

# Ferramenta Selenium

# O que é

- A ferramenta Selenium permite que se realizam testes automatizados de aplicativos Web
- Em particular, teste de regressão
- IDE Selenium
- Selenium WebDriver
- <http://www.seleniumhq.org/>

# Os dois componentes

Which part of Selenium is appropriate for me?

## Selenium WebDriver



If you want to

- create robust, browser-based regression automation suites and tests
- 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 capabilities.

## Selenium IDE



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.

# Vamos começar com IDE

- O IDE permite que você navegue no browser
- E vá registrando todos os passos
  - clicar num link
  - digitar uma url
  - apertar botão back
- Ao finalizar a gravação, os passos podem ser gravados
- E depois reproduzidos

# Instalação

- <http://www.seleniumhq.org/>

Which part of Selenium is appropriate for me?

## Selenium WebDriver



If you want to

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

Then you want to use Selenium WebDriver

## Selenium IDE



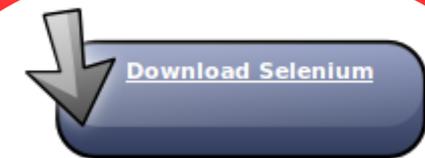
If you want to

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

**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](#).



**Donate to Selenium**

# Instalação

- <http://www.seleniumhq.org/download/>

## Selenium IDE

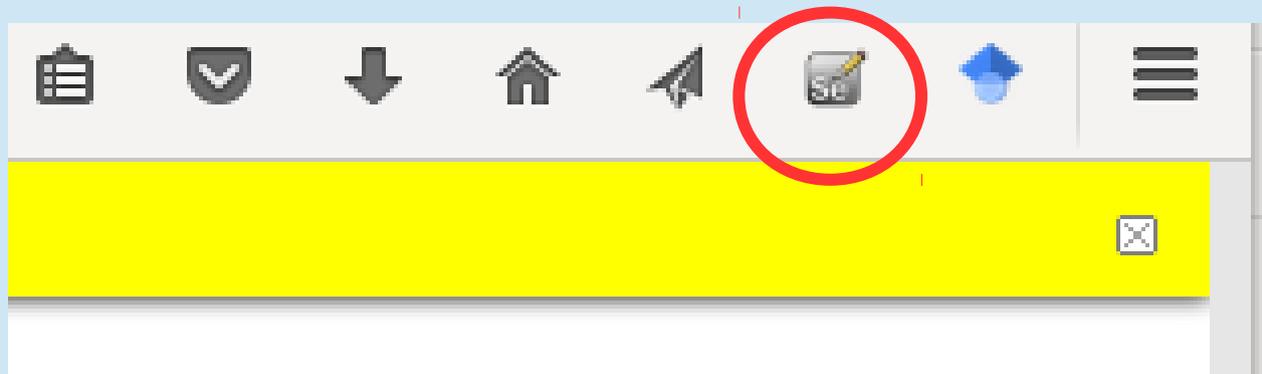
Selenium IDE is a Firefox plugin which records and plays back user interactions with the browser. Use this to either create simple scripts or assist in exploratory testing. It can also export Remote Control or WebDriver scripts, though they tend to be somewhat brittle and should be overhauled into some sort of Page Object-y structure for any kind of resiliency.

Download latest released version [from addons.mozilla.org](http://addons.mozilla.org) or view the [Release Notes](#) and then [install some plugins](#).

Download [previous versions here](#).

# Usando IDE

- Feita a instalação aparece no canto superior direito o icone do IDE



- Clique e veja o que acontece

# Usando IDE – gravar



# Usando IDE – gravar

The screenshot displays the Selenium IDE 2.9.1 interface. At the top, the menu bar includes 'Arquivo (F)', 'Editar', 'Ações', 'Opções', and 'Ajuda'. The 'URL Base' field is set to 'http://icmc.usp.br/'. Below this is a speed control slider from 'Fast' to 'Slow' and several playback buttons (play, stop, refresh, undo). The main area is divided into two tabs: 'Tabela' (active) and 'Código-Fonte'. The 'Tabela' tab shows a table with three columns: 'Comando', 'Alvo', and 'Valor'. Below the table are input fields for 'Comando', 'Alvo', and 'Valor', along with 'Select' and 'Procurar' buttons. On the left, a 'Test Case' pane shows an 'Untitled' case. At the bottom left, a status bar shows 'Runs: 0' and 'Failures: 0'. A message pane at the bottom has tabs for 'Mensagens', 'Reference', 'UI-Element', and 'Rollup'.

Comando	Alvo	Valor

Comando:

Alvo:

Valor:

Runs: 0  
Failures: 0

Mensagens | Reference | UI-Element | Rollup

Esse é o controle do IDE.  
Nele podemos dar comandos para gravar ou reproduzir casos de teste.

# Usando IDE – gravar

The screenshot shows the Selenium IDE 2.9.1 interface. The URL Base is set to `http://icmc.usp.br/`. The interface includes a menu bar (Arquivo (F), Editar, Ações, Opções, Ajuda), a toolbar with playback controls (Fast, Slow, play, pause, stop, refresh), and a main workspace. The workspace is divided into a Test Case pane (Untitled) and a table for recording commands. The table has columns for Comando, Alvo, and Valor. Below the table are input fields for Comando, Alvo, and Valor, along with buttons for Select and Procurar. A red circle button in the top right corner of the workspace is highlighted with a green arrow. A blue callout box contains the text: "Apertando esse botão, tudo o que fizemos no browser vai ser gravado. Até o botão seja apertado novamente."

