Testes Funcionais de Software para Empresas Ágeis

curso gratuito

frameworks

PROF. THIAGO DELGADO PINTO

thiago_dp (at) yahoo (dot) com (dot) br

versão: 2018.05.17

CodeceptJS

visão geral

permite testes síncronos, mais fáceis de escrever

perspectiva do usuário

eu ...

independente de *backend*

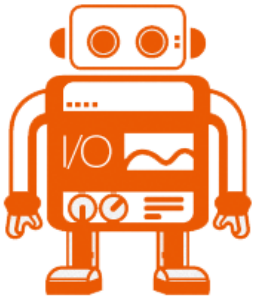
suporta padrão PageObjects

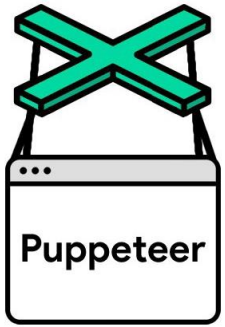
escrito em ES6

integrações

unirest

Lightweight HTTP Request Client Libraries





one framework to rule them all...

- aplicações web
  - WebDriverIO
  - Protractor
  - Puppeteer
- aplicações mobile nativas e baseadas em web
  - Appium
- aplicações desktop
  - Appium
- APIs
  - Unirest (REST) + Rosie (data generator)

instalação

```
# CodeceptJS
npm install -g codeceptjs
# WebDriverIO
npm install –g webdriverio
# Selenium para usar o WebDriverIO no CodeceptJS
npm install selenium-standalone
# Drivers do Selenium
echo "node_modules/.bin/selenium-standalone.cmd" %1 %2
>selenium.cmd"
selenium install
```


Prof. Thiago Delgado Pinto

configuração

Configuração passo-a-passo

```
codeceptjs init
```

```
  Welcome to CodeceptJS initialization tool
  It will prepare and configure a test environment for you

Installing to C:\code\tmp\codecept-js-test
? Where are your tests located? ./*_test.js
? What helpers do you want to use? WebDriverIO
? Where should logs, screenshots, and reports to be stored? ./output
? Would you like to extend I object with custom steps? No
? Do you want to choose localization for tests? English (no localization)
Configure helpers...
? [WebDriverIO] Base url of site to be tested http://localhost
? [WebDriverIO] Browser in which testing will be performed chrome
Config created at C:\code\tmp\codecept-js-test\codecept.json
Directory for temporary output files created at `_output`
Almost done! Create your first test by executing `codeceptjs gt` (generate test) command
```

servidor de testes

```
# Inicia o Selenium Server
# Mais tarde, use Ctrl + C para fechá-lo
selenium-standalone start
```

criando um teste

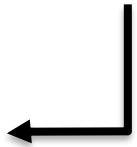
```
# Esqueleto de teste
```

```
codeceptjs gt
```

```
Creating a new test...
----------------------
? Filename of a test logout
? Feature which is being tested Logout
Test for logout was created in C:\code\tmp\codecept-js-test\logout_test.js
```

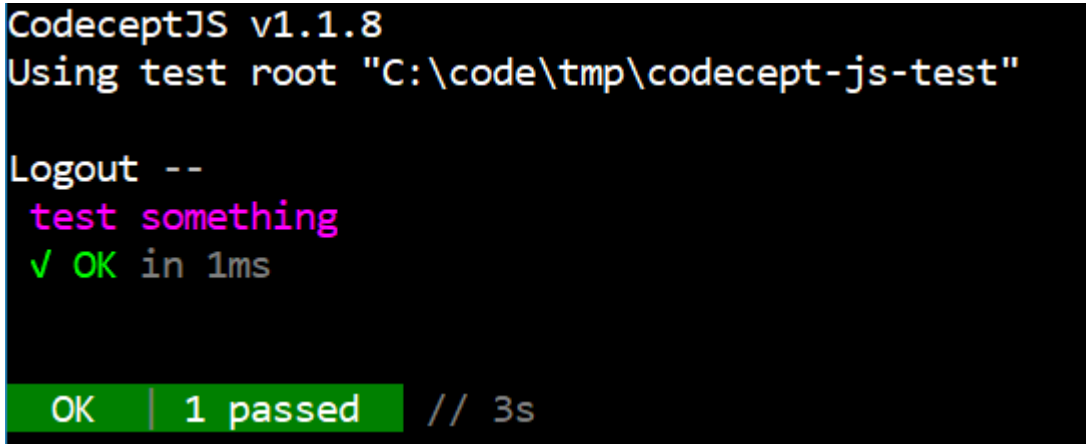
JS logout_test.js

```
1
2   Feature('Logout');
3
4   Scenario('test something', (I) => {
5
6   });
7
```



