

WORKSHOP

Produzindo Especificações Ágeis e Testes Funcionais com Concordia

THIAGO DELGADO PINTO

SEMANA DE EXTENSÃO 2019

BSI - CEFET/RJ NOVA FRIBURGO

13/05/2019

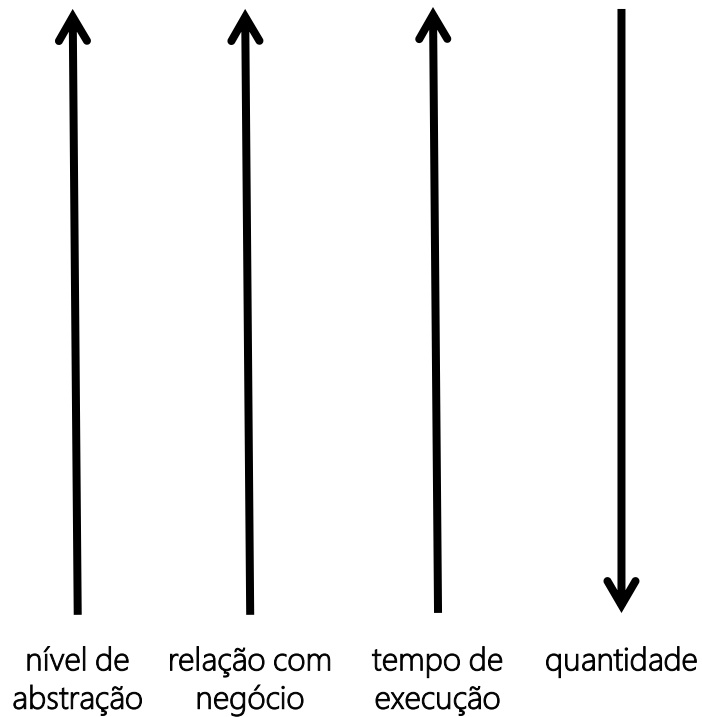
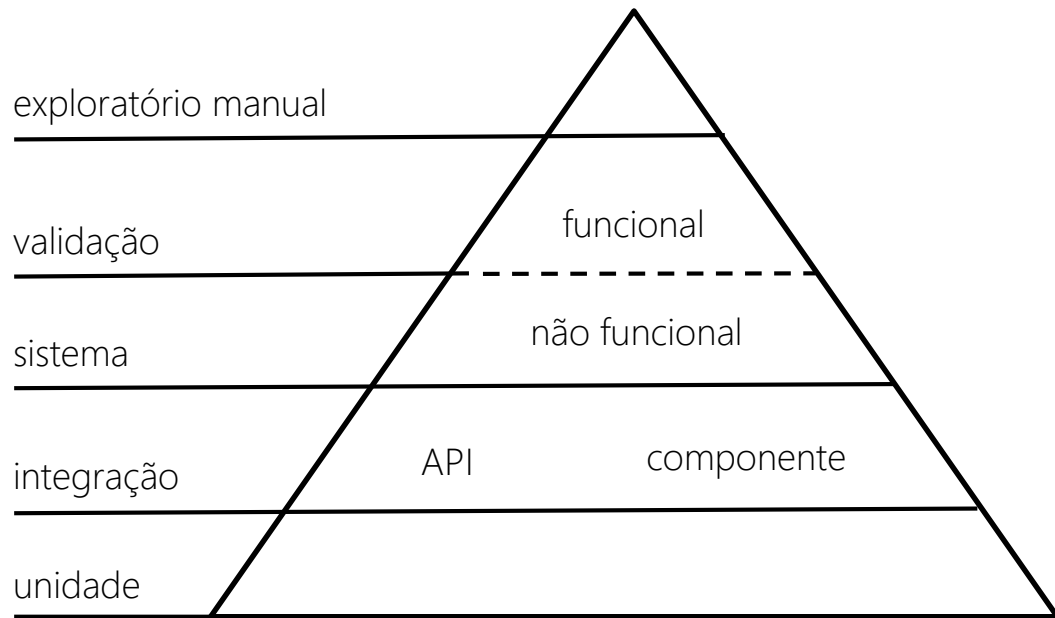