Comando	Alvo	Valor
---------	------	-------

Comando

Alvo

Valor

Runs: 0  
Failures: 0

Mensagens Reference UI-Element Rollup

# Vamos tentar

- Inicie a gravação
- Digite a url: [www.icmc.usp.br](http://www.icmc.usp.br)
- Clique em “Pessoas”
- Clique em “Docentes”
- Procure o nome “Silva”
- Selecione um professor
- Termine a gravação

# Resultado

Untitled (untitled suite) - Selenium IDE 2.9.1 \*

Arquivo (F) Editar Ações Opções Ajuda

URL Base

Fast Slow    

Test Case

**Procura docente Silva \***

Comando	Alvo	Valor
click	link=Docentes	
type	name=nome_busca	Silva
click	css=button.btn.btn-primary	
clickAndWait	//img[contains(@src,'https://...]	

Comando

Alvo

Valor

Runs: **1**

Failures: **0**

Mensagens Reference UI-Element Rollup

**open(url)**  
Arguments:  
• url - the URL to open; may be relative or absolute  
Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load

# Resultado

The screenshot shows the Selenium IDE 2.9.1 interface. The URL Base is set to `http://www.icmc.usp.br/`. The test case is named "Procura docente Silva \*". The table below lists the commands for this test case:

Comando	Alvo	Valor
click	link=Docentes	
type	name=nome_busca	Silva
click	css=button.btn.btn-primary	
clickAndWait	//img[contains(@src,'https://...]	

Below the table, there are input fields for "Comando", "Alvo", and "Valor", along with "Select" and "Procurar" buttons. The "Runs" counter shows 1 and "Failures" shows 0. The bottom section displays the "open(url)" command with its arguments and description.

Nome, que pode ser alterado

# Resultado

The screenshot shows the Selenium IDE 2.9.1 interface. The URL Base is set to `http://www.icmc.usp.br/`. The test case is named "Procura docente Silva \*". The "Comandos executados" (Executed Commands) table is highlighted with a blue oval and contains the following data:

Comando	Alvo	Valor
click	link=Docentes	
type	name=nome_busca	Silva
click	css=button.btn.btn-primary	
clickAndWait	//img[contains(@src,'https://...]	

Below the table, the "Comando" dropdown is set to "click", the "Alvo" field is empty, and the "Valor" field is empty. The "Select" and "Procurar" buttons are visible. The status bar shows 1 Run and 0 Failures.

Annotations on the image:

- A blue oval around the test case name "Procura docente Silva \*" is accompanied by the text "Nome, que pode ser alterado".
- A blue oval around the "Comandos executados" table is accompanied by the text "Comandos executados".

At the bottom of the interface, the "Reference" tab is active, showing the documentation for the `open(url)` command:

```

open(url)
Arguments:
  • url - the URL to open; may be relative or absolute
Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load
  
```

# Resultado

The screenshot shows the Selenium IDE 2.9.1 interface. The URL Base is set to `http://www.icmc.usp.br/`. The test case is named "Procura docente Silva \*". The "Comandos executados" (Executed Commands) table is highlighted with a blue circle and contains the following data:

Comando	Alvo	Valor
click	link=Docentes	
type	name=nome_busca	Silva
click	css=button.btn.btn-primary	
clickAndWait	//img[contains(@src,'https://...]	

Below the table, the "Comando" dropdown is set to "click", the "Alvo" field is empty, and the "Valor" field is empty. The "Select" and "Procurar" buttons are visible. The status bar shows 1 Run and 0 Failures.

Annotations on the screenshot:

- A blue circle around the test case name "Procura docente Silva \*" is accompanied by the text "Nome, que pode ser alterado".
- A blue circle around the "Comandos executados" table is accompanied by the text "Comandos executados".
- A blue circle around the "Comando" dropdown and the "Alvo" and "Valor" input fields is accompanied by the text "Descrição dos comandos".

At the bottom of the screenshot, the "Reference" tab is selected, showing the description for the `open(url)` command:

**open(url)**  
Arguments:  
• url - the URL to open; may be relative or absolute  
Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load

# Reproduzindo

Esse botão vai fazer com que o browser reproduza a navegação

The screenshot shows the Selenium IDE 2.9.1 interface. The URL Base is set to `http://www.icmc.usp.br/`. A table of commands is visible, and the 'Replay' button (represented by a play icon) is circled in blue. Below the table, there are input fields for 'Comando', 'Alvo', and 'Valor', along with 'Select' and 'Procurar' buttons. At the bottom, there is a 'Reference' section for the `open(url)` command.

Comando	Alvo	Valor
click	link=Docentes	
type	name=nome_busca	Silva
click	css=button.btn.btn-primary	
clickAndWait	//img[contains(@src,'https://...]	