executando

```
# Para executar os testes exibindo os passos
codeceptjs run --steps
```



API básica

funcionalidade e cenários

```
// Funcionalidade - única por arquivo de teste.
Feature('Minha Funcionalidade');

// Cenário - ao menos um por arquivo de teste.
Scenario('Cenário 1', function(I) {
  I.see('Exemplo');
} );

// Cenário com opções
Scenario('Cenário 2', { browser: 'firefox' }, function(I) {
  I.see( 'Exemplo 2' );
} );

// Cenário assíncrono (ES6)
Scenario('Cenário 3', async function(I) {
  let title = await I.grabTitle();
  let assert = require('assert');
  assert.equal(title, 'CodeceptJS');
} );
```

eventos - antes ou depois de cada cenário

```
Feature('Minha Funcionalidade');

// Executa antes de cada Cenário
Before( function(I) { // Ou Background
  I.amOnPage('/pagina');
} );

// Executa depois de cada Cenário
After( function(I) {
  I.dontSeeInCurrentUrl('/pagina');
} );

...
```

eventos - uma vez, antes ou depois de todos

não tem acesso ao browser
geralmente usados para configuração do ambiente da funcionalidade

```
// uma vez antes de todos os cenários
BeforeSuite( function(I) {
  I.seeFile('dados.json');
});

// uma vez depois de todos os cenários
AfterSuite( function (I) {
  I.writeFile('dados.json',
    [ { nome: 'Ana', idade: 27 }, { nome: 'Bob', idade: 19 } ]
  );
});
```

API para WebDriverIO

alfabeticamente

_locate
_locateCheckable
_locateClickable
_locateFields
acceptPopup
amOnPage
appendField
attachFile
cancelPopup
checkOption
clearCookie
clearField
click
closeCurrentTab
closeOtherTabs
defineTimeout
dontSee
dontSeeCheckboxIsChecked
dontSeeCookie
dontSeeCurrentUrlEquals
dontSeeElement
dontSeeElementInDOM
dontSeeInCurrentUrl
dontSeeInField
dontSeeInSource
dontSeeInTitle
doubleClick
dragAndDrop
executeAsyncScript
executeScript
fillField
grabAttributeFrom
grabBrowserLogs
grabCookie
grabCssPropertyFrom
grabCurrentUrl
grabHTMLFrom
grabNumberOfOpenTabs
grabNumberOfVisibleElements
grabPageScrollPosition
grabPopupText
grabSource
grabTextFrom
grabTitle
grabValueFrom
moveCursorTo
openNewTab
pressKey
refreshPage
resizeWindow
rightClick
runInWeb
runOnAndroid
runOnIOS
saveScreenshot
scrollPageToBottom
scrollPageToTop
scrollTo
see
seeAttributesOnElements
seeCheckboxIsChecked
seeCookie
seeCssPropertiesOnElements
seeCurrentUrlEquals
seeElement
seeElementInDOM
seeInCurrentUrl
seeInField
seeInPopup
seeInSource
seeInTitle
seeNumberOfElements
seeNumberOfVisibleElements
seeTextEquals
seeTitleEquals
selectOption
setCookie
switchTo
switchToNextTab
switchToPreviousTab
uncheckOption
wait
waitForDetached
waitForElement
waitForEnabled
waitForInvisible
waitForText
waitForValue
waitForVisible
waitInUrl
waitNumberOfVisibleElements
waitToHide
waitUntil
waitUrlEquals
locator

