

CURSO DE TESTES AUTOMATIZADOS SELENIUM WEBDRIVER

Wagner Costa

wcaquino@gmail.com

ERA UMA VEZ...

- Jason Huggins (ThoughtWorks) – 2004

- Selenium Core
 - Selenium IDE
 - Selenium RC



- Simon Stewart (Google) – 2006

- WebDriver

- 2009

- Selenium WebDriver

- Futuro...

- Selenium 3



BROWSERS SUPORTADOS

Browser	Selenium IDE	Selenium Remote Control	Selenium Core	Selenium 2/Webdriver API
Firefox 10	Record and playback tests	Start browser, run tests	Run tests	Run tests
Firefox 9	Record and playback tests	Start browser, run tests	Run tests	Run tests
Firefox 8	Record and playback tests	Start browser, run tests	Run tests	Run tests
Firefox 7	Record and playback tests	Start browser, run tests	Run tests	Run tests
Firefox 6	Record and playback tests	Start browser, run tests	Run tests	Run tests
Firefox 5	Record and playback tests	Start browser, run tests	Run tests	Run tests
Firefox 4	Record and playback tests	Start browser, run tests	Run tests	Run tests
Firefox 3.6	Record and playback tests	Start browser, run tests	Run tests	Run tests
Firefox 3	Record and playback tests	Start browser, run tests	Run tests	not supported
IE 9	not supported	Start browser, run tests	Run tests	Run tests
IE 8	not supported	Start browser, run tests	Run tests	Run tests
IE 7	not supported	Start browser, run tests	Run tests	Run tests
Safari 3	not supported	Start browser, run tests	Run tests	not supported
Safari 2	not supported	Start browser, run tests	Run tests	not supported
Opera 9	not supported	Start browser, run tests	Run tests	Run tests
Opera 8	not supported	Start browser, run tests	Run tests	Run tests
Chrome	not supported	Start browser, run tests***	Run tests***	Run tests***
Others	not supported	Partial support possible*	Run tests**	not supported

[* Selenium Remote Control server can start any executable, but depending on browser security settings, there may be technical limitations that would limit certain features.]

