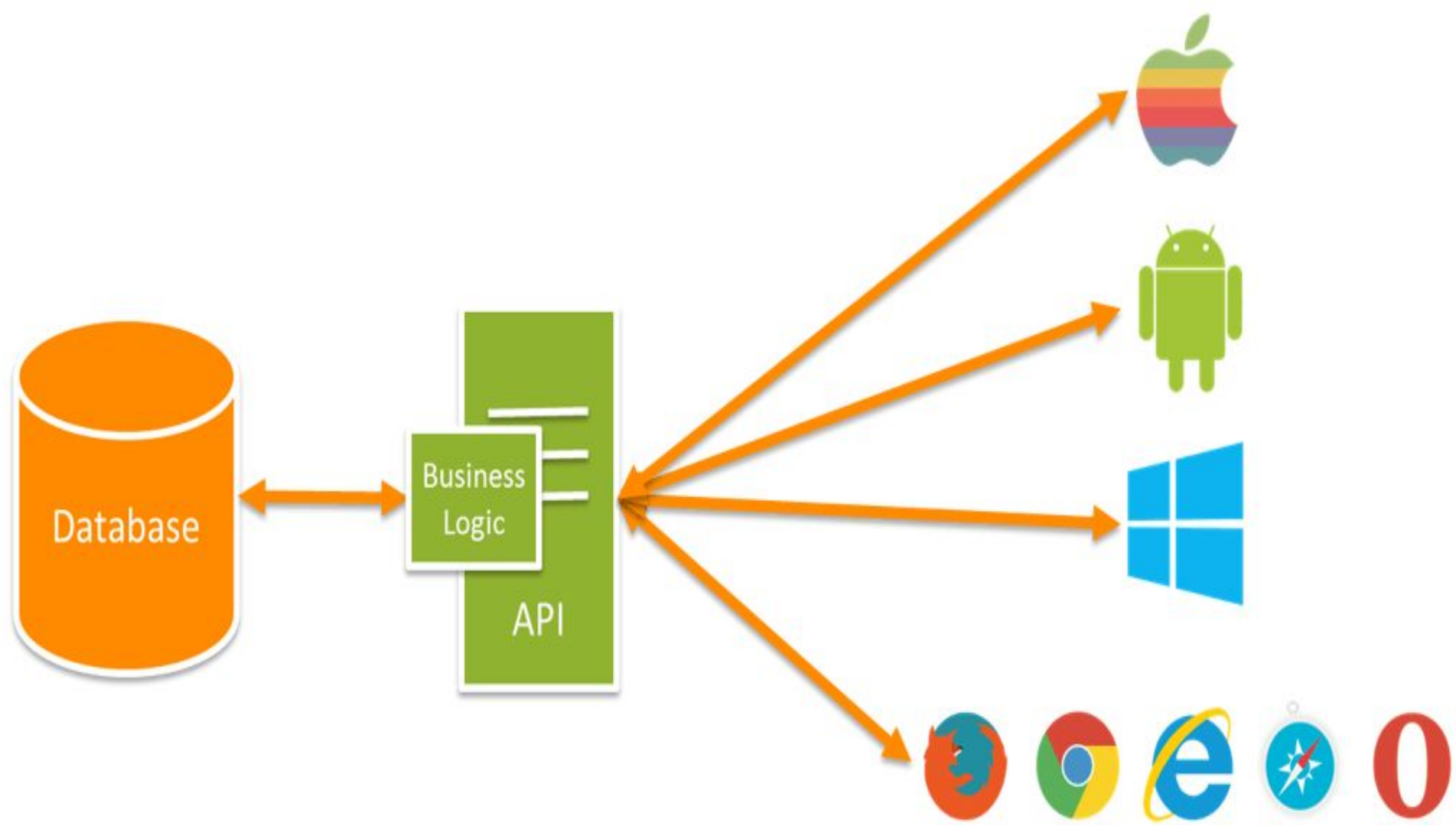




API - Application Programming Interface

API

- - Uma API é um serviço (negócio);
 - Uma ponte que conecta aplicações;
 - É invisível ao usuário comum;
 - Possibilita conectar diferentes tecnologias.





