

# Testes de Software

1

## AULA 04 – FERRAMENTAS DE AUTOMAÇÃO DE TESTES FUNCIONAIS

**Anne Caroline O. Rocha – Tester Certified – BSTQB – NTI|UFPB**

# Conteúdo Programático – Aula 04

2

- Automação de testes funcionais
- Ferramentas para testes funcionais em Aplicações WEB
- Tutorial para a ferramenta *Selenium*
  - Visão geral
  - Selenium IDE
  - Selenium Remote Control
  - Selenium Grid
- Tutorial para a ferramenta *Badboy*
- Referências

# Automação de Testes Funcionais

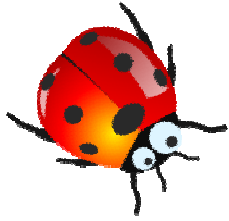
3

É uma maneira de realizar testes automáticos sobre a interface dos sistemas, onde os testes tem foco nas funcionalidades do sistema.

# Ferramentas para Testes Funcionais em Aplicações WEB

4

- São ferramentas que verificam as funcionalidades de uma aplicação web através da interface gráfica.
- Ferramentas:
  - Selenium - <http://seleniumhq.org>
  - BadBoy - <http://www.badboy.com.au>
  - Watir - <http://wtr.rubyforge.org>
  - actiWATE - <http://www.actiwate.com>
  - Canoo WEBTest - <http://WEBtest.canoo.com>
  - Apodora - <http://www.apodora.org>



# Testes de Software

5

## AULA04 – TUTORIAL PARA FERRAMENTA SELENIUM

# Selenium – Visão Geral

6

- Desenvolvido pela ThoughtWorks
  - Distribuída sob a Apache License 2.0
- Simula a interação (digitação, cliques) de usuário e faz assertivas sobre valores das páginas WEB
  - Testes são executados como **Javascript** no browser
- Tem script próprio, mas também fornece API para outras linguagens de programação
  - Não orientado a objetos

# Selenium – Visão Geral

7

- Alguns conhecimentos importantes:
  - XPath, HTML, Javascript, Expressões Regulares, Testes...
- Existem várias formas de execução
  - Selenium IDE
  - Selenium RC (Remote Control)
  - Selenium Grid

# Selenium – Selenium Core

8

- Alguns conhecimentos importantes:
  - XPath, HTML, Javascript, Testes... Selenium deve ser instalado no mesmo domínio
  - Utiliza-se o TestRunner.html para rodar os testes
- Compatível com todos os browsers que suportam Javascript
- Testes devem estar hospedados no servidor
  - Escritos em simples HTML
  - Podem ser agrupados em Test Suites
- Desvantagens:
  - Necessária instalação remota no servidor
  - Pode ter comportamento irregular com testes de páginas AJAX
    - ✦ Mas é possível adicionar comandos para realizar estes testes



# Selenium – Selenium IDE

9

- Extensão do Firefox
  - Comandos podem ser gravados e executados automaticamente
  - Ambiente para edição e debug dos testes
- Exporta scripts para outras linguagens de programação
- Vantagens
  - Fácil de usar / instalar
  - Dispensa programação para vários comandos
- Desvantagens
  - Só há plugin para o Firefox

# Selenium – Selenium IDE

10

- Tabela HTML com 3 colunas:
  1. Nome do comando
  2. Alvo (target)
    - Id, name ou XPath de elemento HTML
    - Ou parâmetro da função
  3. Valor (value)
    - Segundo parâmetro da função

Test		
type	q	dsc.ufcg
clickAndWait	btnG	
verifyTextPresent	www.dsc.ufcg.edu.br	

# Selenium – Selenium RC (Remote Control)

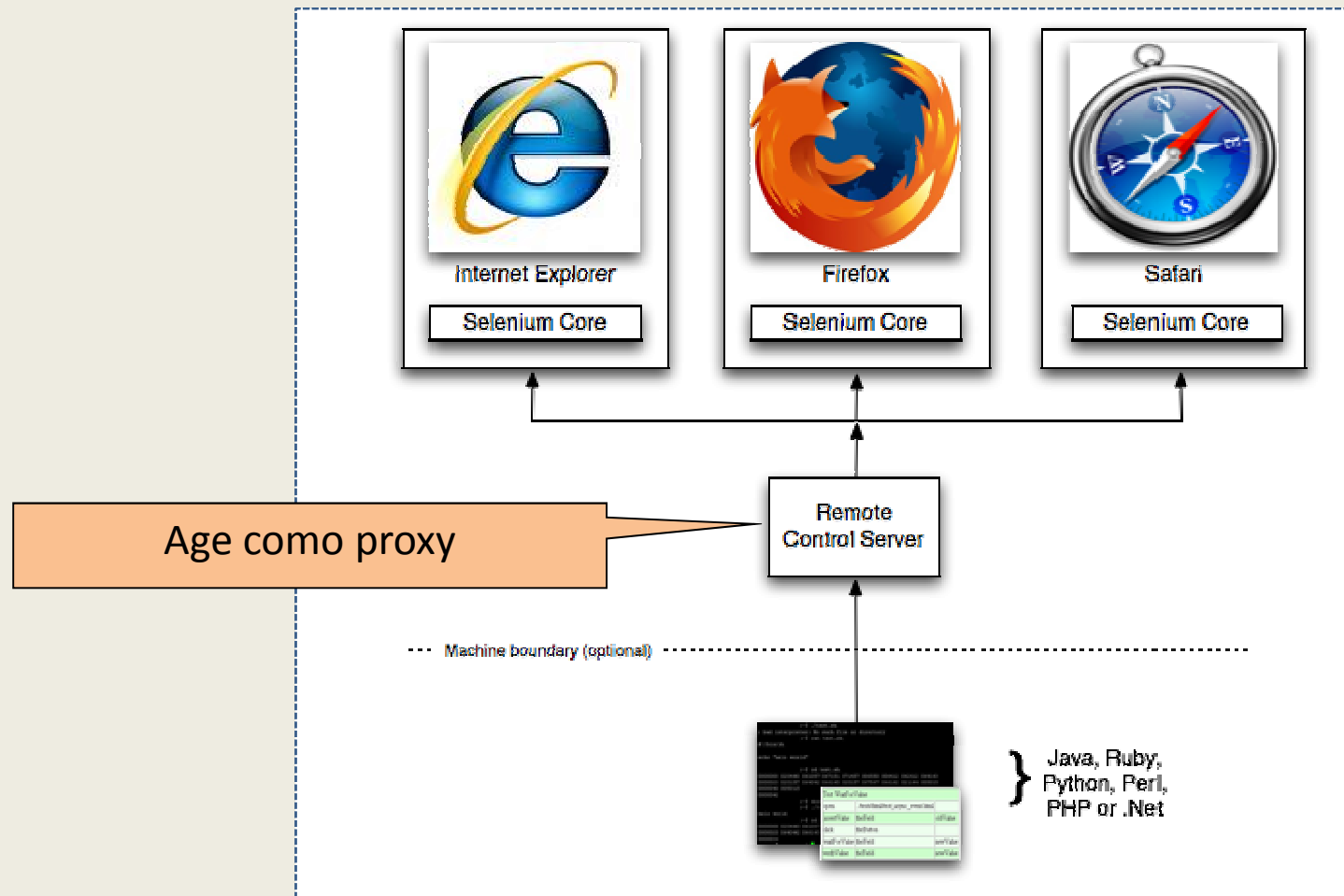
11

- Permite escrever e executar testes usando uma linguagem de programação
  - Java, C#, Python, Ruby, PHP e Perl.
- Contém o **Selenium Server**
  - Executa browsers como um **proxy**
- Ideal para:
  - Ambientes de **integração contínua**
    - ✦ Facilita adição dos testes de interface com os demais
  - Testar interfaces com muito AJAX
  - Integrar-se com linguagens de programação mais **expressivas** (além de **compiláveis**) que HTML

# Selenium – Selenium RC (Remote Control)

12

Windows, Linux, or Mac (as appropriate)...