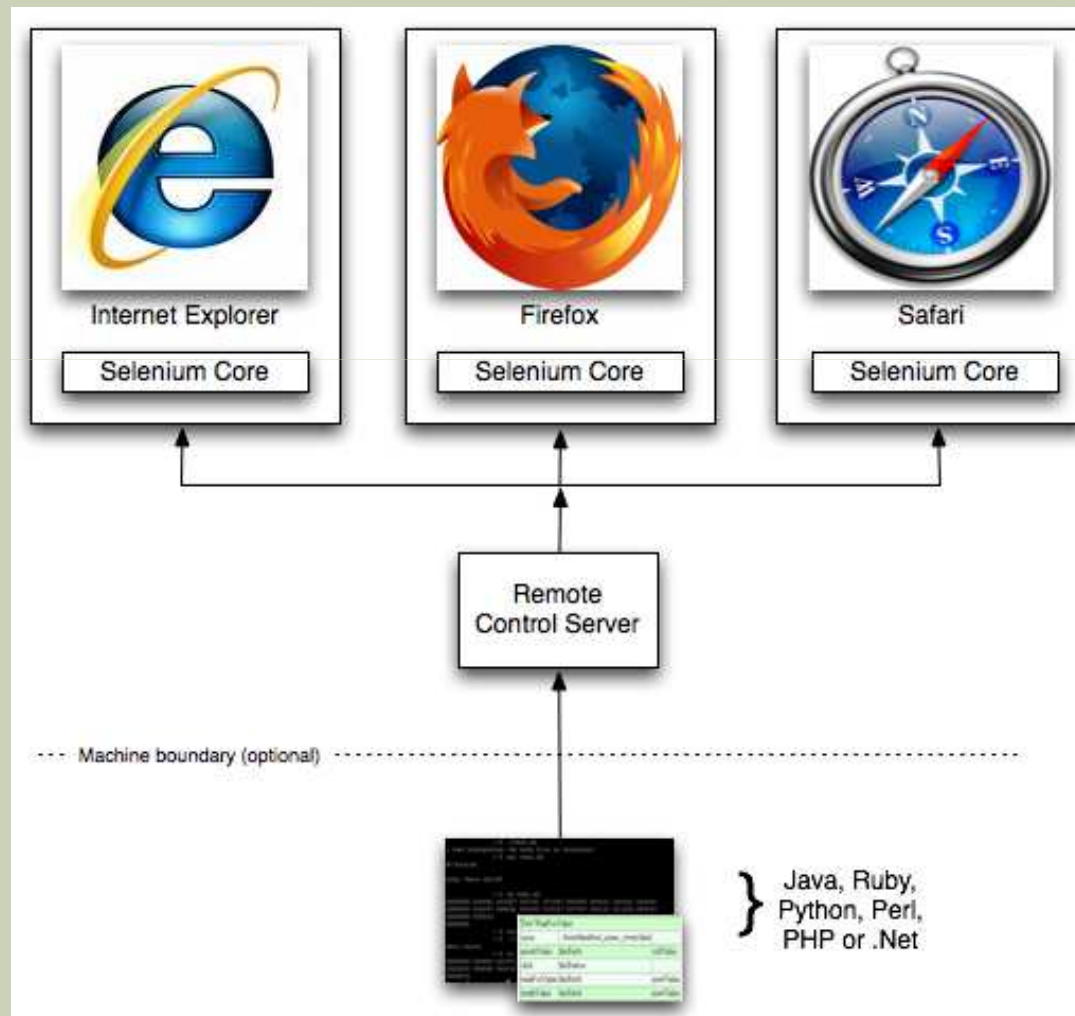
[** Selenium Core is written in Javascript so it should work (in theory) on most browsers with decent Javascript support. Some operations are only possible via Selenium Remote Control because of browser security settings.]

[*** Chrome is supported in collaboration with the Chromium team. See [this website](#) for more information on Chromedriver.]