Comando:

Alvo:

Valor:

Runs: 1  
Failures: 0

**open(url)**  
Arguments:  
• url - the URL to open; may be relative or absolute  
Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load

# Reproduzindo

The screenshot shows the Selenium IDE 2.9.1 interface. The URL Base is set to `http://www.icmc.usp.br/`. The test suite contains the following commands:

Comando	Alvo	Valor
click	link=Docentes	
type	name=nome_busca	Silva
click	css=button.btn.btn-primary	
clickAndWait	//img[contains(@src,'https://...]	

Below the table, the 'Comando' field is empty, and the 'Alvo' and 'Valor' fields are also empty. The 'Select' and 'Procurar' buttons are visible. The 'Runs' section shows 1 run and 0 failures. The 'open(uri)' command is highlighted in the bottom section.

**open(uri)**  
Arguments:  
• url - the URL to open; may be relative or absolute  
Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load

Esse botão vai fazer com que o browser reproduza a navegação

Indica se os resultados estão corretos

# O que são resultados corretos

- Por exemplo, no nosso script é possível que o último comando não possa ser executado
- Ao clicar no link “Docente” a página não é alterada
- Ela apenas muda. Mas o script não espera isso acontecer

# Consertando o script

Procura docente Silva (untitled suite) - Selenium IDE 2.9.1 \*

Arquivo (F) Editar Ações Opções Ajuda

URL Base

Fast Slow

Test Case

**Procura docente Silva \***

Comando	Alvo	Valor
clickAndWait	link=Pessoas	
click	link=Docentes	
pause	10000	
type	name=nome_busca	Silva
click	css=button.btn.btn-primary	
pause	10000	

Comando

Alvo

Valor

Runs: 1

Failures: 0

Mensagens Reference UI-Element Rollup

**pause(waitTime)**

Arguments:

- waitTime - the amount of time to sleep (in milliseconds)

Wait for the specified amount of time (in milliseconds)

# Verificando resultados

- Ao criar casos de teste, podemos acrescentar comandos que façam verificações dos resultados que aparecem no browser.
- Ao final da pesquisa, queremos verificar se lá em cima aparece a categoria “Docente”

# Verificando resultados

Procura docente Silva (untitled suite) - Selenium IDE 2.9.1 \*

Arquivo (F) Editar Ações Opções Ajuda

URL Base

Fast Slow

Test Case

**Procura docente Silva \***

Comando	Alvo	Valor
click	css=button.btn.btn-primary	
pause	10000	
clickAndWait	//img[contains(@src,'https://...]	
pause	10000	
assertText	css=h4.media-heading	Docente:

Comando

Alvo

Valor

Runs: 1

Failures: 0

Mensagens Reference UI-Element Rollup

Returns:  
the text of the element  
Gets the text of an element. This works for any element that contains text. This command uses either the textContent (Mozilla-like browsers) or the innerText (IE-like browsers) of the element, which is the rendered text shown to the user.

# Verificando resultados

Procura docente Silva (untitled suite) - Selenium IDE 2.9.1 \*

Arquivo (F) Editar Ações Opções Ajuda

URL Base

Fast Slow

Test Case

**Procura docente Silva \***

Comando	Alvo	Valor
click	css=button.btn.btn-primary	
pause	10000	
clickAndWait	//img[contains(@src,'https://...]	
pause	10000	
assertText	css=h4.media-heading	Docente:

Comando

Alvo

Valor

Runs: 1

Failures: 0

Mensagens Reference UI-Element Rollup

Returns:  
the text of the element  
Gets the text of an element. This works for any element that contains text. This command uses either the textContent (Mozilla-like browsers) or the innerText (IE-like browsers) of the element, which is the rendered text shown to the user.

# Salvando

- O caso de teste pode ser salvo
- Vários casos de teste podem ser salvos salvos como uma “Test suite”
- E casos de teste ou teste suites podem ser exportados
- Arquivo → Exportar teste como → Java/JUnit4/Webdriver

# JUnit

```
public class ProcuraDocenteSilva {  
    private WebDriver driver;  
    private String baseUrl;  
    private boolean acceptNextAlert = true;  
    private StringBuffer verificationErrors = new StringBuffer();  
  
    @Before  
    public void setUp() throws Exception {  
        driver = new FirefoxDriver();  
        baseUrl = "http://www.icmc.usp.br/";  
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);  
    }  
}
```

# JUnit

```
@Test
```

```
public void testProcuraDocenteSilva() throws Exception {  
    driver.get(baseUrl + "/");  
    driver.findElement(By.linkText("Pessoas")).click();  
    driver.findElement(By.linkText("Docentes")).click();  
    driver.findElement(By.name("nome_busca")).clear();  
    driver.findElement(By.name("nome_busca")).sendKeys("Silva");  
    driver.findElement(By.cssSelector("button.btn.btn-primary")).click();  
  
    driver.findElement(By.xpath("//img[contains(@src,'https://web.icmc.usp.br/SCAPINST/fotos_pessoas/5765587.jpg')]")).click();  
  
    assertEquals("Docente:", driver.findElement(By.cssSelector("h4.media-heading")).getText());  
}
```