# Selenium – Selenium RC (Remote Control)

13

```
public class GoogleTest extends TestCase {

    private Selenium browser;

    public void setUp() {
        browser = new DefaultSelenium("localhost",
            4444, "*firefox", "http://www.google.com");
        browser.start();
    }
    public void testGoogle() {
        browser.open("http://www.google.com/webhp?hl=en");
        browser.type("q", "hello world");
        browser.click("btnG");
        browser.waitForPageToLoad("5000");
        assertEquals("hello world - Google Search", browser.getTitle());
    }
    public void tearDown() {
        browser.stop();
    }
}
```

Equivalente ao HTML:

GoogleTest		
open	http://www.google.com/webhp?hl=en	
type	q	hello world
click	btnG	
waitForPageToLoad	5000	
verifyTitle	hello world - Google Search	

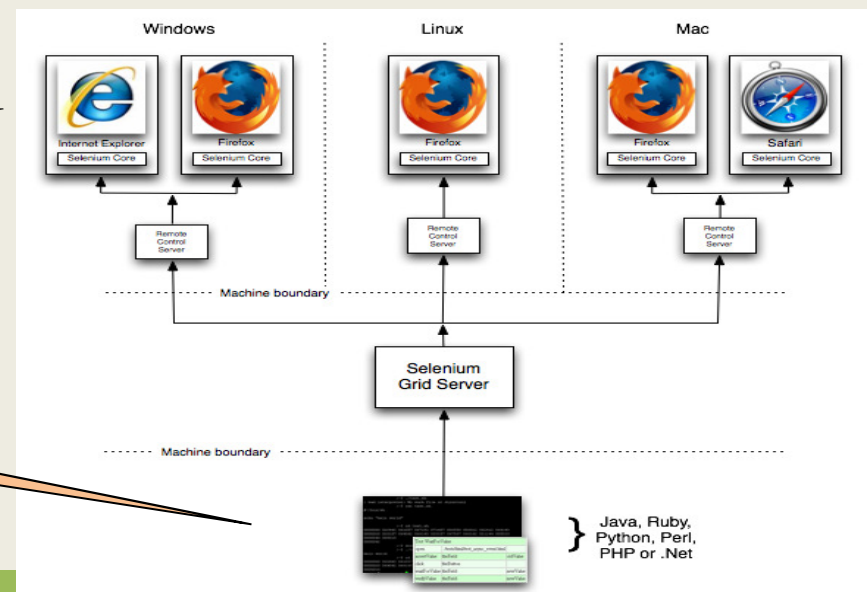
# Selenium – Selenium Grid

14

- Semelhante ao Selenium RC
- Pode executar mais de um browser e executar **vários testes em paralelo**
- Economiza tempo
- Pode testar em mais de um S.O. / browser

Vários  
Browsers / S.O.s

Código dos testes



# Selenium – Comparação

15

	<b>Selenium IDE</b>	<b>Selenium RC</b>
<b>Browsers</b>	Só Firefox	Muitos
<b>Instalação remota?</b>	Não	Não
<b>HTTPS/SSL?</b>	Sim	Sim
<b>Múltiplos domínios</b>	Sim	Sim
<b>Linguagens</b>	HTML/Selenese	Muitas

# Selenium – Comandos

16

## Comandos

## Elementos

## Valores

open	/page	
select	pessoa_datanascimento_3i	label=1
select	pessoa_datanascimento_2i	label=Junho
select	pessoa_datanascimento_1i	label=1998
clickAndWait	pessoa_submit	
clickAndWait	link=Alterar Dados Pessoais	
verifyNotValue	//*[@id="endereco_email"]	regexp:^([a-zA-Z0-9_\\-\\.]+)@((([a-z0-9]+(\\.[a-z0-9]+)*\\.([a-z]{2,3})) (((01)?\\d\\d? 2[0-4]\\d 25[0-5])\\.){3}((01)?\\d\\d? 25[0-5] 2[0-4]\\d))\$
type	endereco_email	email@
clickAndWait	endereco_submit	
verifyTextPresent	Email não é válido.	

Diagram illustrating the use of XPath and Regular Expressions in Selenium WebDriver:

- XPath:** Used to locate elements by their ID or Name. Example: `//*[@id="endereco_email"]`.
- Expressão Regular (Regular Expression):** Used to validate the value of an element. Example: `regexp:^([a-zA-Z0-9_\\-\\.]+)@((([a-z0-9]+(\\.[a-z0-9]+)*\\.([a-z]{2,3}))|(((01)?\\d\\d?|2[0-4]\\d|25[0-5])\\.){3}((01)?\\d\\d?|25[0-5]|2[0-4]\\d))$`.

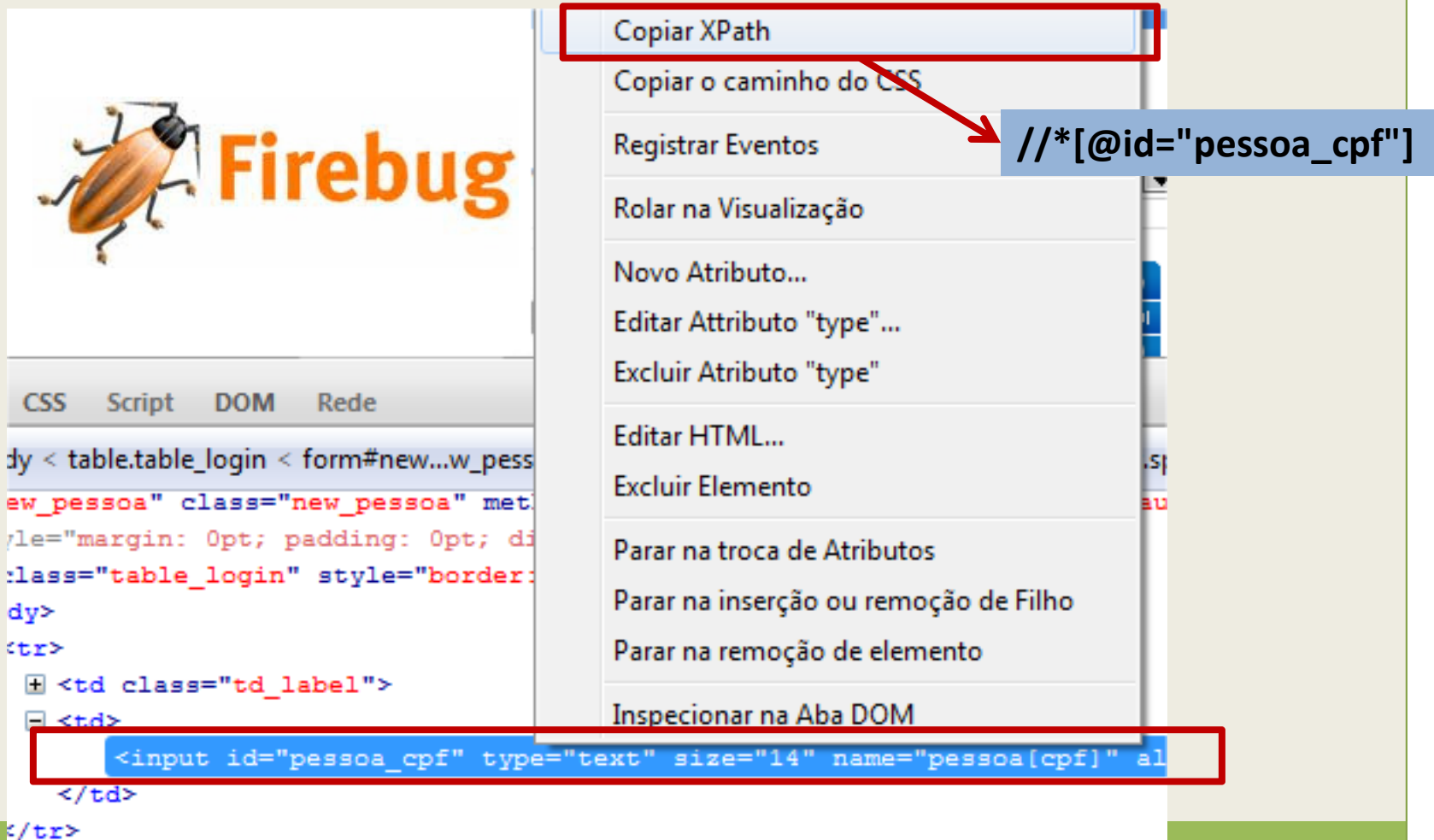
# XPath

## Expressão Regular



# Selenium – XPath

17



The screenshot shows the Firebug interface with the DOM tree selected. A context menu is open over an  element. The menu options are:

- Copiar XPath
- Copiar o caminho do CSS
- Registrar Eventos
- Rolar na Visualização
- Novo Atributo...
- Editar Atributo "type"...
- Excluir Atributo "type"
- Editar HTML...
- Excluir Elemento
- Parar na troca de Atributos
- Parar na inserção ou remoção de Filho
- Parar na remoção de elemento
- Inspecionar na Aba DOM


The XPath expression `//*[@id="pessoa_cpf"]` is shown in a blue box, with a red arrow pointing to it from the 'Copiar XPath' option. The DOM tree shows the following structure:

```
<table.table_login < form#new...w_pess  
ew_pessoa" class="new_pessoa" met  
yle="margin: 0pt; padding: 0pt; di  
:class="table_login" style="border:  
dy>  
<tr>  
+ <td class="td_label">  
= <td>  
  <input id="pessoa_cpf" type="text" size="14" name="pessoa[cpf]" al  
</td>  
</tr>
```

# Selenium – XPath

18

- Exemplos de XPaths utilizados
  - `//div[@id='msg_box']`
  - `//table[@class='grid']//*[contains(text(),'Eric')]`
  - `//a[contains(@href,'remove')]`

Nome	Sobrenome		
Eric	Amundsen		

# Selenium – Expressão Regular

19

- É um método formal para especificar padrão de texto
- Utilizado para validar strings como:
  - Data
  - Horário
  - E-mail
  - URL
  - Telefone
  - RG
  - CPF
  - Cartão de Crédito

# Selenium – Expressão Regular

20

- Caracteres:
  - `^`: simboliza o início de uma linha em um arquivo
  - `$`: simboliza o fim de uma linha em um arquivo
  - `[]`: encontra um ou outro caractere da expressão
    - ✦ Ex.: `[Cc]arlos` -> Procura por **Carlos** ou **carlos**
    - ✦ `^[ab]` -> Procura por palavras iniciando com **a** ou **b** no início do arquivo.
  - `.`: encontra qualquer caractere em uma determinada posição.
    - ✦ Ex.: `.[aeiou]` -> Procura por palavras cuja segunda letra seja uma vogal.
    - ✦ `^.....$` -> Procura por linhas com 5 caracteres.

# Selenium – Expressão Regular

21

- Caracteres:
  - {}: simboliza a quantidade de caracteres repetidos
    - ✦ Ex.: `^{5}$` é equivalente a expressão `^.....$`
    - ✦ `^a{20, 40}$` define um intervalo de repetições mínimas e máximas.
    - ✦ `^[0123456789]{3,}$` define um intervalo aberto de repetições apenas com o *mínimo*.
      - Retorna palavras que possuam números com 3 ou mais dígitos, por exemplo, *Maria tem 150 bolas*.
  - .\*: quando se procura dois trechos específicos em uma mesma linha.
    - ✦ Ex.: `^[aeiou].*vida$` -> a linha começa com uma vogal e possui a palavra vida no final da linha.

# Selenium – Expressão Regular

22

- Caracteres:
  - |: para fazer um OR lógico, onde se procura por uma coisa ou outra.
    - ✦ Ex.: **^(ana|carlos)**: procura por linhas que iniciem com **ana:** ou **carlos:**
  - ?, \*, +: definem quantidade e funcionam como a {}.
    - ✦ ? = {0,1} pode aparecer ou não aparecer.
      - Ex. a? -> o caractere 'a' pode aparecer 0 ou não aparecer.
    - ✦ \* = {0,} pode aparecer em qualquer quantidade.
      - Ex. a\* -> o caractere 'a' pode aparecer 0 ou mais vezes.
    - ✦ + = {1,} deve aparecer no mínimo 1 vez .
      - Ex. a+ -> o caractere 'a' pode aparecer 1 ou mais vezes.

# Selenium – Expressão Regular

23

- Caracteres:

- **[^]**: para não incluir na pesquisa alguns caracteres.

- ✦ Ex.: **^[^aeiou]** -> a linha deve começar com qualquer caractere diferente de vogal.

- **[-]**: para fazer um intervalo em listas.

- ✦ Ex.: **[0-9]{3,}**: equivalente à expressão **[0123456789]{3,}** que procura por números com 3 dígitos ou mais na linha.

- ✦ **[A-Za-z]**: caracteres entre A-Z ou entre a-z.

# Selenium – Expressão Regular

24

- Caracteres:
  - **\d** -> qualquer número decimal
  - **\D** -> qualquer caractere que não seja um número decimal
  - **\s** -> qualquer caractere de espaço em branco
  - **\S** -> qualquer caractere que não seja de espaço em branco
  - **\w** -> qualquer caractere de letra, número ou underline \_
  - **\W** -> qualquer caractere que não seja letra, número ou underline \_



# Selenium – Expressão Regular

25

- Aplicando as expressões regulares:
  - Máscara de Telefone
    - ✦ `^\(\d{2}\) \d{4}-\d{4}$`
      - Ex.: (83) 3234-5678
  - Máscara de Data:
    - ✦ `^\d{1,2}(\-|\V)\d{1,2}(\-|\V)\d{4}$`
      - Ex.: 1/2/2011 ou 20/10/2011 ou 5-12-2009