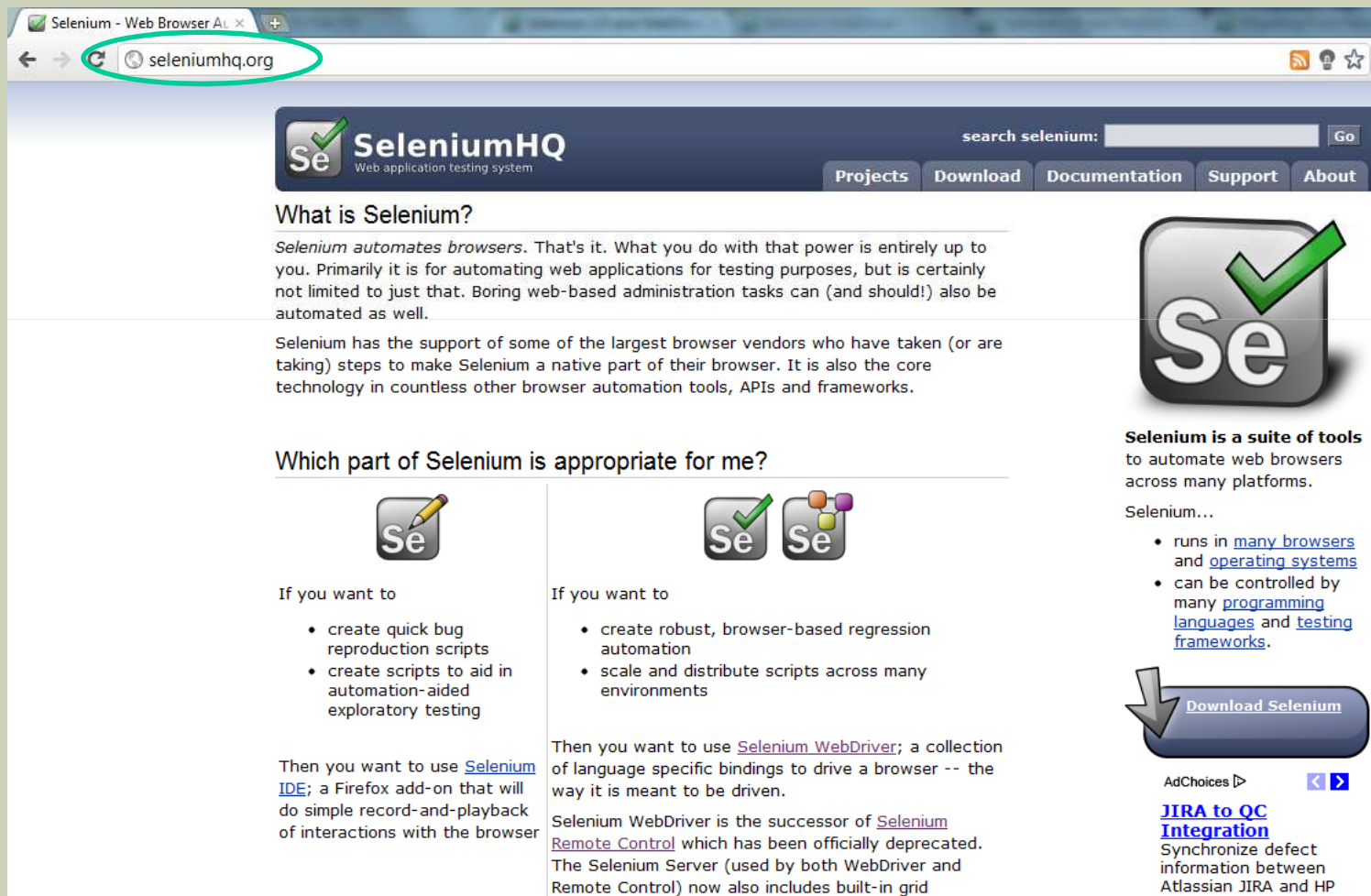
LINGUAGENS SUPORTADAS

- Java
- Ruby
- Python
- C#
- PHP*
- Perl*

FUNCIONAMENTO - RC



ONDE BAIXAR?



Selenium - Web Browser AL x

← → ↻ seleniumhq.org

SeleniumHQ
Web application testing system

search selenium: Go


Projects Download Documentation Support About

What is Selenium?

Selenium automates browsers. That's it. What you do with that power is entirely up to you. Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should!) also be automated as well.

Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks.



Which part of Selenium is appropriate for me?



If you want to

- create quick bug reproduction scripts
- create scripts to aid in automation-aided exploratory testing

Then you want to use [Selenium IDE](#); a Firefox add-on that will do simple record-and-playback of interactions with the browser




If you want to

- create robust, browser-based regression automation
- scale and distribute scripts across many environments

Then you want to use [Selenium WebDriver](#); a collection of language specific bindings to drive a browser -- the way it is meant to be driven.

Selenium WebDriver is the successor of [Selenium Remote Control](#) which has been officially deprecated. The Selenium Server (used by both WebDriver and Remote Control) now also includes built-in grid




Selenium is a suite of tools to automate web browsers across many platforms.

Selenium...

- runs in [many browsers](#) and [operating systems](#)
- can be controlled by many [programming languages](#) and [testing frameworks](#).

Download Selenium

AdChoices 

[JIRA to QC Integration](#)
Synchronize defect information between Atlassian JIRA and HP

COMO INSTALAR?