amOnPage

abre uma página web em um navegador

requer uma url relativa ou absoluta

se a url começa com barra, a concatena à definida na configuração

```
I.amOnPage('/'); // abre a página principal do website
I.amOnPage('https://github.com'); // abre o github
I.amOnPage('/login'); // abre uma página de login
```

fillField

preenche um *input* ou uma *textarea*

campo pode ser localizado por nome, label, CSS ou XPath

```
// pelo label
I.fillField('Email', 'hello@world.com');
// pelo nome
I.fillField('password', '123456');
// por CSS
I.fillField('form#login input[name=username]', 'John');
// por XPath
I.fillField('//form input[name=username]', 'John');
```

click

```
// link
I.click('Logout');
// botão de formulário
I.click('Submit');
// botão CSS
I.click('#form input[type=submit]');
// XPath
I.click('//form/*[@type=submit]');
// link em um contexto
I.click('Logout', '#nav');
// localizador estrito
I.click({css: 'nav a.login'});
```


Prof. Thiago Delgado Pinto

pressKey

pressiona uma tecla no elemento com o foco

teclas especiais são substituídas pelos respectivos códigos

como "Enter", "Control", "Shift", "Alt", "Command"

array indica uma combinação de teclas

```
I.pressKey('Enter');
I.pressKey(['Control', 'a']);
```


Prof. Thiago Delgado Pinto

checkOption / uncheckOption

seleciona / deseleciona um *checkbox* ou um *radio button*

```
I.checkOption('#concordo');
I.checkOption('Concordo com os Termos e Condições');
I.checkOption('concordo', '//form');

I.uncheckOption('#concordo');
```

dragAndDrop

arrasta um item para um destino

```
I.dragAndDrop('#origem', '#destino');
```

moveCursorTo

move o curso para um elemento

deslocamento extra (X e Y) pode ser informado

```
I.moveCursorTo('.tooltip');
I.moveCursorTo('#mapa', 5, 5);
```

selectOption

seleciona uma ou mais opções em um *select*

se segundo argumento for um *array*, seleciona seus itens

```
I.selectOption('Plano', 'Mensal'); // pelo label
I.selectOption('plano', 'Mensal'); // pelo texto da opção
I.selectOption('plano', '0'); // pelo valor
I.selectOption('form select[name=conta]', 'Premium'); // CSS
I.selectOption('//form/select[@name=conta]', 'Premium'); // XPath
I.selectOption({css: 'form select[name=conta]'}, 'Premium');

I.selectOption('Qual S.O. você usa?', ['Android', 'iOS']);
```

see

é uma assertiva (oráculo)

verifica se a página possui o texto informado

se informado, segundo parâmetro refina a pesquisa

```
I.see('Bem-vindo');
I.see('Bem-vindo', '.conteudo'); // texto dentro da div conteudo
I.see('Cadastrar', {css: 'form.cadastro'}); // localizador estrito
```

seeElement

é uma assertiva (oráculo)

verifica se o elemento informado está visível

permite CSS, XPath ou localizador estrito

```
I.seeElement('#modal');
```

seeInField

é uma assertiva (oráculo)

verifica se um campo contém o texto informado

localiza pelo texto do label, nome, CSS e XPath

```
I.seeInField('Nome', 'Ana');
I.seeInField({css: 'form textarea'}, 'Digite seu comentário');
I.seeInField('form input[type=hidden]', '10');
I.seeInField('#formPesquisa input', 'Pesquisa');
```

seeInCurrentUrl

é uma assertiva (oráculo)