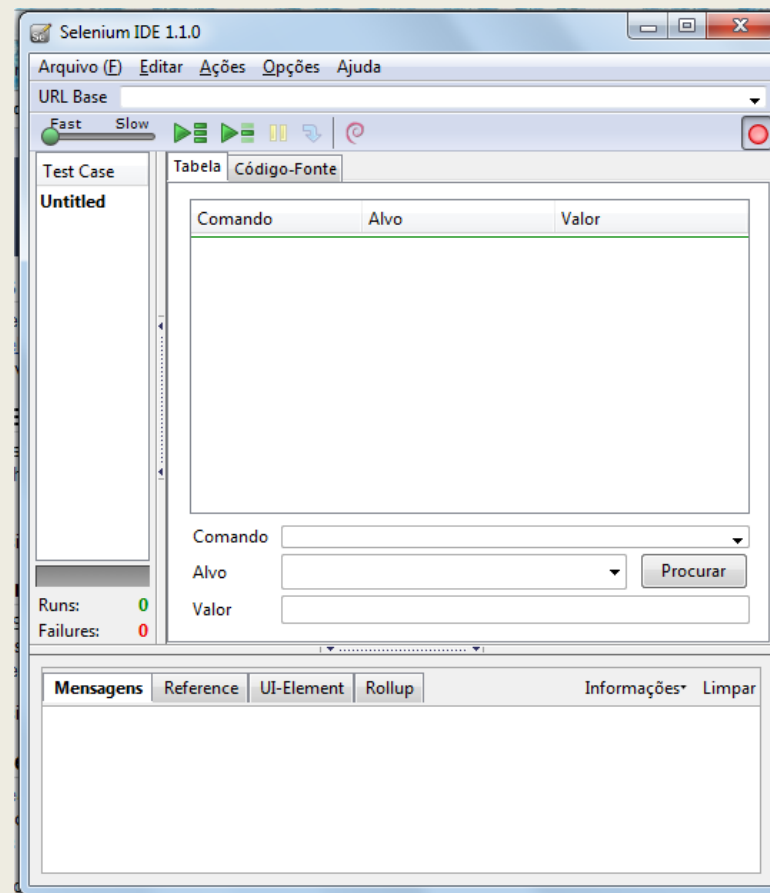
# Selenium IDE

26

- Plugin para Firefox



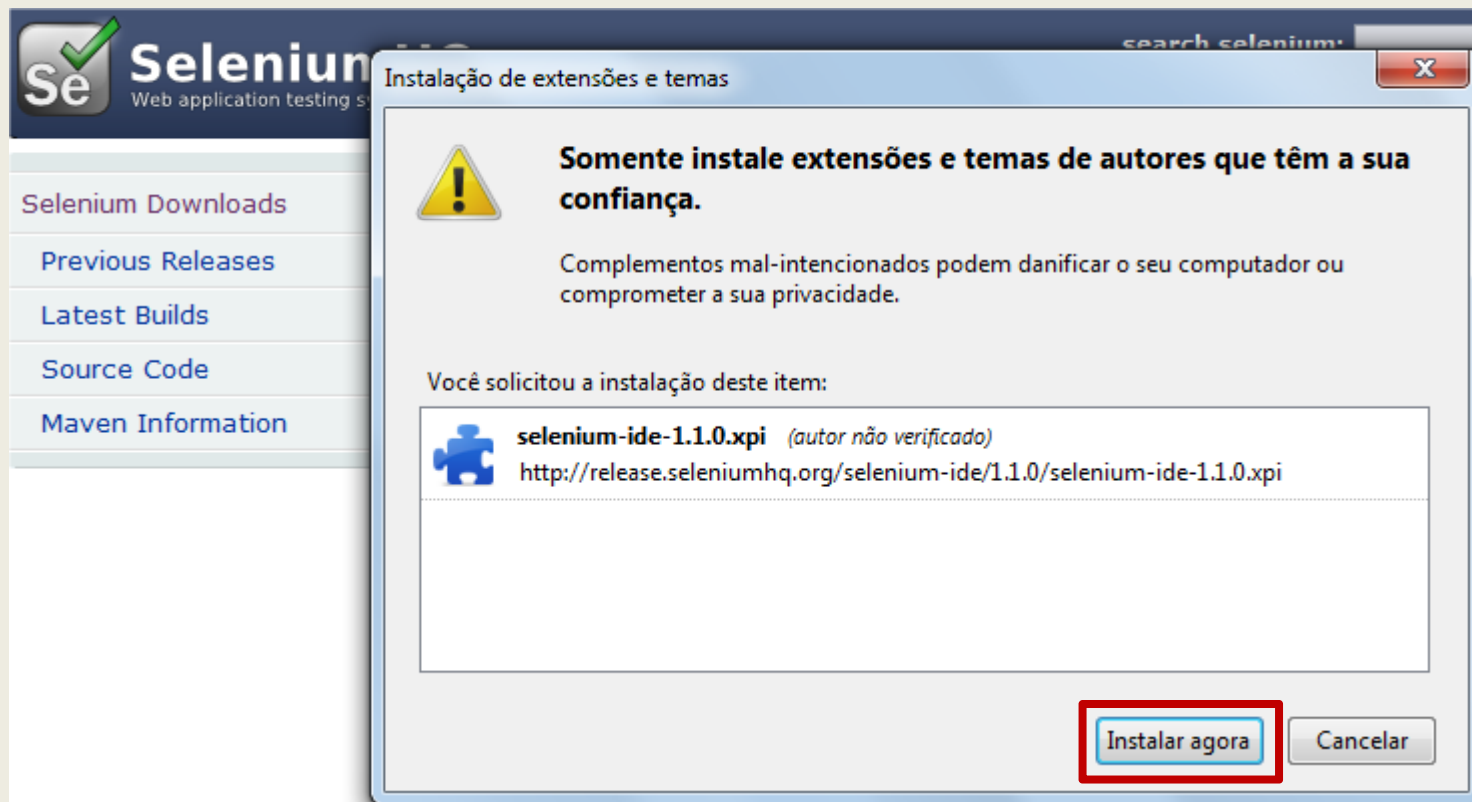
## Selenium IDE



# Selenium IDE – Instalando

27

- Selenium IDE – Plugin para Firefox
  - Acessar: <http://seleniumhq.org/> > Download

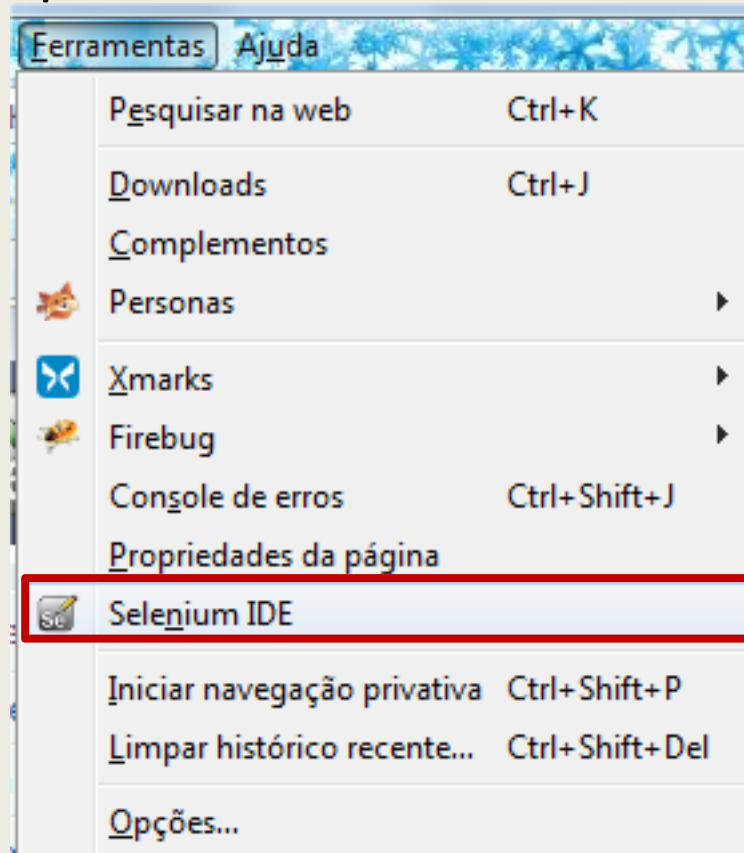


# Selenium IDE – Instalando

28

- Selenium IDE – Plugin para Firefox

- Abrir a ferramenta:



# Selenium IDE

29

Execução de testes

URL Base

Código-fonte

Botão “gravar”

Comandos e parâmetros

Documentação dos comandos

Highlight alvo (se ele for XPath)

The screenshot shows the Selenium IDE application window. At the top, there's a menu bar with 'Arquivo', 'Editar', 'Opções', and 'Ajuda'. Below it is the 'URL Base' field containing 'http://www.google.com.br/'. A toolbar with various icons is next to it. The main area is divided into two tabs: 'Tabela' and 'Código-Fonte'. The 'Código-Fonte' tab is active, showing a table of commands. The table has three columns: 'Comando', 'Alvo', and 'Valor'. The first row is 'type' with target 'q' and value 'dsc ufcg'. The second row is 'clickAndWait' with target 'btnG'. The third row is 'verifyTextPresent' with target 'www.dsc.ufcg.edu.br'. Below the table, there's a section for editing a command. The 'Comando' dropdown is set to 'verifyTextPresent'. The 'Alvo' field contains 'www.dsc.ufcg.edu.br' and the 'Valor' field is empty. A 'Procurar' button is next to the 'Alvo' field. At the bottom, there's a 'Mensagens' tab showing the documentation for 'verifyTextPresent(pattern)'. The documentation states it is generated from 'isTextPresent(pattern)', takes a 'pattern' argument to match the page text, and returns 'true' if it matches. It also includes a description: 'Verifies that the specified text pattern appears somewhere on the rendered page shown to the user.'

Comando	Alvo	Valor
type	q	dsc ufcg
clickAndWait	btnG	
verifyTextPresent	www.dsc.ufcg.edu.br	

Comando: verifyTextPresent  
Alvo: www.dsc.ufcg.edu.br  
Valor:

Procurar

**verifyTextPresent(pattern)**  
Generated from isTextPresent(pattern)  
Arguments:  
    ♦ pattern - a pattern to match with the text of the page  
Returns:  
    true if the pattern matches the text, false otherwise  
Verifies that the specified text pattern appears somewhere on the rendered page shown to the user.