Licença Creative Commons 4

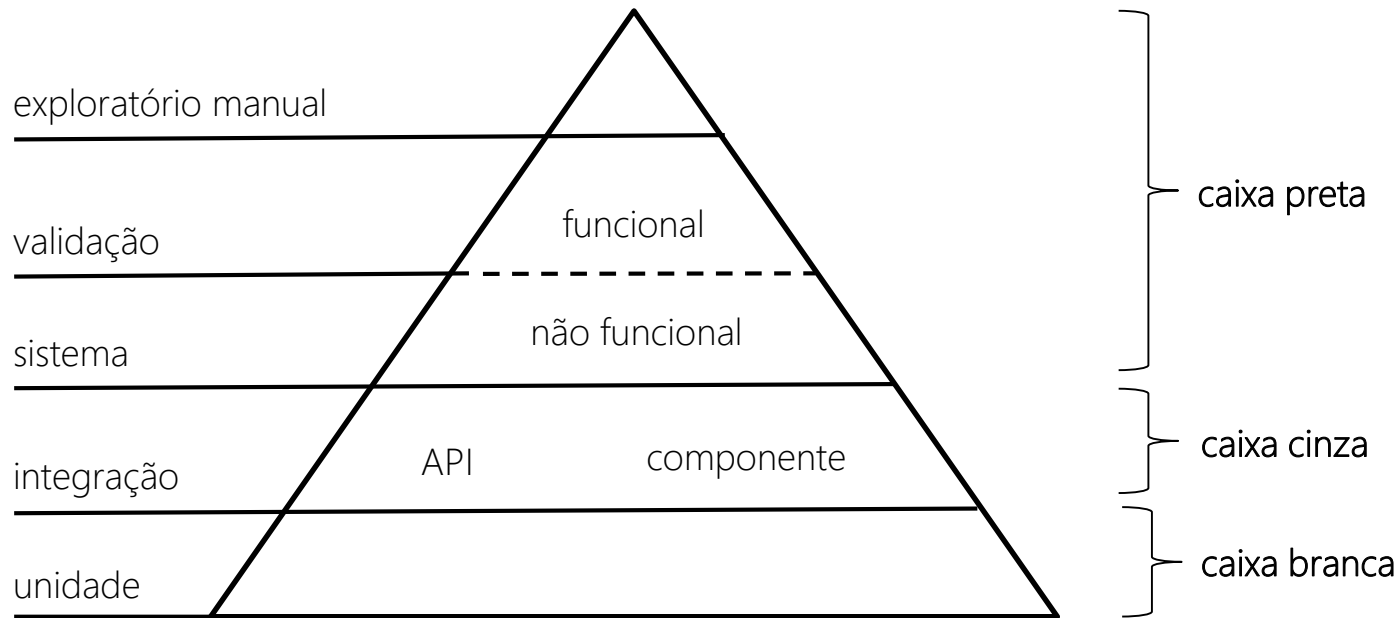
teste
automatizado

Por quê?	Descobrir defeitos
O/em que?	Qualquer software
Quem?	Desenvolvedor, Testador, CQ/QA, DevOps
Quando?	Ao longo (TDD vs BDD vs outro)
Quanto custa?	Depende. Quanto custa não testar?
Como?	Varia (tipo de teste, tecnologias, etc.)

PIRÂMIDE



NÍVEL DE ABSTRAÇÃO



O QUE É VERIFICADO

teste unitário

- caminhos independentes
- manipulação de erros
- condições limite
- estrutura de dados ou lógica
- interface de classes ou funções

teste de integração

- funcionamento em conjunto de classes ou funções

O QUE É VERIFICADO

teste de sistema

se requisitos **não funcionais** estão sendo atendidos
recuperação → em caso de falhas
segurança → acesso indevido, invasão, etc.
estresse → demanda anormal de recursos
desempenho
... (ver características de qualidade)