- Adicionar a lib do selenium (e suas dependências) no classpath do projeto

- Para o Selenium RC

- Iniciar o Selenium Server

- `java -jar selenium-server<versao>.jar -singleWindow -port 4444`

- Para o Selenium 2

- NADA

INICIANDO SELENIUM

```
private static Selenium selenium;
```



Selenium RC

```
@BeforeClass
```

```
public static void setUp(){
```

```
    selenium = new DefaultSelenium("localhost",  
        4444,
```

```
        "*chrome c:\\Program Files (x86)\\Mozilla Firefox\\firefox.exe",  
        "http://localhost:8888");
```

```
    selenium.start();
```

```
}
```

```
private static WebDriver driver;
```



Selenium WebDriver

```
@BeforeClass
```

```
public static void setUp(){
```

```
    driver = new FirefoxDriver();
```

```
    driver.get("http://localhost:8888");
```

```
}
```


ATENÇÃO

- Algumas operações estão no Selenium RC, outras no Selenium WebDriver. Para não haver confusão considere a variável de acesso às páginas:
 - *selenium* = Selenium RC
 - *driver* = Selenium WebDriver
- Forma para conversão da sintaxe:

```
private static WebDriver driver;  
private static Selenium selenium;  
  
@BeforeClass  
public static void setUp(){  
    driver = new FirefoxDriver();  
    selenium = new WebDriverBackedSelenium(driver, baseUrl);  
}
```

```
WebDriver driverInstance = ((WebDriverBackedSelenium) selenium).getWrappedDriver();
```

API

- **Pode ser encontrada em:**

- `http://selenium.googlecode.com/svn/trunk/docs/api/java/index.html`

- **Selenium RC :** `com.thoughtworks.selenium`

- **Selenium WebDriver:** `org.openqa.selenium`

PRINCIPAIS COMANDOS

- **findByElement(By)**
 - Retorna uma instância de WebElement localizada através do By
- **sendKeys(String)**
 - Envia valor aos campos input
- **Clean()**
 - Limpa campos input
- **Click()**
 - Clica em um valor encontrado
- **getText()**
 - Retorna o texto do campo
- **isEnabled()**
 - Indica se o campo está habilitado