# Selenium IDE

30

Gravar

The screenshot displays the Selenium IDE 1.1.0 interface. On the left, a Mozilla Firefox browser window shows the 'Sipervenet' login page. The page has a red banner indicating '3 erros' (3 errors) and lists validation errors for the CPF field. A context menu is open over the CPF field, with 'verifyTextPresent CPF não pode ficar em branco' highlighted. On the right, the Selenium IDE window shows the 'Código-Fonte' (Source Code) tab. A table lists the test script commands:

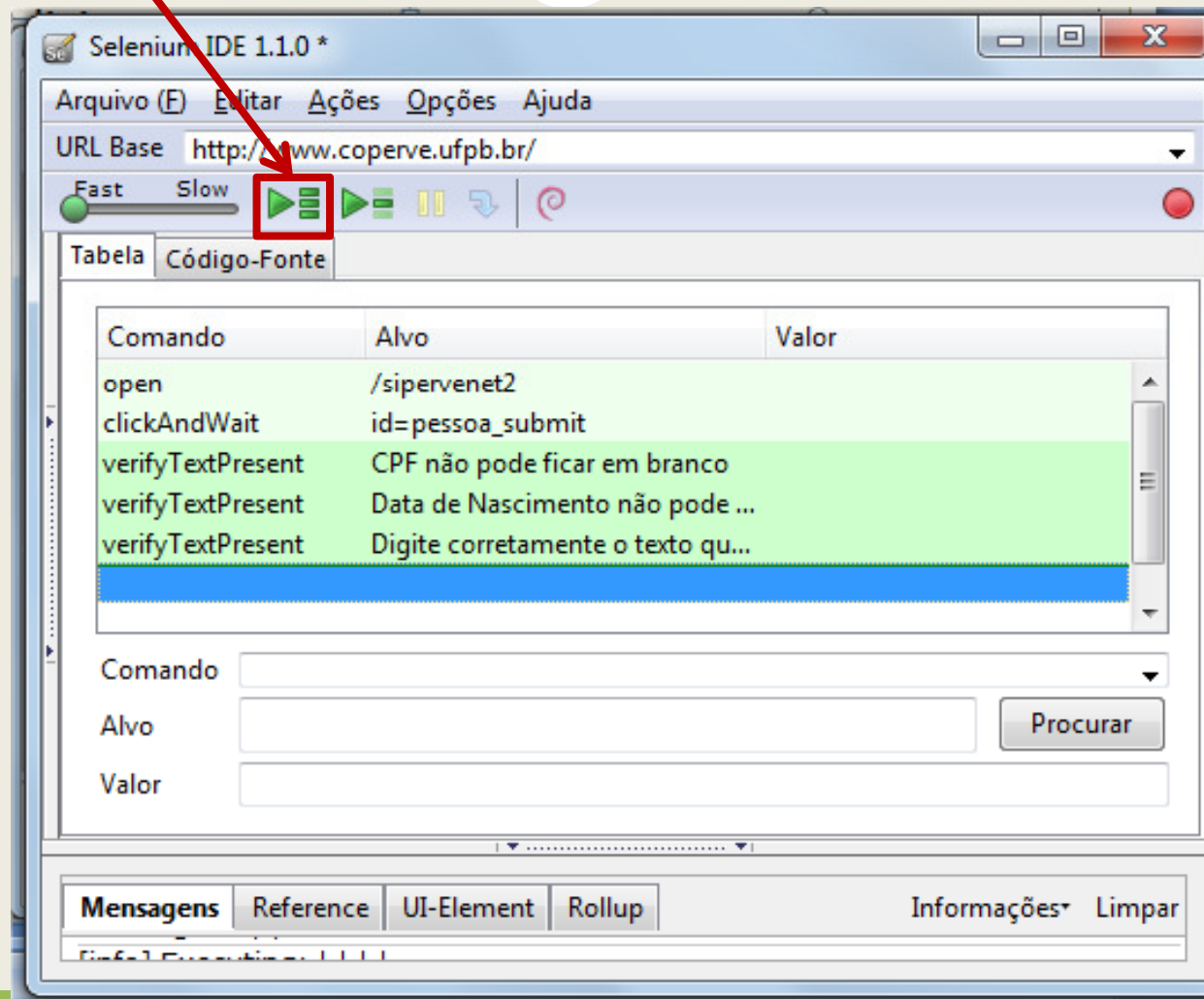
Comando	Alvo	Valor
open	/sipervenet2	
clickAndWait	id=pessoa_submit	
verifyTextPresent		CPF não pode ficar em branco

Below the table, there are input fields for 'Comando', 'Alvo', and 'Valor', along with a 'Procurar' (Search) button. At the bottom of the Selenium IDE window, there are tabs for 'Mensagens', 'Reference', 'UI-Element', and 'Rollup', and buttons for 'Informações' and 'Limpar'.

# Selenium IDE

Executar

31



# Selenium IDE

32

- Comandos básicos do Selenium:

- open
- select
- type
- click
- verifyTextPresent
- verifyTextNotPresent
- verifyValue
- verifyNotValue
- verifyElementPresent
- verifySelectedLabel



# Selenium IDE

33

- Comandos avançados do Selenium:

- Store

- ✦ O comando 'store' permite utilizar funções JavaScript para guardar valores do sistema em variáveis e tornar os testes mais dinâmicos.
    - ✦ Ex.: store | javascript{var d=new Date(); d.toString()} | dataHoje

- storeValue

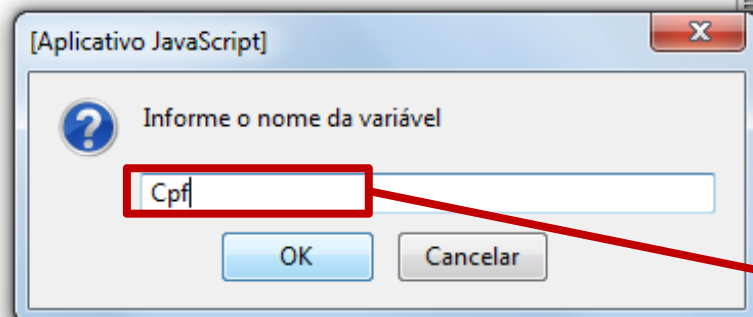
- ✦ O comando 'store' permite guardar valores de campos do sistema em variáveis.
    - ✦ Ex.: storeValue | id=pessoa\_cpf | Cpf

- echo

- ✦ O comando 'echo' exibe para o testador o valor de uma variável.
    - ✦ Ex.: echo | \${dataHoje}

# Selenium IDE

34



CPF: 123.456.789-11

Data Nascimento: dia mês ano

Selenium IDE 1.1.0 \*

Arquivo (F) Editar Ações Opções Ajuda

URL Base <http://www.coperve.ufpb.br/>

Fast Slow

Tabela Código-Fonte

Comando	Alvo	Valor
open	/sipervenet2	
clickAndWait	id=peessoa_submit	
verifyTextPresent	CPF não pode ficar em branco	
verifyTextPresent	Data de Nascimento não pode ...	
verifyTextPresent	Digite corretamente o texto qu...	
type	id=peessoa_cpf	123.456.789-11
storeValue	id=peessoa_cpf	Cpf
echo	\${Cpf}	

Comando echo

Alvo \${Cpf}

Valor

Procurar

Mensagens Reference UI-Element Rollup Informações Limpar

[info] Executing: |verifyTextPresent | Digite corretamente o texto que aparece na imagem. | |