# JUnit

```
@Test
public void testProcuraDocenteSilva() throws Exception {
    driver.get(baseUrl + "/");
    driver.findElement(By.WebDriver s")).click();
    driver.findElement(By.linkText("Docentes")).click();
    driver.findElement(By.name("nome_busca")).clear();
    driver.findElement(By.name("nome_busca")).sendKeys("Silva");
    driver.findElement(By.cssSelector("button.btn.btn-primary")).click();

    driver.findElement(By.xpath("//img[contains(@src,'https://web.icmc.usp.br/SCAPINST/fotos_pessoas/5765587.jpg')]")).click();

    assertEquals("Docente:", driver.findElement(By.cssSelector("h4.media-heading")).getText());
}
```

# WebDriver

- É uma biblioteca que vai permitir que você controle a execução do browser de dentro de um programa Java
- Ou outra linguagem
- Fazer o download

# WebDriver – download

- <http://www.seleniumhq.org/download/>

## Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on google code.

Language	Client Version	Release Date			
Java	3.4.0	2017-04-21	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">Javadoc</a>
C#	3.4.0	2017-04-21	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Ruby	3.4.0	2017-04-21	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Python	3.4.1	2017-04-27	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Javascript (Node)	3.4.0	2017-04-21	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>

# WebDriver – download

- <http://www.seleniumhq.org/download/>

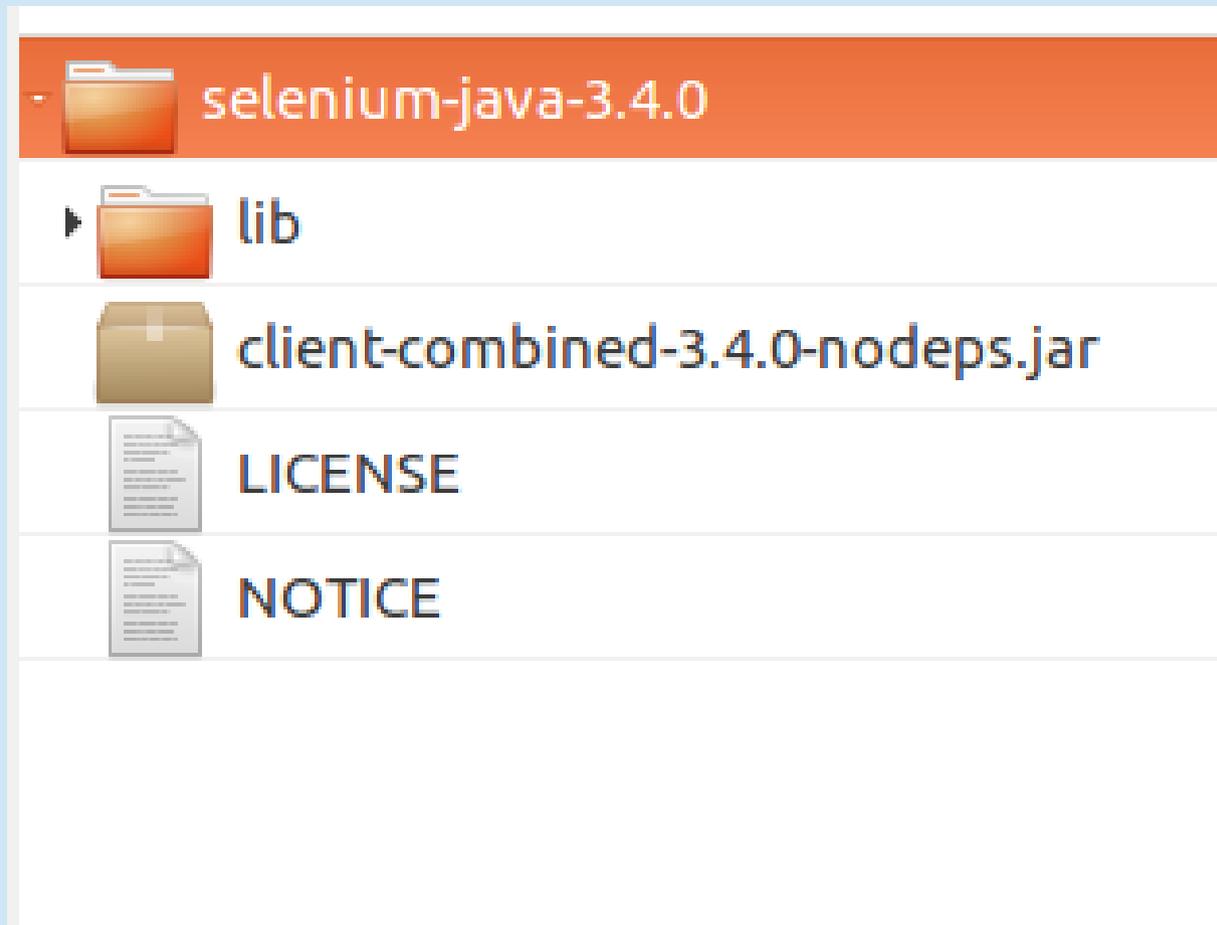
## Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