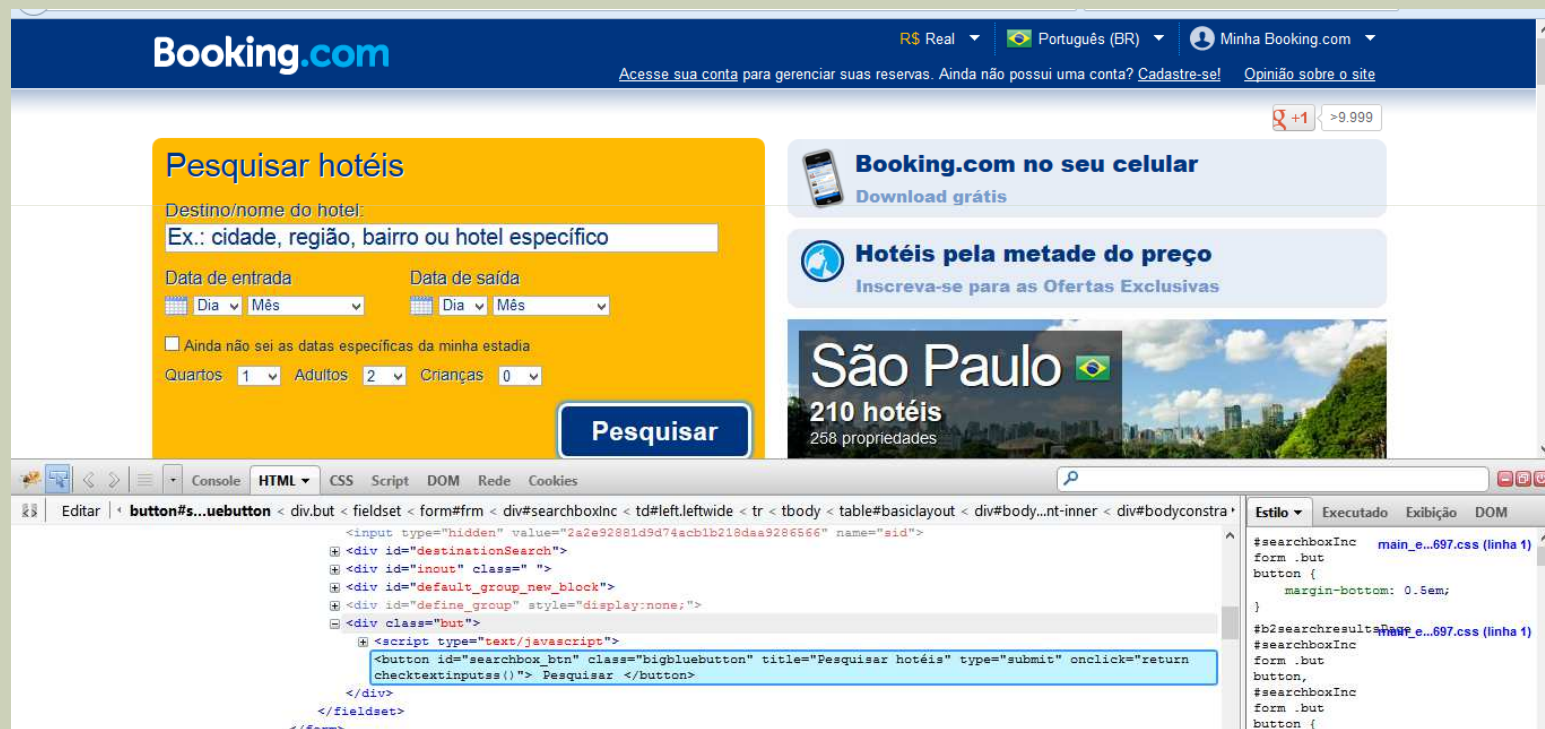
BY

- Interface utilizada para localizar elementos no documento.

```
●S className(String className) : By - By  
●S cssSelector(String selector) : By - By  
●S id(String id) : By - By  
●S linkText(String linkText) : By - By  
●S name(String name) : By - By  
●S partialLinkText(String linkText) : By - By  
●S tagName(String name) : By - By  
●S xpath(String xpathExpression) : By - By
```

FIREBUG

- Extensão do Firefox (e outros browsers)
- Ajuda na descoberta dos atributos dos elementos da página



- F12 para ativá-lo

CLICAR EM ELEMENTOS

- Clicar em algum elemento da tela

```
protected void clicarElemento(By by){  
    driver.findElement(by).click();  
}
```

```
protected void clicarLink(String link){  
    clicarElemento(By.linkText(link));  
}
```

```
protected void clicarBotao(String value){  
    clicarElemento(By.xpath("//input[@value='" + value + "']"));  
}
```

OBTER TEXTO DE UM ELEMENTO

- Recupera o texto em um determinado componente

```
driver.findElement(  
    By.xpath("//div[@id='conteudo_pagina']/table/tbody/tr[3]/td/table/tbody/tr[8]/td[2]")).getText();
```

IS TEXT PRESENT

- Verifica se determinado texto está presente na página

```
selenium.isTextPresent("texto");
```

```
public void verificarTextosPresentes(String... textos){  
    for(String texto: textos){  
        Assert.assertTrue("Valor: '" + texto + "' não encontrado",  
                           selenium.isTextPresent(texto));  
    }  
}
```

```
driver.findElement(By.tagName("body")).getText().contains("texto");
```

```
public void verificarTextosPresentes(String... textos){  
    String pagina = driver.findElement(By.tagName("body")).getText();  
    for(String texto: textos){  
        Assert.assertTrue("Valor: '" + texto + "' não encontrado",  
                           pagina.contains(texto));  
    }  
}
```


SELECT

- Selecionar opção em combo

```
protected void seleccionarCombo(By by, String valor){  
    Select combo = new Select(driver.findElement(by));  
    combo.selectByVisibleText(valor);  
}
```

ALERT

- Verificar mensagem em Alert (JS)

```
protected void verificarAlerta(String texto){  
    Alert alert = driver.switchTo().alert();  
    Assert.assertEquals(texto, alert.getText());  
    alert.accept();  
}
```