[info] Executing: |type | id=peessoa\_cpf | 123.456.789-11 |

[info] Executing: |storeValue | id=peessoa\_cpf | Cpf |

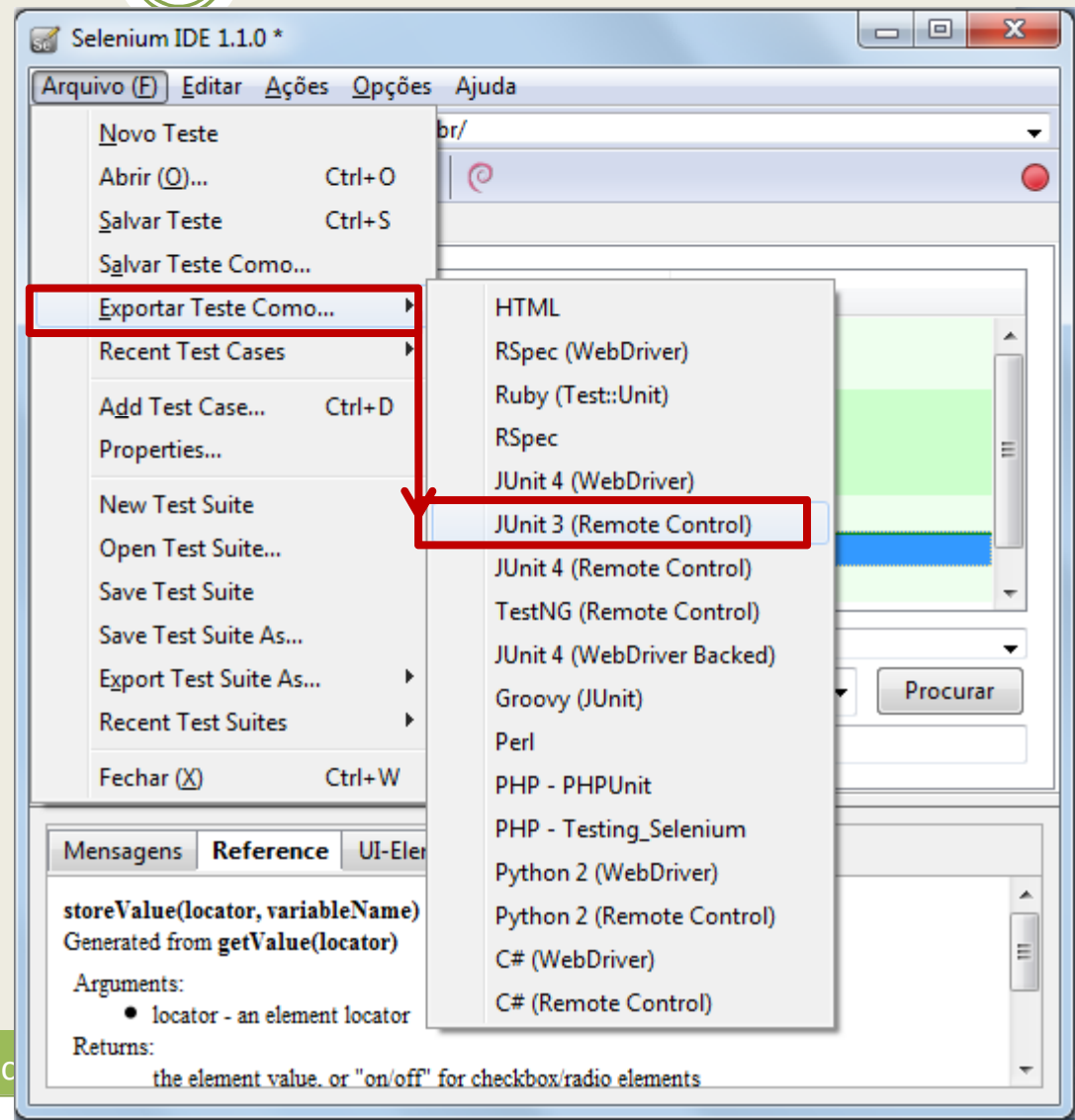
[info] Executing: |echo | \${Cpf} | |

[info] echo: 123.456.789-11

# Selenium IDE para Selenium RC

35

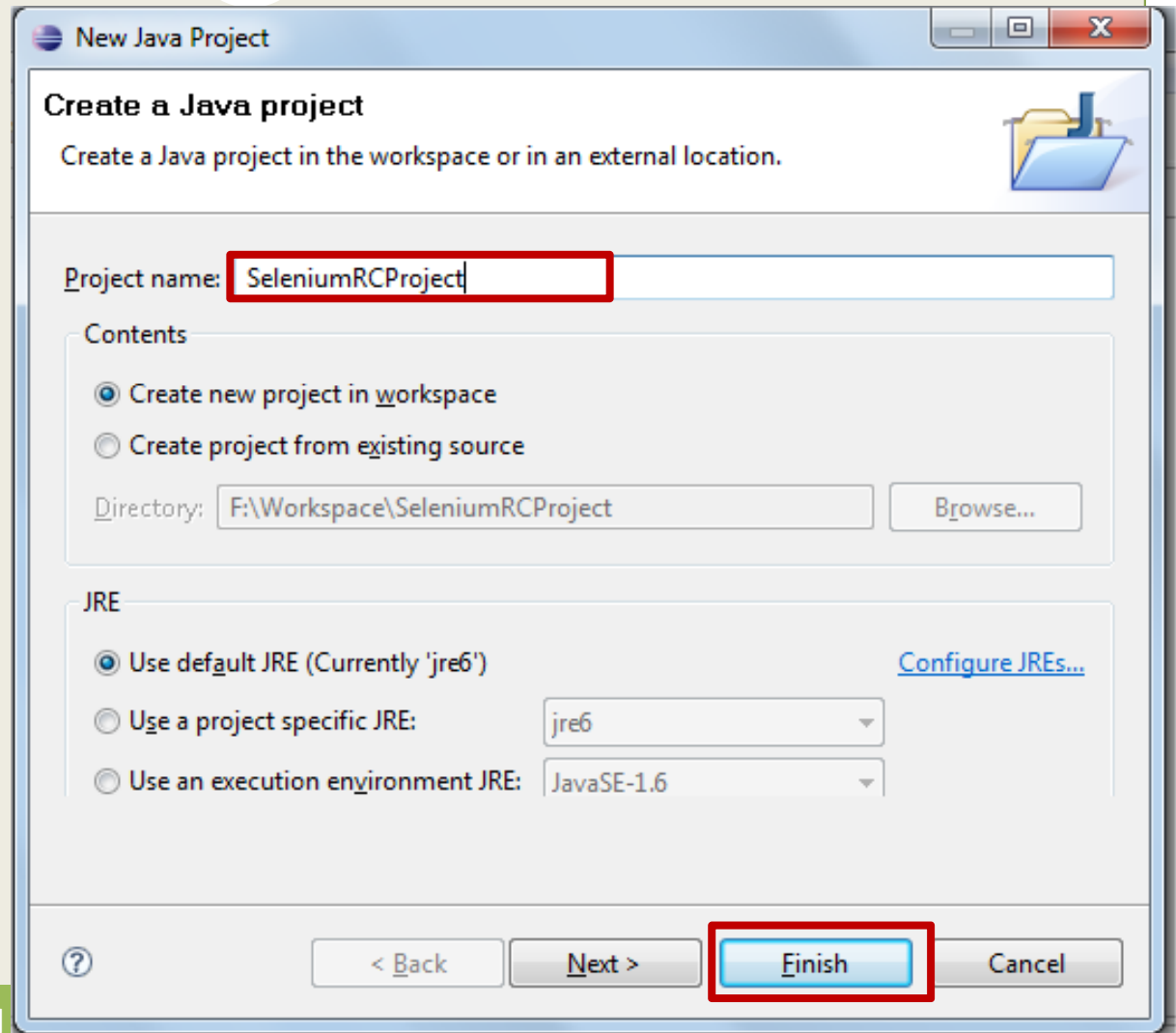
- Exportar o script HTML para:
  - JUnit3 (Remote Control)



# Selenium IDE para Selenium RC

36

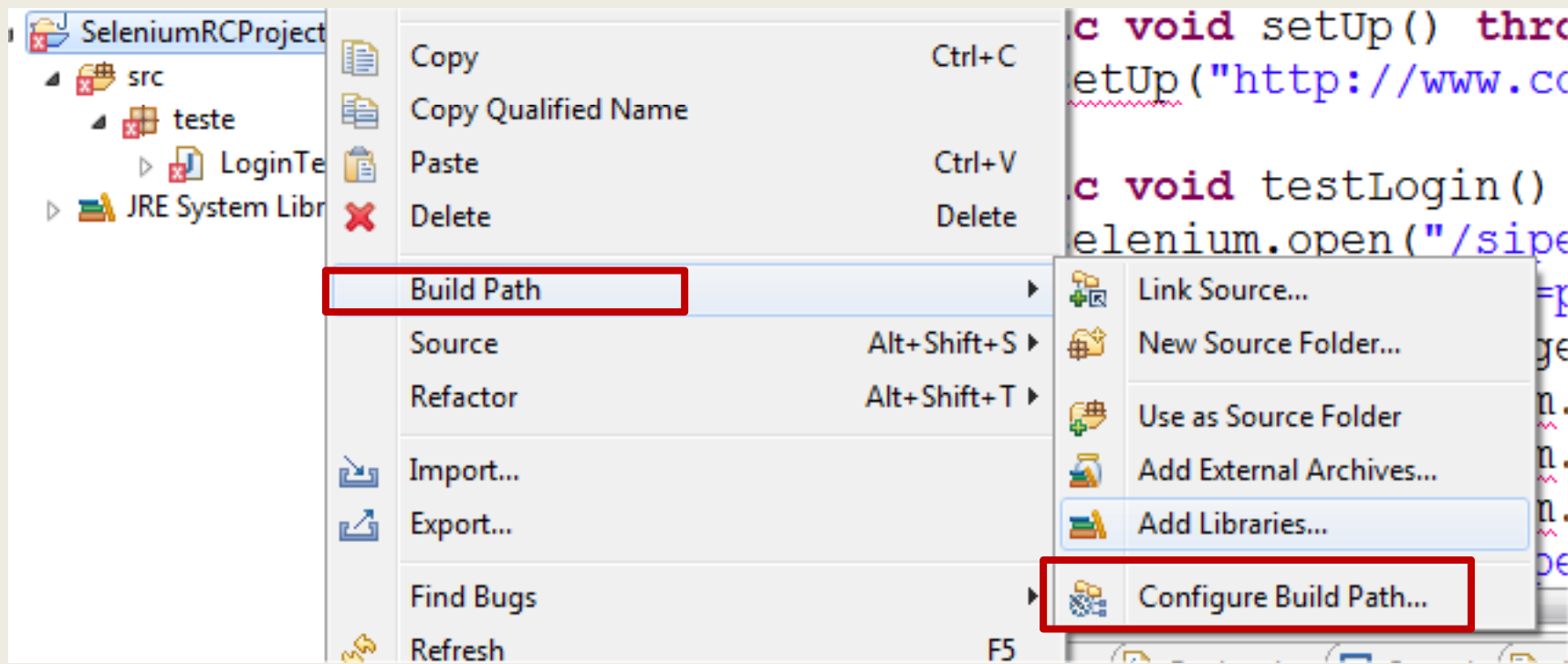
- Criar projeto Java no Eclipse com nome:
  - 'SeleniumRCProject'
- Criar o package:
  - 'teste'