teste de validação

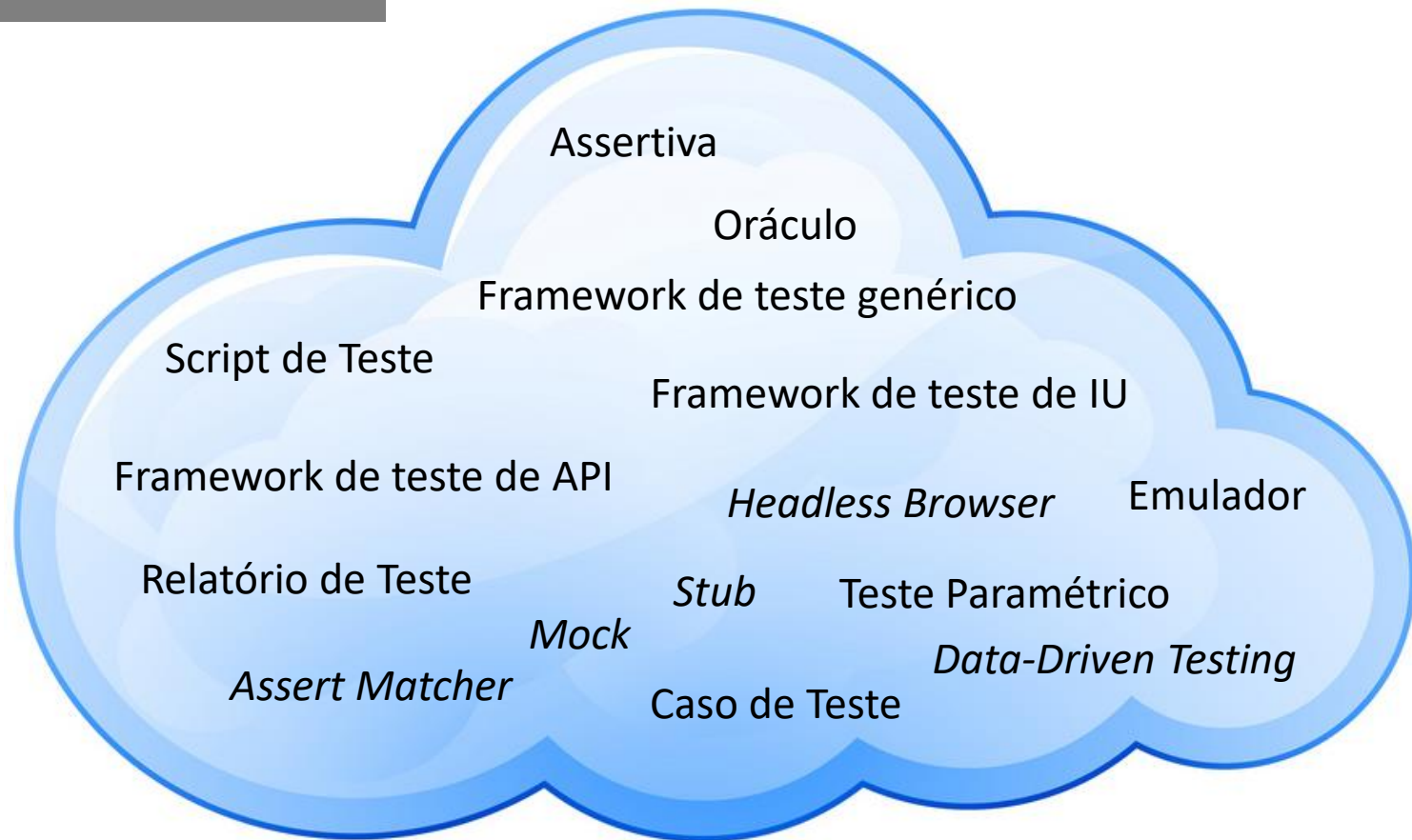
ações visíveis ao usuário (entradas, saídas)
conformidade com a **especificação de requisitos funcionais**
conformidade com necessidades (e desejos)

DETECÇÃO DE DEFEITOS

<i>atividade</i>	<i>mais baixo</i>	<i>modal</i>	<i>mais alto</i>
revisão informal de projeto	25%	35%	40%
inspeção formal de projeto	45%	55%	65%
revisão informal de código	20%	25%	35%
inspeção formal de código	45%	60%	70%
programação em pares	40%	50%	60%
modelagem ou prototipagem	35%	65%	80%
verificação pessoal em cópia impressa do código	20%	40%	60%
teste unitário	15%	30%	50%
teste de integração	25%	35%	40%
teste de regressão	15%	25%	30%
teste de sistema	25%	40%	55%

(JONES, 1986; JONES, 1996; SHULL *et al.*, 2002; MCCONNELL, 2004)