ESCREVER

- Preenche campos textuais

```
driver.findElement(by).sendKeys(texto);
```

```
protected void escrever(By by, String texto){  
    driver.findElement(by).clear();  
    driver.findElement(by).sendKeys(texto);  
}
```

WAIT

- Aguardar por algum elemento na tela

```
@BeforeClass
public static void setUp(){
    driver = new FirefoxDriver();
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    driver.get("http://localhost:8888");
}
```

WAIT EXPLÍCITO

- Permite definir o tempo máximo (*timeout*) para esperar por algum elemento

```
public static void waitForTextPresent(String texto){
    try {
        int trys = ((int) Integer.valueOf(properties.timeout) / 500);
        while(true) {
            if(selenium.isTextPresent(texto)) {
                break;
            }
            Thread.sleep(500);
            if(trys-- < 0)
                Assert.fail("[ "+texto+" ] Não encontrado");
        }
    } catch (InterruptedException e) {
        Assert.fail("Problemas com o sleep do waitForTextPresent");
    }
}
```

```
Wait<WebDriver> wait = new WebDriverWait(driver, 30);
WebElement element= wait.until(visibilityOfElementLocated(By.id("some_id")));
```

NAVEGAÇÕES

- `Driver.navigate().to(url)`
- `Driver.navigate().forward()`
- `Driver.navigate().back()`

- `Driver.navigate().refresh()`
- `Driver.getCurrentUrl()`

SCREENSHOT

- TakeScreenShot armazena uma imagem da tela do browser
- Utilizado para gerar uma evidência do erro

```
private void takeScreenShot() {  
    WebDriver augmentedDriver = new Augmenter().augment(driver);  
    File screenshot = ((TakesScreenshot) augmentedDriver).  
        getScreenshotAs(OutputType.FILE);  
  
    try {  
        FileUtils.copyFile(screenshot, new File("ss.png"));  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

EXTENSÕES

- Por padrão, o Selenium cria um perfil “limpo” para cada execução