# Selenium IDE para Selenium RC

37

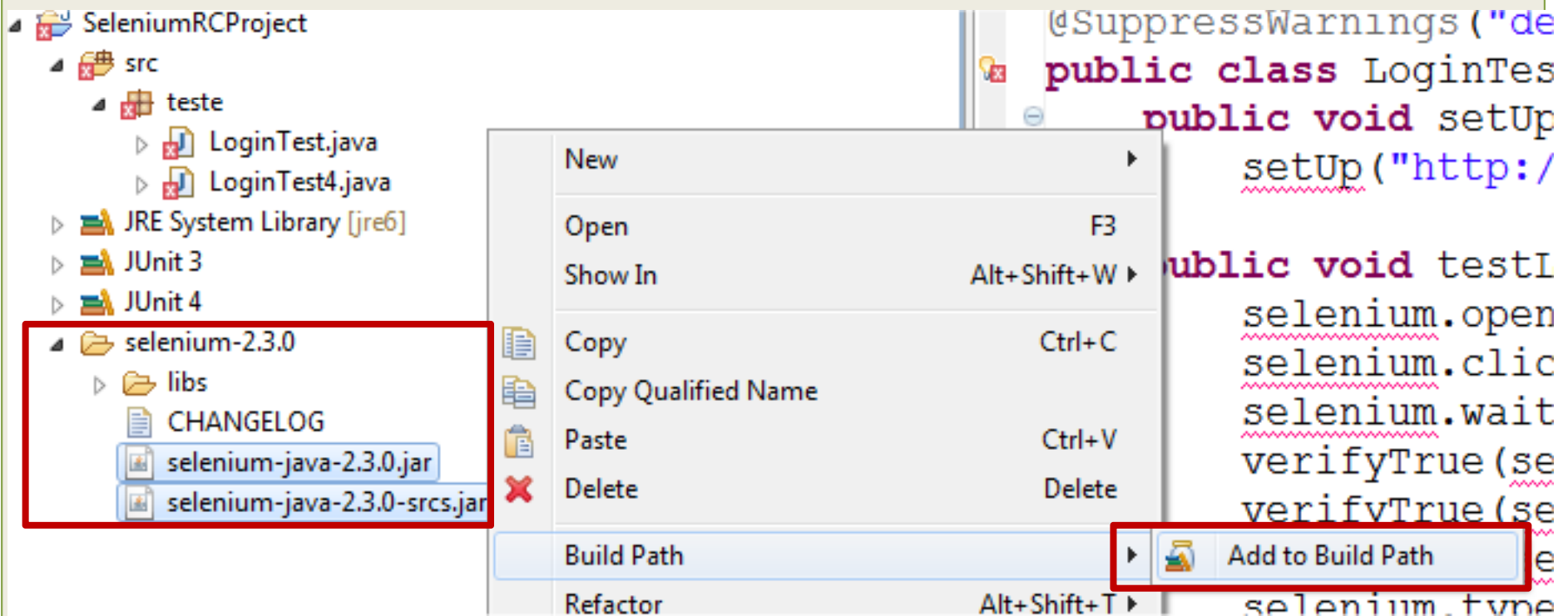
- Colar a classe de teste LoginTest.java do Selenium no pacote 'teste'
- Configurar o Build Path para o JUnit3.



# Selenium IDE para Selenium RC

38

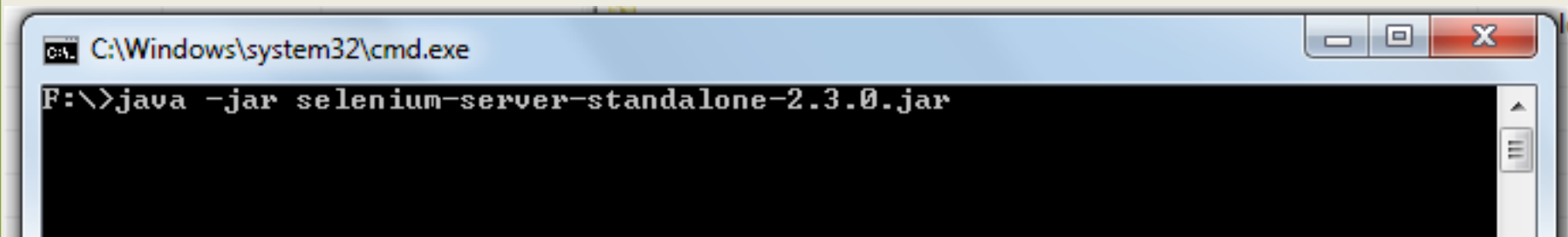
- Fazer download do [selenium-2.3.0.zip](#) - Cliente Java
- Descompactar e colar a pasta no projeto SeleniumRCProject
- Configurar o BuildPath para todos os .jar desta pasta.



# Selenium IDE para Selenium RC

39

- Iniciar o servidor para o Selenium RC
  - Fazer download de 'selenium-server-standalone-2.3.0.jar'
  - Abrir um console onde está localizado o arquivo.
  - Digitar no console:
    - ✦ `java -jar selenium-server-standalone-2.3.0.jar`

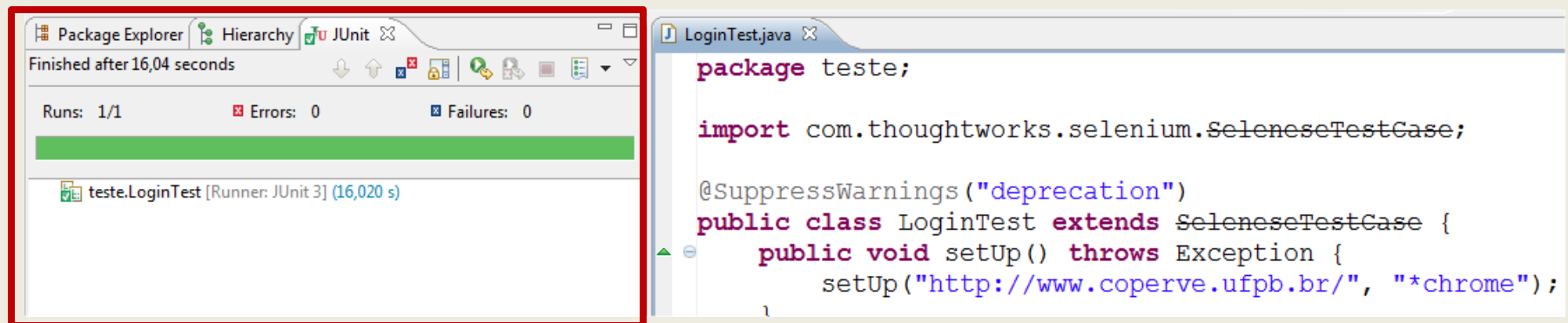
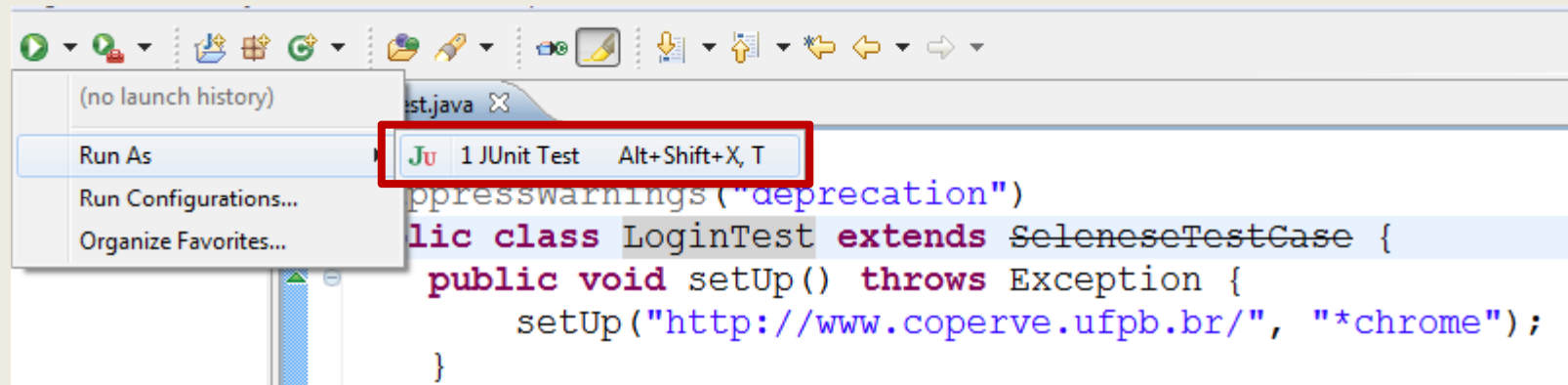