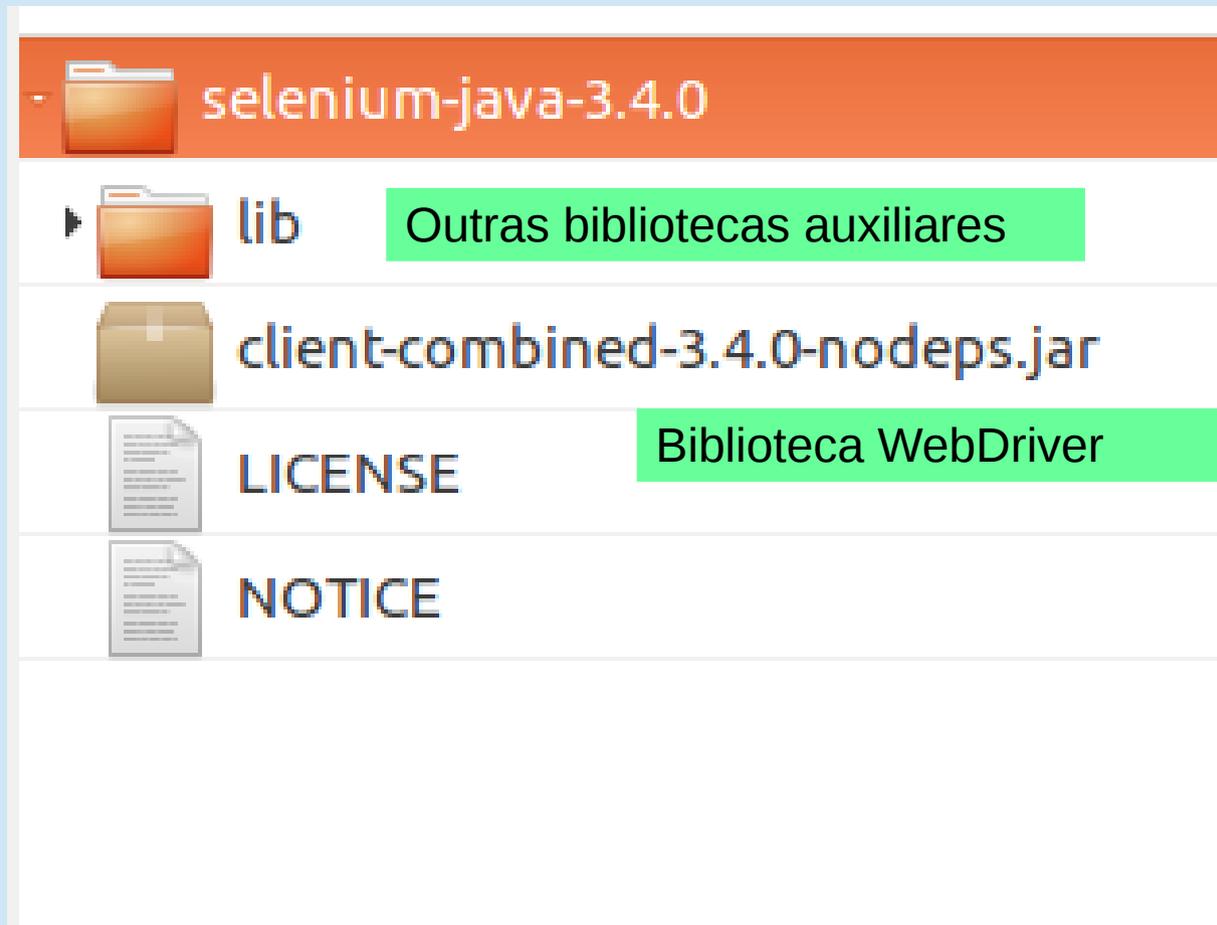
While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on google code.

Language	Client Version	Release Date	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">Javadoc</a>
Java	3.4.0	2017-04-21	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">Javadoc</a>
C#	3.4.0	2017-04-21	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Ruby	3.4.0	2017-04-21	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Python	3.4.1	2017-04-27	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Javascript (Node)	3.4.0	2017-04-21	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>