```
@BeforeClass
public static void setUp() throws IOException{
    File firebug = new File("firebug-1.10.6-fx.xpi");
    FirefoxProfile profile = new FirefoxProfile();
    profile.addExtension(firebug);
    driver = new FirefoxDriver(profile);
}
```

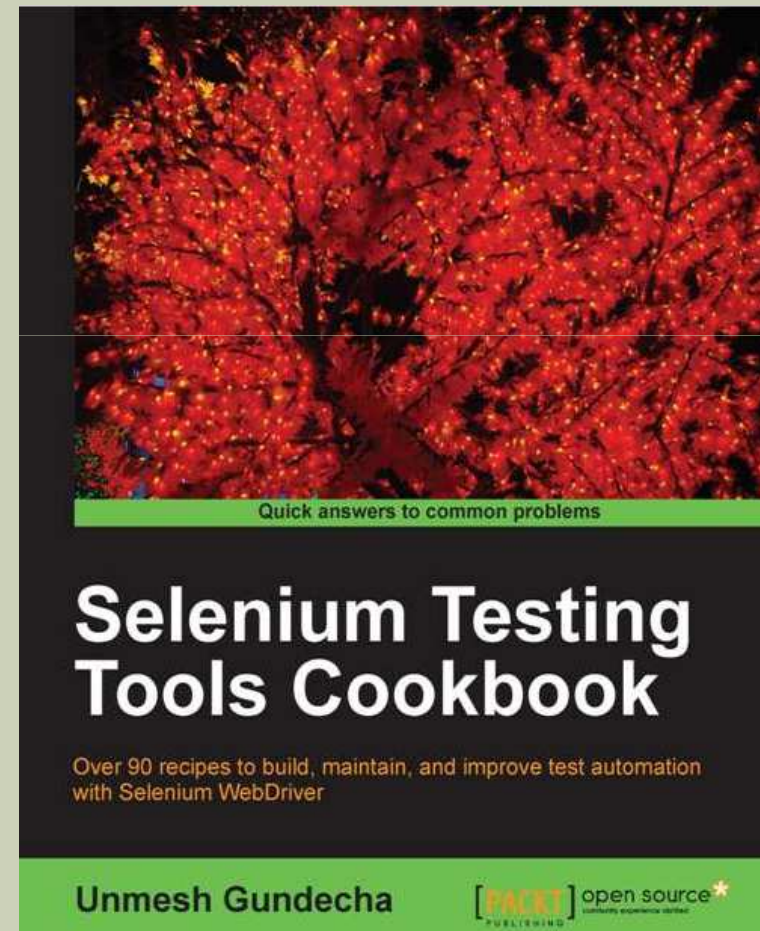
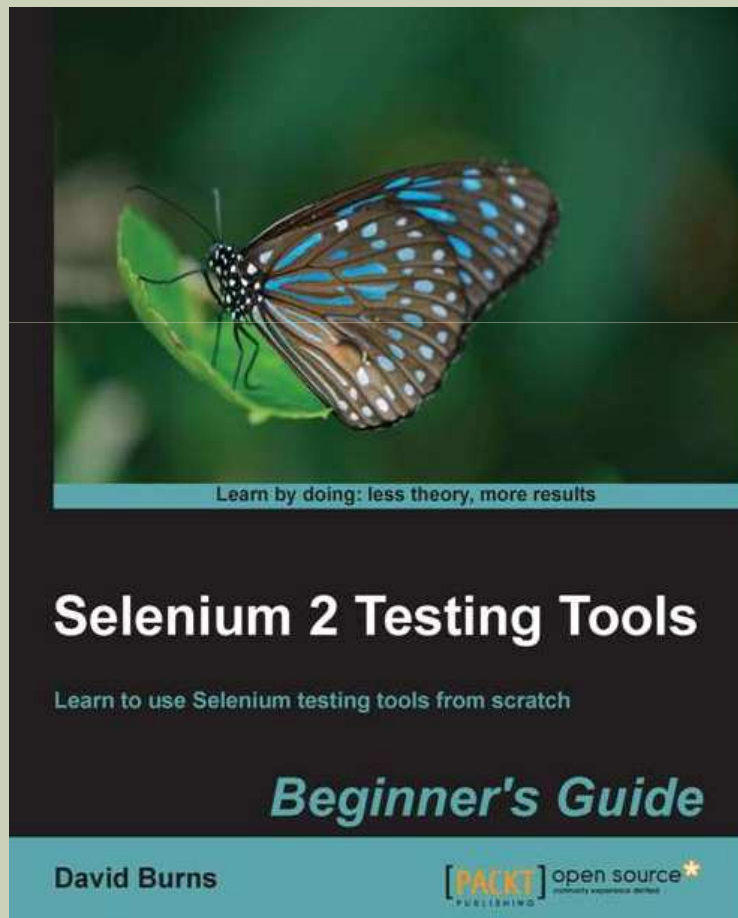

PROXY

```
@BeforeClass
public static void setUp() throws IOException{
    String PROXY = "endereco_proxy:numero_porta";
    org.openqa.selenium.Proxy proxy = new Proxy();
    proxy.setHttpProxy(PROXY).setFtpProxy(PROXY).setSslProxy(PROXY);
    DesiredCapabilities capabilities = new DesiredCapabilities();
    capabilities.setCapability(CapabilityType.PROXY, proxy);
    driver = new FirefoxDriver(capabilities);
}
```

RECOMENDAÇÕES

- **DESENVOLVEDORES:** Coloquem id nos elementos Web.
- Ao entrar em uma página, verificar se está na tela certa
 - Checar elementos, título ou algo que possa identificar a página
- Criar um método por cenário, indicando todos os passos através de comentários (facilita, E MUITO, a rastreabilidade)
 - Desconsiderar caso use BDD
- Eliminar pausas estáticas
- Assista sua bateria executando

MAIS CONTEÚDO...



MÃOS À OBRA

- Criar um projeto novo com o Selenium
- Automatizar no Selenium WebDriver o mesmo roteiro criado no exercício passado.