A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt shows the command 'F:\>java -jar selenium-server-standalone-2.3.0.jar' entered at the prompt.

# Selenium IDE para Selenium RC

40

- Executar o Teste > LoginTest.java no Eclipse
- O Firefox é aberto, os testes executados e a barra fica verde.

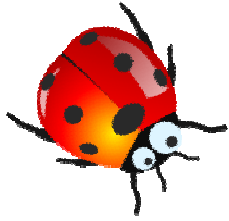




## Selenium IDE - Exercício

41

1. Abrir o roteiro de teste: [RT0005]\_Cadastro\_de\_Responsaveis.doc e executar manualmente o roteiro sobre o sistema da Coperve e ao mesmo tempo gravar um script no Selenium IDE.



# Testes de Software

42

**AULA04 – TUTORIAL PARA FERRAMENTA BADBOY**

# Ferramenta Badboy

43

1. BadBoy é uma ferramenta desenvolvida em C++.
  1. Não funciona no linux
2. Grava todas ações que você faz em uma página web (java, php, ruby, etc...)
3. Fazer download da ferramenta em:
  1. <http://www.badboy.com.au/>

# Ferramenta Badboy

44

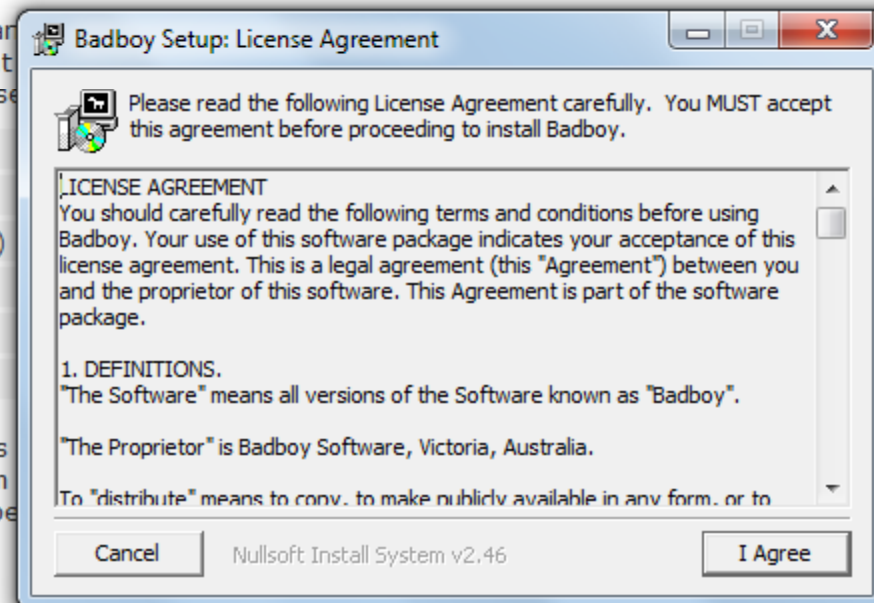
1. Instalar a ferramenta em seu computador (apenas Windows)
2. Abrir a ferramenta Badboy.

## Badboy

Badboy is being continuously developed. You can use the stable version, which may be more stable, or beta versions that are still under development and have not been tested. The following versions are available: false

Version	Description
2.1.2_beta_1	2.1.2 Beta Release
2.1.1	2.1 Update 1 (Recommended)
2.1	2.1
2.0.7	2.0 Update 7
1.6U2	1.6 Update 2

The current recommended version for new users should backup their saved scripts and test them before upgrading. There may be some minor changes in playback behavior.



# Ferramenta Badboy

45

Barra de URL

Botão Recorder

Barra de Ferramentas

Sistema Web a ser testado

Estrutura do Script

Tools



# Ferramenta Badboy

46

- **Como Gravar um Script no BadBoy**

1. Abra o BadBoy. Ele já estará em mode de gravação, que pode ser visualizado pelo botão Record pressionado na barra de ferramentas.
2. Digite a URL na barra de endereço. O BadBoy Automaticamente criará a o *request* com todos os parâmetros da página requisitada.
3. Comece a interagir com a página no painel de visualização da página. Todas as ações na página serão adicionadas na estrutura do script do BadBoy.

# Ferramenta Badboy

47

The screenshot displays the Badboy tool interface, which is used for testing web applications. The main window is titled "Script - Badboy" and contains a menu bar (File, Edit, View, Tools, Preferences, Help) and a toolbar with various icons for file operations and testing.

The address bar shows the URL: <http://www.coperve.ufpb.br/sipervenet2/login/autenticar>.

The left pane, labeled "Script", shows a test script structure:

- Roteiro de Teste
  - Coperve
    - Login
      - http://www.coperve.ufpb.br/sipervenet2
      - /sipervenet2/login/autenticar
      - Check for text "CPF não pode ficar em branco" (marked with a red X)
      - Check for text "CPF não pode ficar em branco" (marked with a red X)
      - Page contains pattern CPF não pode ficar em branco (marked with a red X)
      - Page contains pattern CPF não pode ficar em branco (marked with a red X)
      - Check for text "Data de Nascimento não pode ficar em branco" (marked with a red X)
      - Check for text "Digite corretamente o texto que aparece na imagem" (marked with a green checkmark)
      - Check for text "Digite corretamente o texto que aparece na imagem" (marked with a green checkmark)

A red arrow points to the "Check for text 'CPF não pode ficar em branco'" item in the script.

The right pane shows the login page being tested. The page has a blue header with the text "Login". Below the header, there is a red banner with the text "Por favor, verifique seus dados: 3 erros." (Please check your data: 3 errors). Below the banner, there is a list of errors:

- CPF não pode ficar em branco
- Data de Nascimento não pode ficar em branco
- Digite corretamente o texto que aparece na imagem.

The login form includes fields for "CPF:" (a text input field), "Data Nascimento:" (three dropdown menus for "dia", "mês", and "ano"), and a reCAPTCHA image showing the text "francals llationi". Below the reCAPTCHA, there is a text input field and a button labeled "Entrar".

The bottom pane shows the "Checks" tab, which lists various checks available in the tool:

- Content Check
- Color Check
- Response Check
- Summary Check
- Variable Check
- JScript Check
- Window Caption Check

# Referências

48

- [GTSW] Grupo de Testadores de Software - <http://gtsw.blogspot.com> Acessado em Maio/2011.
- [Molinari, 2010] Molinari, L. “Inovação e Automação de Testes de Software”. 1ª Edição, Ed. Érica. São Paulo, 2010.
- [Selenium, 2011] Selenium - Ferramenta para Testes Funcionais Web. Disponível em: <http://seleniumhq.org>
- [Badboy, 2011] Badboy - Ferramenta para Testes Funcionais para Web. Disponível em: <http://www.badboy.com.au/>