- Redes Sociais
 - Facebook
 - LinkedIn
- Pagamentos
 - Paypal
 - Cielo
- Localização
 - Google Maps





Requisições e Protocolos HTTP

REST - REpresentational State Transfer



- CSV, RSS, JSON e YAML;
- Protocolo HTTP/HTTPS;
- GET, POST, PUT e DELETE;
- Melhor curva de Aprendizado;
- Mensagens menores e mais eficientes com JSON comparada com XML;
- Mais rápido.



Exemplo de requisição

The screenshot displays a REST client interface with the following details:

- URL:** localhost:8000/notes/585185b33c501107d65fca0b
- Method:** PUT
- Environment:** No Environment
- Body Type:** x-www-form-urlencoded
- Body Fields:**
 - title: My Second Note Title
 - body: What a great note AGAIN
- Status:** 200 OK
- Time:** 118 ms
- Size:** 272 B
- Response Body (JSON):**

```
{
  "text": "What a great note AGAIN",
  "title": "My Second Note Title"
}
```

Two black arrows highlight the PUT method and the response body.

Métodos HTTP

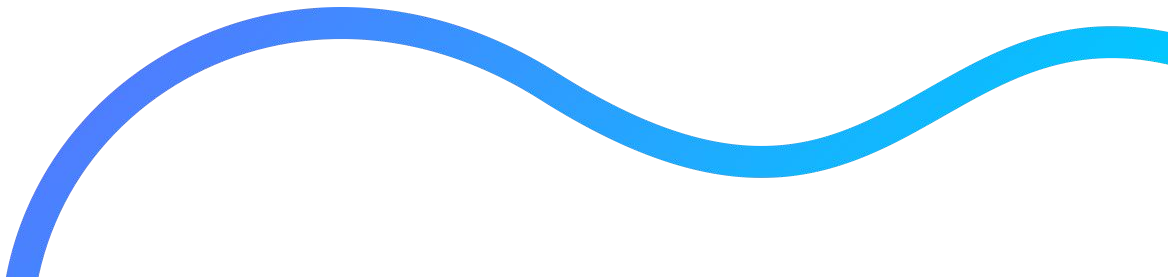


Existem 9 verbos, mas vamos explorar apenas 4:

- **GET**
 - A mais comum de todas, através dela pedimos a representação de um recurso;
- **POST**
 - O método post é usado quando queremos **criar** um recurso;
- **PUT**
 - Utilizado para atualizar objeto já existente;
- **DELETE**
 - Exclui o recurso especificado

API

HTTP Status Code



HTTP Status Code

- - Presente nas arquiteturas do tipo REST;
 - Os status code são emitidos pelo servidor em resposta a uma solicitação feita do cliente para o servidor;
 - O primeiro dígito especifica uma das cinco classes de resposta;

HTTP Status Code



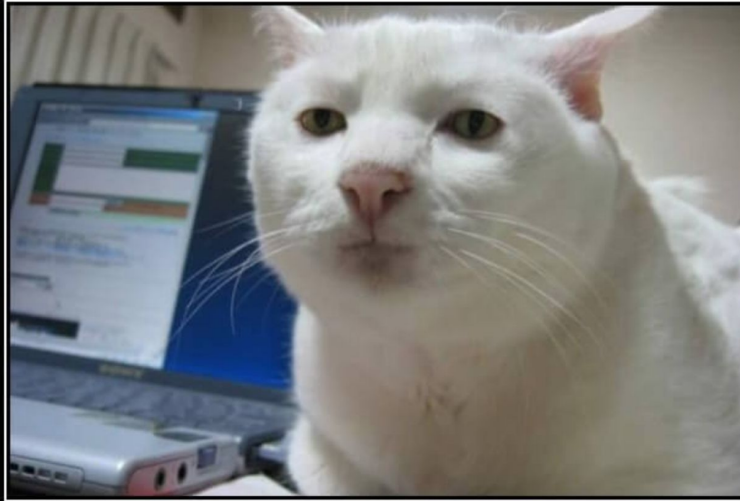
- Os códigos informativos tem o padrão 1xx;



HTTP Status Code



- Os códigos de sucesso tem o padrão 2xx;