ALGUNS TERMOS



ferramentas
e frameworks

Software Testing Landscape



observações

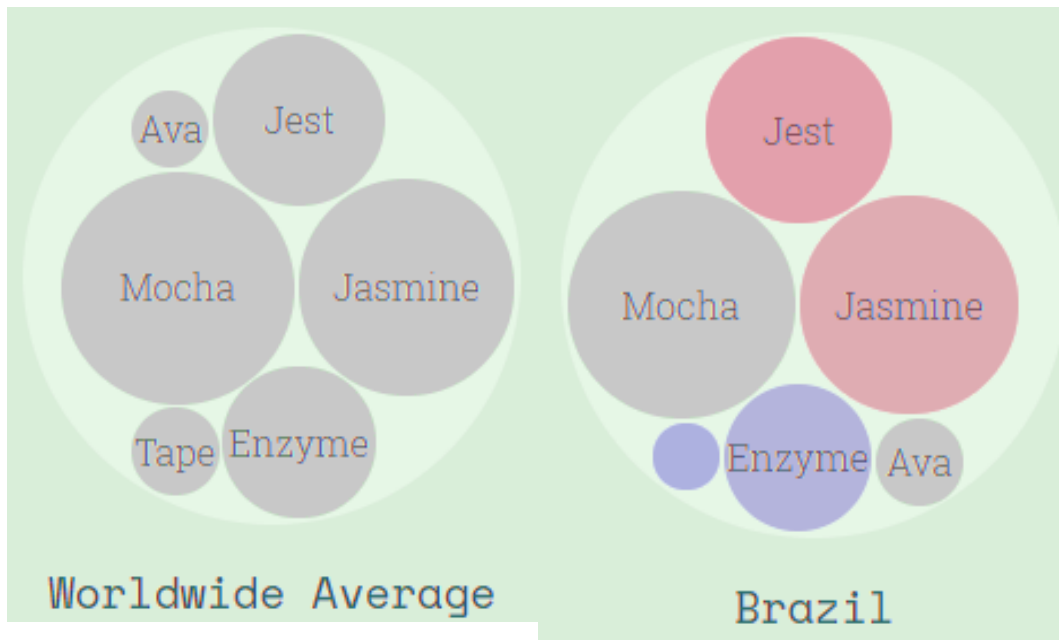
visão geral

JavaScript

soluções opensource

há muitas soluções pagas interessantes, mas...

não inclui alguns tipos de teste, como o de usabilidade
fica para pesquisa 😊



*Uso em 2017, segundo o State of JS:
<https://stateofjs.com/2017/testing/worldwide/>

exemplo com Mocha:

```
var potencia = require('potencia');

describe('potência', function() {
  it('número elevado a zero dá um', function() {
    var obtido = potencia( 2, 0 );
    assert.equal( obtido, 1 );
  });
});
```

exemplo com Jest:

```
var potencia = require('potencia');

describe('potência', function() {
  it('número elevado a zero dá um', function() {
    var obtido = potencia( 2, 0 );
    expect( obtido ).toEqual( 1 );
  });
});
```

forneem **estrutura de testes**

[Mocha](#), [Jasmine](#), [Jest](#), [Cucumber](#), [CodeceptJS](#)

forneem **assertivas** (para oráculos)

[Chai](#), [Jasmine](#), [Jest](#), [Unexpected](#)

forneem **exibição de resultados e monitoramento** (watch)

[Mocha](#), [Jasmine](#), [Jest](#), [Karma](#)

forneem **snapshots** (serializa e compara na próxima execução)

[Jest](#), [Ava](#)

forneem **mocks, spies e stubs** (imitam comportamento)

[Sinon](#), [Jasmine](#), [enzyme](#), [Jest](#), [testdouble](#)

forneem **relatórios de cobertura de código**

[Istanbul](#), [Jest](#), [Blanket](#)

forneem ou simulam um **navegador**

[Protractor](#), [Nightwatch](#), [Phantom](#), [Casper](#), [Cypress](#), [CodeceptJS](#)

[supertest](#), [apickli](#), [api-easy](#), [frisby](#), [chakram](#)

exemplo de Chakram:

```
describe('GET para /usuario', function() {  
  it('responde com json', function() {  
    var r = chakram.get( '/usuario' );  
    expect( r ).to.have.header( 'Content-Type', /json/ );  
    expect( r ).to.have.status( 200 );  
    return chakram.wait();  
  });  
});
```

[supertest](#), [apickli](#), [api-easy](#), [frisby](#), [chakram](#)

exemplo de SuperTest:

```
describe('GET para /usuario', function() {  
  it('responde com json', function(done) {  
    api.get('/usuario')  
      .expect( 'Content-Type', /json/ )  
      .expect( 200, done );  
  });  
});
```

desempenho

[benchmark.js](#)

exemplo:

```
var suite = new Benchmark.Suite;  
var raiz1 = require( 'raiz' ).raiz1, raiz2 = require( 'raiz' ).raiz2;
```

```
suite  
.add('raiz quadrada com algoritmo 1', function() {  
  raiz1( 1000000 );  
})  
.add('raiz quadrada com algoritmo 2', function() {  
  raiz2( 1000000 );  
})  
.on('complete', function() {  
  console.log( Mais rápido é ' + this.filter('fastest').map('name'));  
})  
.run({ 'async': true });
```

carregamento

[Artillery](#)

exemplo:

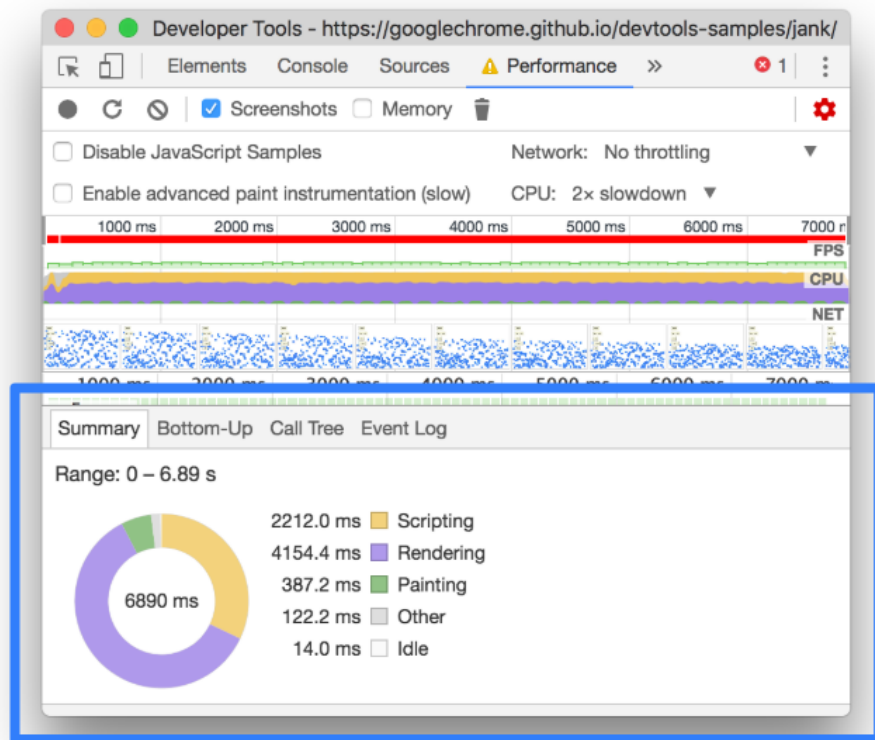
```
artillery quick --count 10 -n 20 https://artillery.io/
```

irá criar 10 "usuários virtuais", cada um disparando 20 requisições HTTP GET para o endereço informado

desempenho detalhado de aplicações web

[Google Chrome](#) (Dev Tools)

[Mozilla Firefox Developers Edition](#)



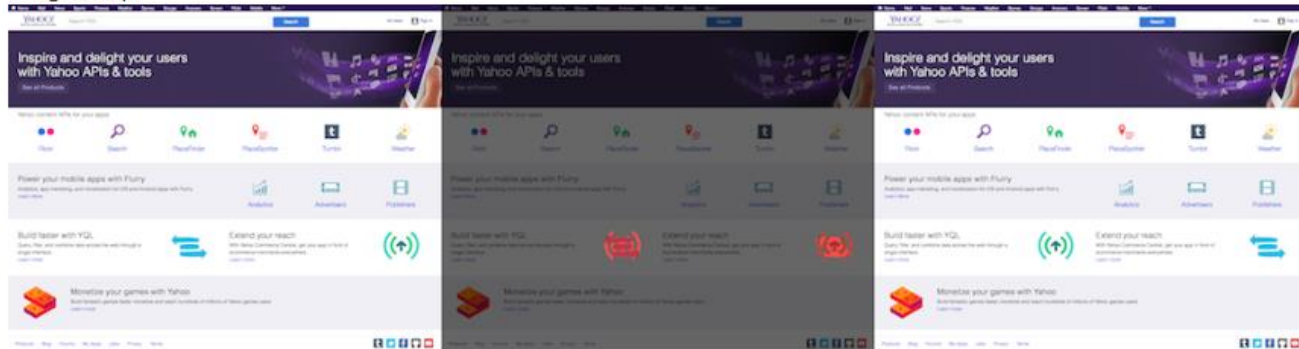
teste visual (comparação de conteúdos)