verifica se a url atual contém o fragmento informado

```
I.seeInCurrentUrl('/cadastro');
```

seeInTitle

é uma assertiva (oráculo)

verifica se o título da página possui o texto informado

```
I.seeInTitle('Sistema v1.0');
```

seeCssPropertiesOnElements

é uma assertiva (oráculo)

verifica se todos os elementos com o localizador fornecido contêm as propriedades css indicadas

```
I.seeCssPropertiesOnElements('h3', { 'font-weight': 'bold'});
```

não vejo...

```
I.dontSee('Login');
I.dontSeeCheckboxIsChecked('#concordo');
I.dontSeeCurrentUrlEquals('http://localhost/app/login');
I.dontSeeInCurrentUrl('/login');
I.dontSeeElement('#foto');
I.dontSeeInField('#nome', 'Ana');
I.dontSeeInTitle('Planos');
```


Prof. Thiago Delgado Pinto

saveScreenshot

salva uma foto da tela para o diretório de saída

ex.: *output* (ver codecept.json)

caminho é relativo à esse diretório

```
I.saveScreenshot('home.png');
```

wait

aguarda um certo número de segundos

```
I.wait( 2 ); // 2 segundos
```

waitForElement

espera por um elemento estar presente na página

por default, espera 1 segundo

tempo pode ser informado

```
I.waitForElement('.btn.continuar');
I.waitForElement('.btn.continuar', 5); // 5 segundos
```

alguns outros tipos de wait

```
// espera não estar no DOM
I.waitForDetached('#popup');
// espera estar invisível ou ser removido
I.waitForInvisible('#carregando');
// espera estar visível
I.waitForVisible('#carregando');
// espera estar habilitado
I.waitForEnabled('#salvar');
// espera estar oculto
I.waitForHide('#novo');
// espera fragmento na url
I.waitInUrl('/editar', 2);
```


Prof. Thiago Delgado Pinto

métodos sobre cookies

```
// verifica se existe
I.seeCookie('logado');
// verifica se não existe
I.dontSeeCookie('logado');
// define
I.setCookie({name: 'logado', value: true});
```

listagem de comandos pelo console

```
codeceptjs list
```

exercícios

sistema de exemplo

vamos usar um sistema obtido da web → *mapos*

https://github.com/RamonSilva20/mapos

ordens de serviço - feito em PHP + MySQL

baixe-o de https://github.com/thiagodp/ctf

ajustes já foram feitos para agilizar a configuração

veja pasta **`apps`**

extraia-o em **`C:\dev\wamp\www`**

irá criar pasta **`mapos`**

sistema de exemplo – configuração

acesse http://127.0.0.1/phpmyadmin

entre com **root** e **senha vazia**

criando o banco de dados

1. clique "**Bancos de Dados**"
2. digite "**mapos**" no campo "**Nome da base de dados**"
3. escolha "**utf8_unicode_ci**" no campo "**Agrupamento (Collation)**"
4. clique em "**Criar**"

importando estrutura e dados

1. clique no banco de dados "**mapos**" na lista à esquerda
2. clique em "**Importar**"
3. escolha o arquivo **`C:\dev\wamp\www\mapos\banco.sql`**
4. clique em "**Executar**"

sistema de exemplo – acesso

acesse http://127.0.0.1/mapos/

entre com usuário "**admin@admin.com**" e senha "**123456**"

exercício guiado

Exercício **0** – *login*

→ vamos fazer juntos

criaremos a especificação em **features/login.feature**

criaremos testes em **tests/login_test.js**

passo-a-passo

1. acesse a pasta do **mapos** pelo console
   ex.: tecle **Windows + R** e digite **cmd /k cd C:\dev\wamp\www\mapos**
2. abra o Visual Studio Code (VS Code)
   ex.: digite **code .**
3. crie uma pasta **features** e nela o arquivo **login.feature**