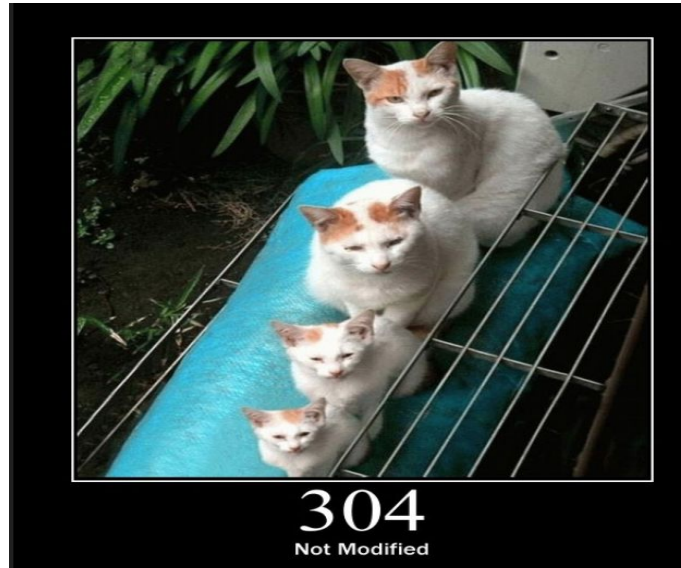


200

HTTP Status Code



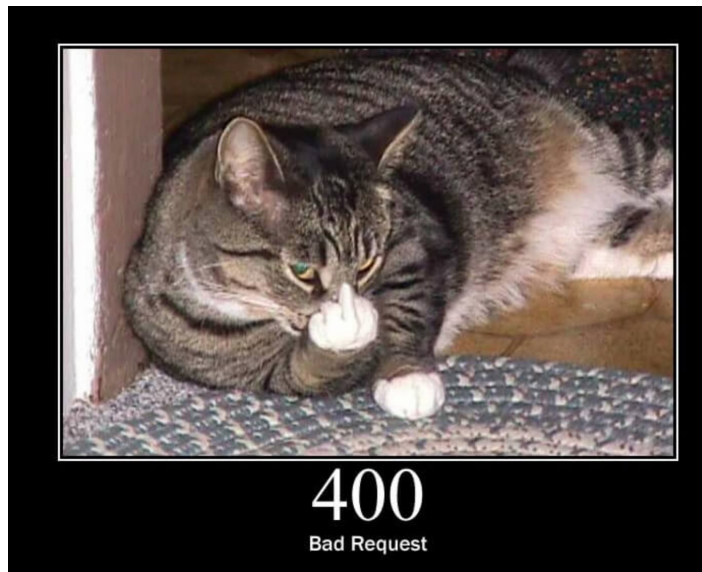
- Os de redirecionamento 3xx;



HTTP Status Code

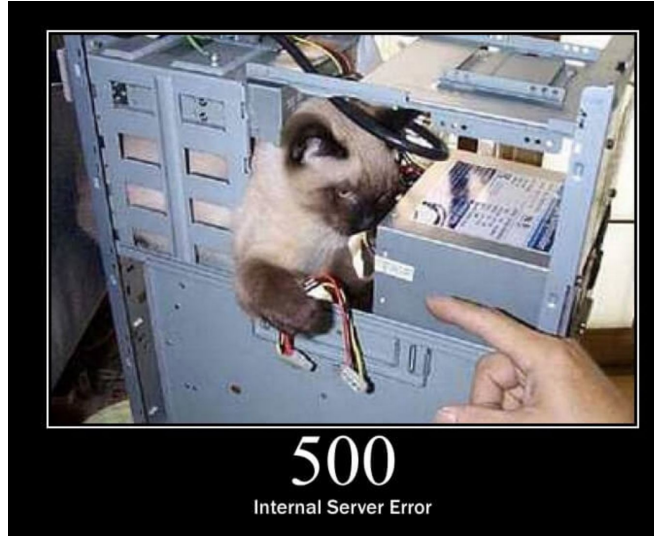


- Erro do cliente 4xx;