Kobold

Appraise

BlinkDiff

Image swap

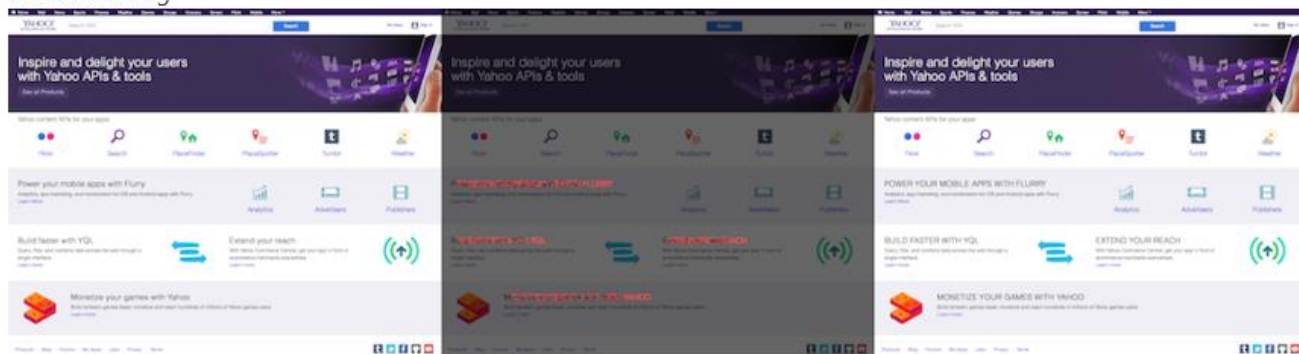


Exemplo do Kobold:

\$ `kobold test/ui/regression`

Irá comparar imagens em subdiretórios
`/approved`
`/build`
e irá gerar em caso de diferenças
`/highlight`

Text formatting



[Nightwatch](#), [Puppeteer](#), [WebDriver.io](#), [Casper](#), [TestCafe](#), [Cypress](#), [CucumberJS](#), [Tartare](#), [CodeceptJS](#)

<i>Recurso</i>	<i>Casper</i>	<i>Nightwatch</i>	<i>WebDriver.io</i>	<i>Protractor</i>	<i>CodeceptJS</i>
Screenshots	Sim	Sim	Sim	Sim	Sim
<i>Profiling</i> de memória e desempenho	Não	Sim (Chrome Driver)	Sim (Chrome Driver)	Sim (Chrome Driver)	Sim (Chrome Driver)
Análise de Cobertura	Sim	Sim	Sim	Sim	Sim
Suporta PageObject	Não	Não	Sim	Sim	Sim
Sup. testes Síncronos	Não	Não	Sim	Sim	Sim
Relatórios	CMD, xUnit	HTML, Alure, xUnit	HTML, Allure, xUnit, Perfecto	HTML, xUnit, Alure	HTML, CLI, xUnit
Nuvm e automação	Não	web e mobile	web e mobile	web e mobile	web e mobile

exemplo WebDriver.IO

```
var webdriverio = require('webdriverio');
var options = { desiredCapabilities: { browserName: 'chrome' } };
var client = webdriverio.remote(options);
client
  .init()
  .url('https://localhost/app/')
  .setValue('#login', 'admin')
  .setValue('#senha', '123456')
  .click('#entrar')
  .getText().then(function(text) {
    assert.equal( text, 'Olá' );
  })
  .end();
```

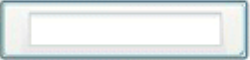
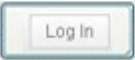

exemplo CodeceptJS

```
Feature('Login');
```

```
Scenario('Login de administrador com sucesso', (I) => {  
  I.amOnPage('http://localhost/app');  
  I.fillField('#login', 'admin');  
  I.fillField('#senha', '123456');  
  I.click('#entrar');  
  I.see('Olá');  
});
```

outras ferramentas

Sikuli (automação de tarefas)

```
31 #Logearse como usuario registrado
32 if exists(  ):
33     print("Usuario ya logeado")
34 else:
35     type( , "javipello")
36     wait(5)
37     type( , "*****")
38     wait(5)
39     click(  )
40     print("Usuario se acaba de logear")
41 wait(8)
42
```

Exemplo de Sikuli em uma aplicação desktop (retirado do site)

outras ferramentas

Selenium IDE

ferramenta de gravação e reprodução

Katalon Automation Recorder

ferramenta de gravação e reprodução

The screenshot displays the Selenium IDE - Home window. On the left, a sidebar shows the 'Tests' tab with a search bar and a list containing 'Selenium IDE*'. The main area shows a test suite with four steps:

	Command	Target	Value
1.	open	/	
2.	click at	css=a[title="Selenium Projects"]	52,15
3.	click at	link=Selenium IDE	67,7
4.	assert text	css=td > p	Selenium IDE is an integrated development enviro...