login.feature

Funcionalidade: Login

Cenário: Administrador acessa com sucesso

Dado que tento acessar o sistema

e vejo a tela de login

Quando entro com as credenciais de administrador

Então consigo acesso à tela principal

Prof. Thiago Delgado Pinto

configurando o codeceptjs

```
C:\dev\wamp\www\mapos>codeceptjs init

  Welcome to CodeceptJS initialization tool
  It will prepare and configure a test environment for you

Installing to C:\dev\wamp\www\mapos
? Where are your tests located? tests
? What helpers do you want to use? WebDriverIO
? Where should logs, screenshots, and reports to be stored? ./output
? Would you like to extend I object with custom steps? No
? Do you want to choose localization for tests? English (no localization)
Adding default test mask: tests/*_test.js
Configure helpers...
? [WebDriverIO] Base url of site to be tested http://127.0.0.1/mapos
? [WebDriverIO] Browser in which testing will be performed chrome
Config created at C:\dev\wamp\www\mapos\codecept.json
Directory for temporary output files is already created at './output'
Almost done! Create your first test by executing `codeceptjs gt` (generate test) command
```

gerando um esqueleto de teste

```
c:\dev\wamp\www\mapos>codeceptjs gt
Creating a new test...
--------------------
? Filename of a test login
? Feature which is being tested Login
Test for login was created in c:\dev\wamp\www\mapos\tests\login_test.js
```

login_test.js

```
Feature('Login');

Scenario('Administrador acessa com sucesso', (I) => {
  I.amOnPage( '' ); // "/" está levando para http://127.0.0.1/
  I.seeInCurrentUrl( '/login' );
  I.fillField( '#email', 'admin@admin.com' );
  I.fillField( 'senha', '123456' );
  I.click( 'Acessar' );
  I.wait( 1 );
  I.dontSeeInCurrentUrl( '/login' );
});
```

executando

```
C:\dev\wamp\www\mapos>codeceptjs run --steps
CodeceptJS v1.1.8
Using test root "C:\dev\wamp\www\mapos"

Login --
 Administrador acessa com sucesso
 * I am on page ""
 * I see in current url "/login"
 * I fill field "#email", "admin@admin.com"
 * I fill field "senha", "123456"
 * I click "Acessar"
 * I wait 1
 * I dont see in current url "/login"
 √ OK in 2131ms


 OK | 1 passed // 4s
```

exercício 1

Especifique um novo cenário em **`login.feature`** no qual um usuário com as credenciais incorretas não consegue entrar no sistema.

Então, crie o teste correspondente em **`login_test.js`**. O teste deve verificar se as respectivas mensagens de verificação são exibidas pelo sistema.

exercício 2

Especifique uma nova funcionalidade "Logout", em `logout.feature`, no qual um usuário consegue sair do sistema.

Analise o sistema **mapos** e então crie o teste correspondente em `logout_test.js`.

exercício 3

Fazendo engenharia reversa, analise o sistema **mapos** e especifique uma funcionalidade "**Cadastro de Serviço**", em **`servico-cadastro.feature`**, contendo ao menos um cenário.

Então, crie os respectivos testes em **`servico-cadastro_test.js`**.

exercício 4

a) Especifique a funcionalidade **Alteração de Serviço**, em `servico-alteracao.feature`, e crie cenários e testes.

b) Especifique a funcionalidade **Exclusão de Serviço**, em `servico-exclusao.feature`, e crie cenários e testes.