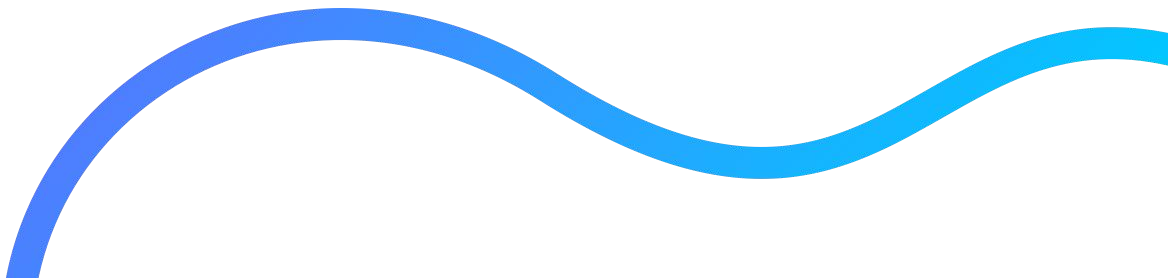


HTTP Status Code

- - Erro de servidor 5xx.



Microserviços

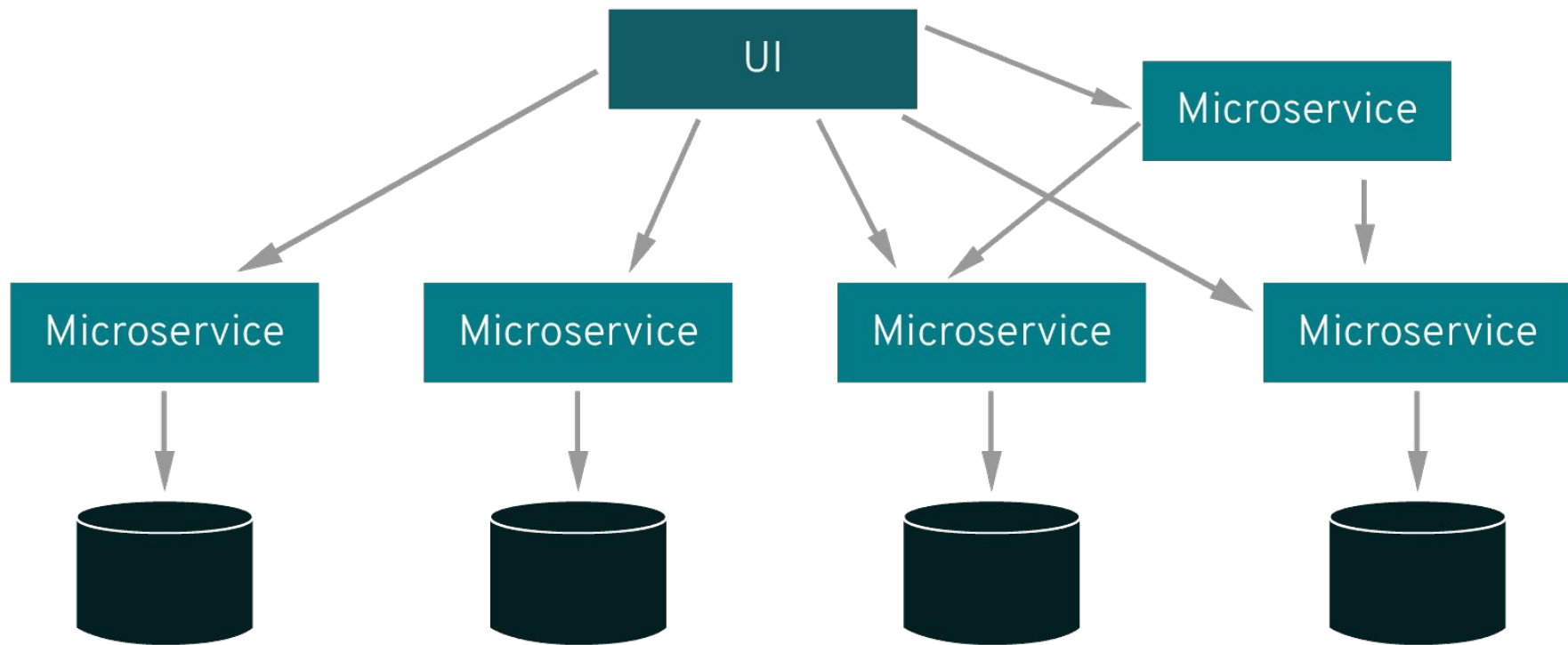


Microserviços



- Uma suíte de pequenos serviços;
- Serviços construídos em cima de capacidades de negócio;
- Escalabilidade;
- Resiliência;
- Capacidade de composição;
- Manutenção mais eficiente.