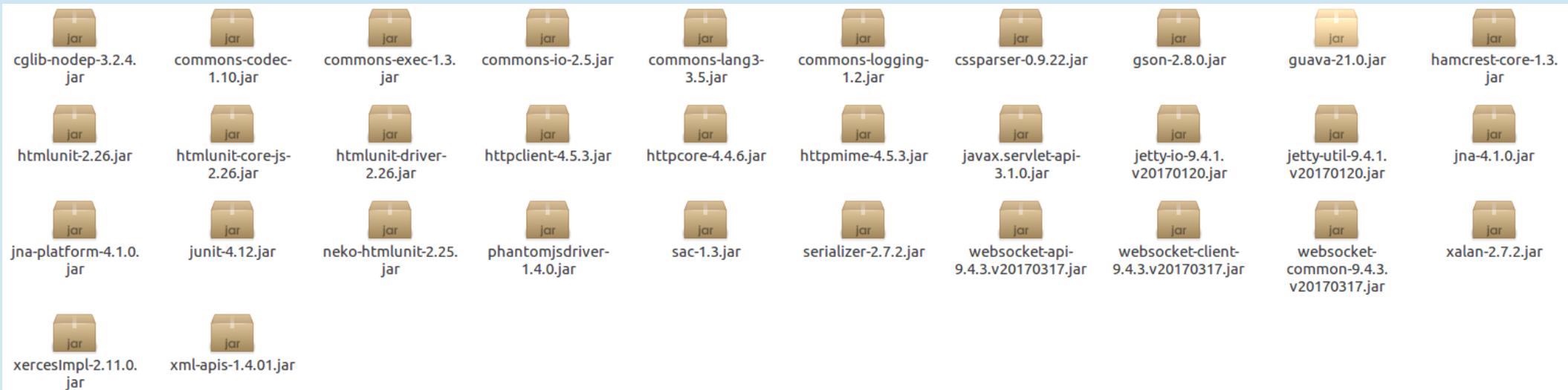
# WebDriver – conteúdo



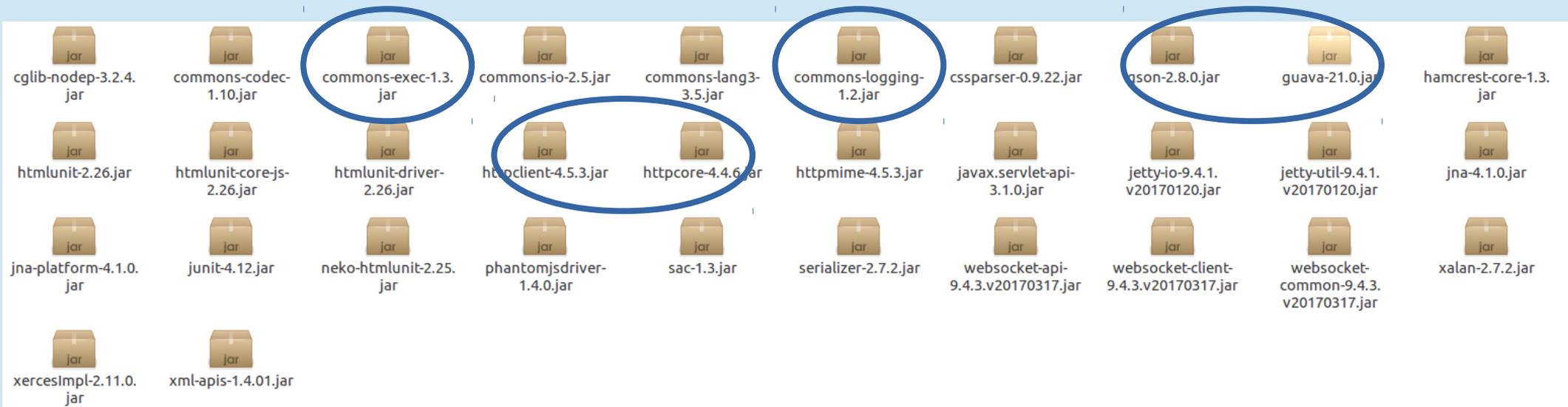
# WebDriver – conteúdo



# WebDriver – conteúdo



# WebDriver – conteúdo



# Mais algumas coisinhas

- Usar Java8
- Fazer download geckodriver e descompactar
  - <https://github.com/mozilla/geckodriver/releases>
- Registrar o driver no código
- Corrigir o código
- Executar como JUnit