Page Object

objeto de página (ou tela)

visa **evitar duplicação de código** entre diferentes testes

um objeto passa a **representar** uma **página/tela**

exemplo: representar a tela de login, para facilitar seu reuso

no **codeceptjs**

- **cria-se** um arquivo JavaScript com um **objeto exportável**
- **indica** o objeto no arquivo de configuração, **`codecept.json`**, em **`include`**
- passa o objeto como **argumento para outros cenários**

estrutura

```
'use strict';

let I;
module.exports = {
  _init() {
    I = actor();
  },
  // métodos aqui
};
```


Prof. Thiago Delgado Pinto

exemplo de objeto de página

```
// login_po.js
'use strict';
let I;
module.exports = {
  _init() {
    I = actor();
  },
  logar( email, senha ) {
    I.amOnPage( '/' );
    I.seeInCurrentUrl( '/login' );
    I.fillField( '#email', email );
    I.fillField( 'senha', senha );
    I.click( 'Acessar' );
    I.wait( 1 );
    I.dontSeeInCurrentUrl( '/login' );
  },
  logarComoAdministrador() {
    this.logar( 'admin@admin.com', '123456' );
  }
};
```

```
// codecept.json
...
"include": {
  "login": "./tests/login_po.js"
},
...
```


Prof. Thiago Delgado Pinto

exemplo de uso

```
Feature( 'Cadastro de Serviço' );

Scenario( 'Cadastro com sucesso', ( I, login ) => {
  login.logarComoAdministrador();
  I.amOnPage( '/servicos/adicionar' );
  I.fillField( '#nome', 'Alinhamento' );
  I.fillField( '#preco', '100.00' );
  I.click( 'button[type="submit"]' );
  I.wait( 1 );
  I.dontSeeInCurrentUrl( '/adicionar' );
} );
```

exercício 5

Crie um *Page Object* de **Login** e o use em ao menos um dos testes relacionados a **Serviço**.

exercício 6

a) Especifique uma funcionalidade **Ordem de Serviço** com um cenário básico, em **`os.feature`**. Crie um teste para esse cenário, em **`os_test.js`**.

Então, especifique e crie o teste dos seguintes cenários adicionais:

b) visualizar a OS

c) adicionar um Produto à OS

d) adicionar um Serviço à OS

e) adicionar um Anexo à OS

f) faturar a OS

testes orientados a dados

data-driven tests

úteis testar um **mesmo cenário** com **dados diferentes**

```
let contasUsuario = new DataTable(['usuario', 'senha']);
contasUsuario.add(['admin', '123456']);
contasUsuario.add(['ana', '654321']);
contasUsuario.add(['bob', 'b0bp4s$']);

// Passe a tabela para o método Data() e invoque o cenário por ele
// Use o parâmetro reservado "current" para obter a tabela
Data( contasUsuario )
.Scenario( 'Acessa o sistema com sucesso', (I, current) => {
  I.fillField('#usuario', current.usuario);
  I.fillField('#senha', current.senha);
  I.click('#entrar');
  I.see('Bem-vindo '+ current.usuario);
});
```

exercício 7

Cadastre no sistema alguns usuários novos e adicione à `login_test.js` um teste orientado a dado.

alguns outros recursos úteis

injeção de objetos nos testes

```
Scenario('exemplo', (I, cfg) => {
  I.fillField('#usuario', cfg.usuarioAdministrador);
  I.fillField('#senha', cfg.senhaAdministrador);
  I.pressKey('Enter');
}).injectDependencies({ cfg: require('./config.js') });

Scenario('exemplo 2', (I, rotas) => {
  I.amOnPage( rotas.perfil );
  I.see( 'Perfil' );
}).injectDependencies({ rotas: require('./rotas.js') });
```


Prof. Thiago Delgado Pinto

usando ator para adicionar recursos em "I"

é possível adicionar métodos ao objeto **I**

bom para coisas repetitivas/compartilhadas, como login

Em **codecept.json**:

```
"include": {
    "I": "./tests/extensao-i.js"
},
```

Em **extensao-i.js**:

```
module.exports = function() {
  return actor({
    login: function(email, senha) {
      this.fillField('#email', email);
      this.fillField('#senha', senha);
      this.click('#entrar');
    }
  });
};
```

referências

CodeceptJS. **WebDriverIO – CodeceptJS**. Disponível em: https://codecept.io/helpers/WebDriverIO/. Acesso em Maio de 2018.