The fourth step, 'assert text', is selected. The right panel shows its configuration:

- Command: assert text
- Target: css=td > p
- Value: Selenium IDE is an integrated development environm
- Comment: (empty)

Below the test suite, a progress bar indicates 'Runs: 1' and 'Failures: 0'. At the bottom, the 'Log' tab shows the execution details:

Running 'Selenium IDE'

1. Trying to execute open on /... **Success**
2. Trying to execute clickAt on css=a[title="Selenium Projects"] with value 52,15... **Success**
3. Trying to execute clickAt on link=Selenium IDE with value 67,7... **Success**
4. Trying to execute assertText on css=td > p with value Selenium IDE is an integrated development environment for Selenium scripts. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests. Selenium IDE includes the entire Selenium Core, allowing you to easily and quickly record and play back tests in the actual environment that they will run in.... **Success**

'Selenium IDE' completed successfully

Exemplo do Selenium IDE (retirado do site)

exercícios – Katalon Recorder

1. Grave uma interação com o YouTube, em que você pesquisa um certo vídeo. Como oráculo, estabeleça que o nome do vídeo deva aparecer nos resultados.
2. Tente gravar ou fazer um script que faça um cadastro em algum serviço de e-mail (ex.: Gmail, Outlook, etc.) e adapte-o para poder cadastrar um novo e-mail na próxima execução.

Analise os problemas encontrados

gravação e reprodução - observações

ferramentas são práticas

porém, podem ter problemas de manutenção

- gravação de ações indesejadas

- mudança de interface (ex.: identificadores) pode requerer regravação

- problemático para funcionalidades grandes

- não permite modularização ou é preciso fazê-la manualmente

efetividade para teste pode ser baixa

- só repete o que foi gravado

- não testa caminhos novos ou dados novos

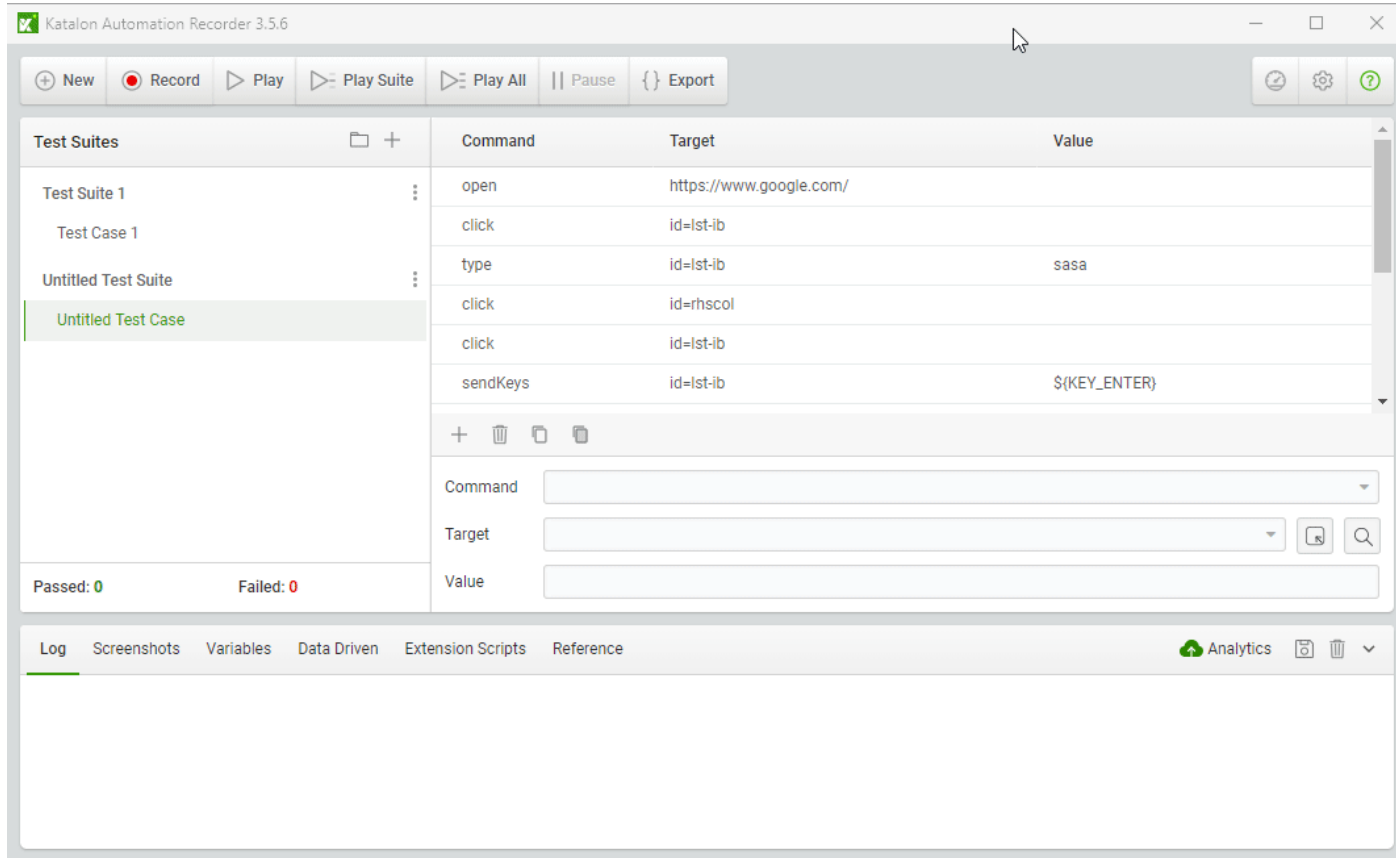
<https://github.com/thiagodp/katalon-concordia>