Mas e como testar toda essa suíte de serviços?

Testing Strategies

```
graph TD; A((Testing Strategies)) -.- B((Unit Testing)); A -.- C((Integration Testing)); A -.- D((Component Testing)); A -.- E((Contract Testing)); A -.- F((End-to-End Testing));
```

Unit
Testing

Integration
Testing

Component
Testing

Contract
Testing

End-to-End
Testing

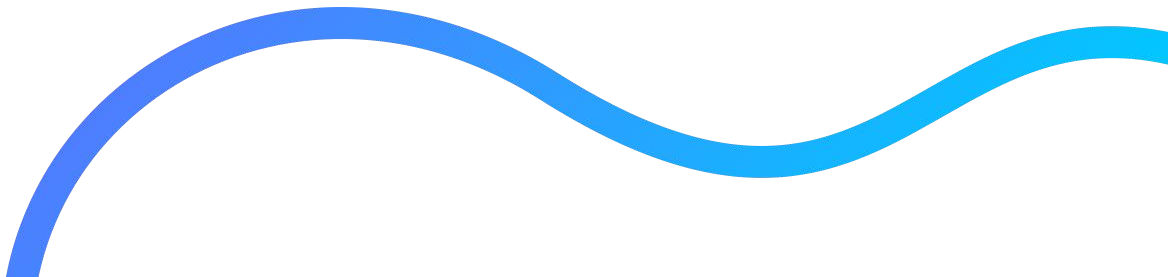


HTTParty - Makes http fun again!

HTTParty

- - É uma gem do Ruby;
 - Realiza testes do tipo - GET, POST, PUT, DELETE e PATCH;
 - Ótima para testes rápidos, quanto para regressão automatizados.

RSpec



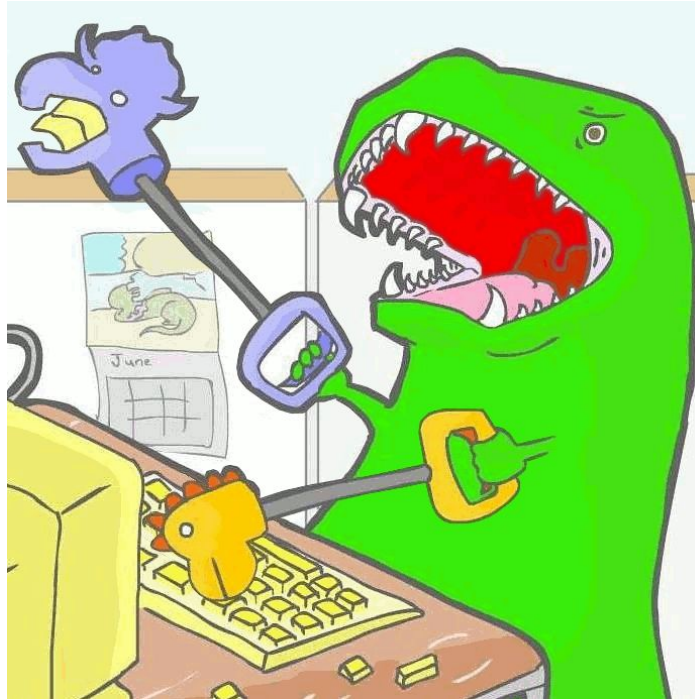
RSpec

- - Desenvolvimento Orientado por Comportamento para Ruby.
 - Sintaxe de fácil compreensão.
 - Exportação de report formato HTML
 - Permite que você escreva testes em uma linguagem mais natural



Hands on!

https://github.com/deyvirsonbm/ministry_of_testing_sp





Dúvidas?

Meetups
OF THE MINISTRY
OF TESTING



RECIFE