# Registrar o driver

@Before

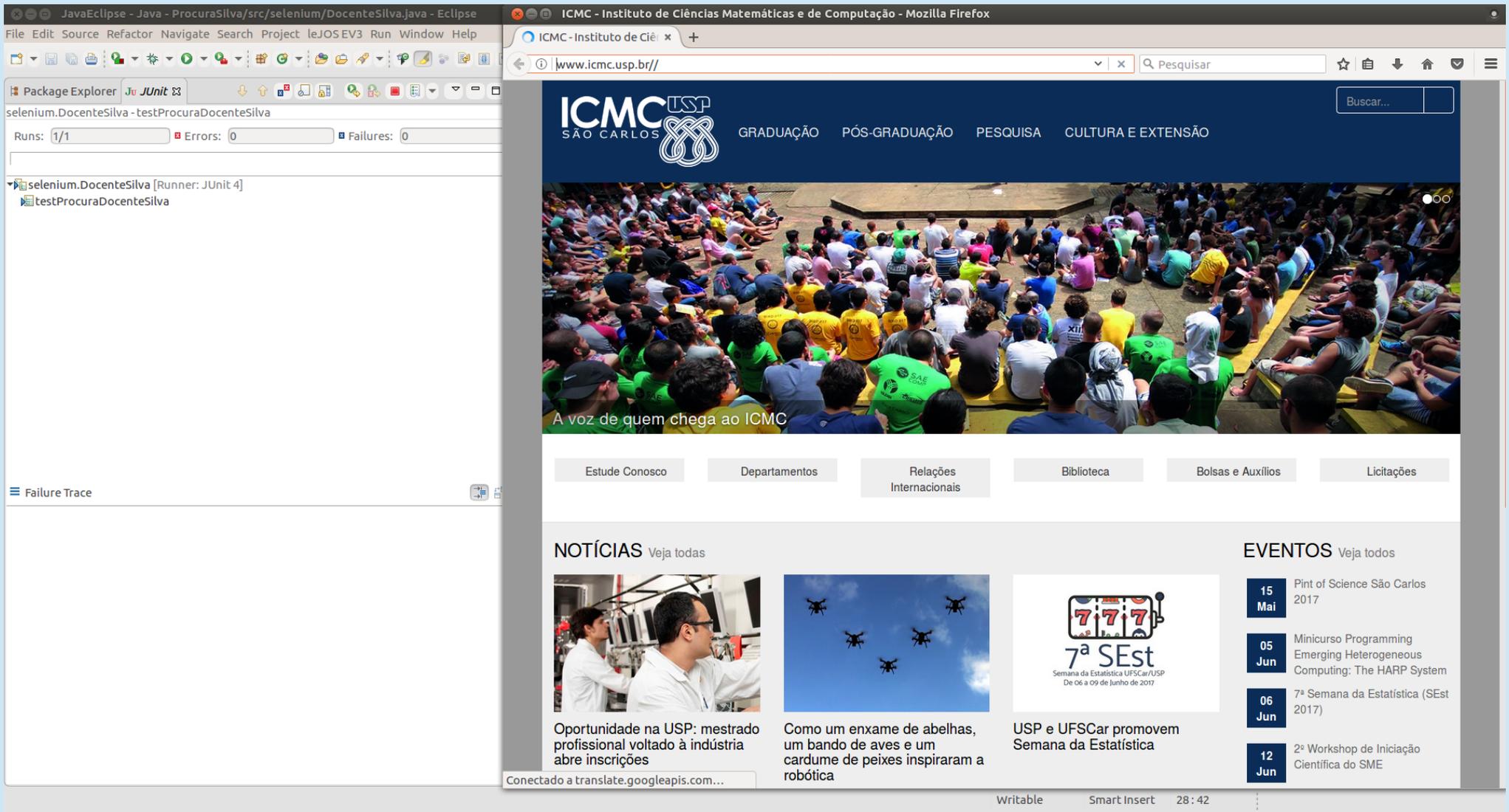
```
public void setUp() throws Exception {  
  
    System.setProperty("webdriver.firefox.marionette", <gecko>);  
  
    driver = new FirefoxDriver();  
  
    baseUrl = "http://www.icmc.usp.br/";  
  
    driver.manage().timeouts().implicitlyWait(30,  
                                                TimeUnit.SECONDS);  
  
}
```

# Corrigir o código

@Test

```
public void testProcuraDocenteSilva() throws Exception {  
    driver.get(baseUrl + "/");  
    driver.findElement(By.linkText("Pessoas")).click();  
    driver.findElement(By.linkText("Docentes")).click();  
    Thread.sleep(10000);  
}
```