Chrome e Firefox

transforma testes gravados com **Katalon Recorder** em **Concordia**

bastante útil para **capturar** identificação de elementos da tela



outras soluções

error mining → "teste em produção" (geralmente pagas ☹)

[Sentry](#)

[TrackJS](#)

[Rollbar](#)

monitoram e reportam erros no lado cliente e servidor
geralmente apresentam em um *dashboard* fácil de compreender

*headless
browsers*

headless browser ("navegador sem cabeça")

simplificando, é um navegador **sem interface gráfica**

executa **sem ser exibido**, em *background*

não exibe conteúdo renderizado (DOM)

usa um *DOM virtual*

usos comuns

simular cliques ou outras ações em elementos da página

preencher formulários

verificar o desempenho do acesso via SSL

verificar o tempo de resposta de páginas

tirar fotos (*screenshots*) dos resultados

renderiza internamente para a foto, mas não exibe

vantagens e desvantagens

vantagens

- mais rápido que um navegador normal
- não assume o mouse/teclado

desvantagens

- pode não lidar bem com AJAX/AJAJ
- renderização pode não ser igual a um navegador normal
 - exibição* pode ajustar os elementos na tela de um jeito diferente

Mozilla Firefox em headless mode

ex. de uso: **firefox -headless -screenshot https://site.com**
pode ser controlado via código pelo [SlimerJS](#)



Google Chrome em headless mode (ou Headless Chrome)

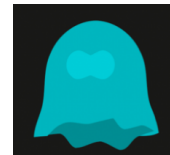
ex. de uso: **chrome --headless --remote-debugging-port=9222 https://site.com**
pode gerar PDF das páginas
pode ser controlado via código pelo [Puppeteer](#)



PhantomJS

bem conhecido, mas está caindo em desuso
ex. de uso:

```
var page = require( 'webpage' ).create();
page.open( 'http://site.com, function (status) {
  // Página carregada!
  phantom.exit();
} );
```



SlimerJS

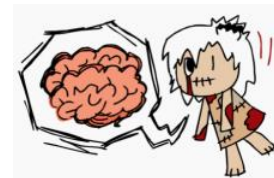
pode operar o Firefox (ou outro baseado na engine Gecko), com ou sem "cabeça"
ex. de uso:

```
var webpage = require('webpage').create();
webpage
  .open( 'http://site.com' )
  .then(function(){
    // Página carregada!
    slimer.exit()
  });
```



ZombieJS

extremamente rápido, apesar do nome ☺



veja mais opções nesse catálogo:

<https://github.com/dhamaniasad/HeadlessBrowsers>

drivers

é uma biblioteca de código capaz de operar um navegador

geralmente segue o padrão W3C *WebDriver*

permite a diferentes *frameworks* o controle de um navegador

é uma opção às bibliotecas de controle nativo

como [Puppeteer](#), [SlimerJS](#), etc.

algumas opções

drivers para o [Selenium Server](#)

[Chrome Driver](#)

[Firefox Driver](#)

[Internet Explorer Driver](#) (!)

instaláveis via `npm install -g selenium-standalone`

podem ser usados pelo [Selenium](#), [CodeceptJS](#) e vários outros frameworks
JavaScript, Python, Ruby, Java, C#, Haskell, Objective-C, Perl, PHP, R

outros: [Geb](#) para Groovy, [Watir](#) para Ruby

veja mais opções nesse catálogo:

<https://github.com/dhamaniasad/HeadlessBrowsers>



perguntas