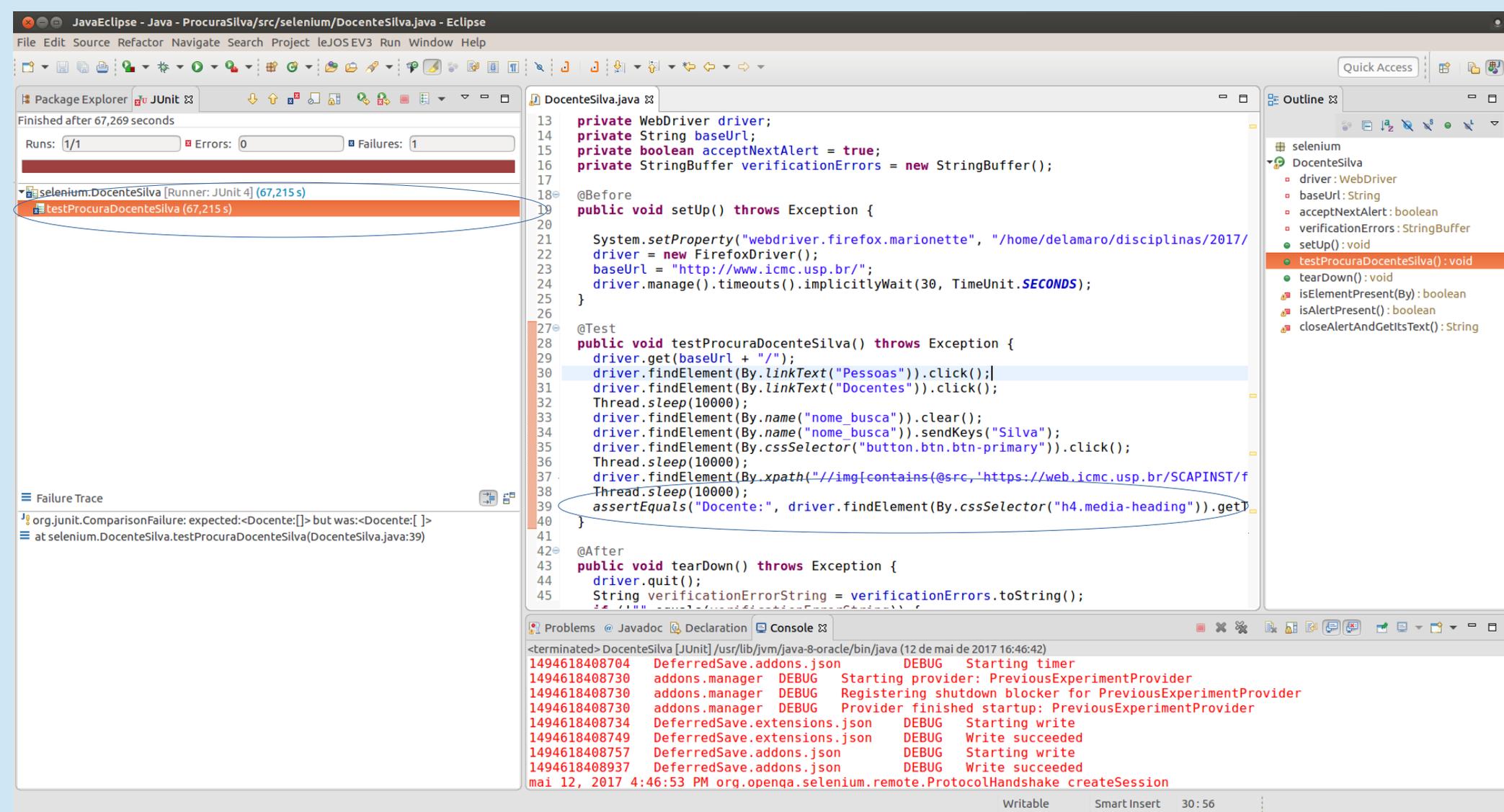
# Executar



The image shows a dual-screen environment. On the left, the Eclipse IDE is running a Selenium test. The Package Explorer shows 'JUnit 3' and 'JUnit 4' tabs. The console displays 'selenium.DocenteSilva - testProcuraDocenteSilva' with 'Runs: 1/1', 'Errors: 0', and 'Failures: 0'. The test runner is 'JUnit 4' and the test class is 'testProcuraDocenteSilva'. A 'Failure Trace' section is visible at the bottom left.

On the right, the Mozilla Firefox browser displays the ICMC website. The address bar shows 'www.icmc.usp/'. The website header includes the ICMC USP logo and navigation links: 'GRADUAÇÃO', 'PÓS-GRADUAÇÃO', 'PESQUISA', and 'CULTURA E EXTENSÃO'. A search bar is located in the top right corner. The main content area features a large image of a crowd of people sitting on the ground, with the text 'A voz de quem chega ao ICMC' overlaid. Below the image are several menu items: 'Estude Conosco', 'Departamentos', 'Relações Internacionais', 'Biblioteca', 'Bolsas e Auxílios', and 'Licitações'. The 'NOTÍCIAS' section includes three news items: 'Oportunidade na USP: mestrado profissional voltado à indústria abre inscrições', 'Como um enxame de abelhas, um bando de aves e um cardume de peixes inspiraram a robótica', and 'USP e UFSCar promovem Semana da Estatística'. The 'EVENTOS' section lists three events: '15 Mai Pint of Science São Carlos 2017', '05 Jun Minicurso Programming Emerging Heterogeneous Computing: The HARP System', and '06 Jun 7ª Semana da Estatística (SEst 2017)'. The date '12 Jun' is also visible.

# Executar



The screenshot shows the Eclipse IDE with the Selenium test runner. The Package Explorer on the left shows the test suite 'selenium.DocentesSilva' and the test 'testProcuraDocenteSilva' which has failed. The main editor displays the source code for 'DocentesSilva.java' with the following content:

```

13 private WebDriver driver;
14 private String baseUrl;
15 private boolean acceptNextAlert = true;
16 private StringBuffer verificationErrors = new StringBuffer();
17
18 @Before
19 public void setUp() throws Exception {
20
21     System.setProperty("webdriver.firefox.marionette", "/home/delamaro/disciplinas/2017/
22     driver = new FirefoxDriver();
23     baseUrl = "http://www.icmc.usp.br/";
24     driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
25 }
26
27 @Test
28 public void testProcuraDocenteSilva() throws Exception {
29     driver.get(baseUrl + "/");
30     driver.findElement(By.linkText("Pessoas")).click();
31     driver.findElement(By.linkText("Docentes")).click();
32     Thread.sleep(10000);
33     driver.findElement(By.name("nome_busca")).clear();
34     driver.findElement(By.name("nome_busca")).sendKeys("Silva");
35     driver.findElement(By.cssSelector("button.btn.btn-primary")).click();
36     Thread.sleep(10000);
37     driver.findElement(By.xpath("//img[contains(@src,'https://web.icmc.usp.br/SCAPINST/f
38     Thread.sleep(10000);
39     assertEquals("Docente:", driver.findElement(By.cssSelector("h4.media-heading")).getT
40 }
41
42 @After
43 public void tearDown() throws Exception {
44     driver.quit();
45     String verificationErrorString = verificationErrors.toString();

```

The Failure Trace at the bottom left shows the error: `org.junit.ComparisonFailure: expected:<Docente:[]> but was:<Docente:[ ]> at selenium.DocentesSilva.testProcuraDocenteSilva(DocentesSilva.java:39)`. The Console at the bottom right shows the Selenium WebDriver initialization logs.