Selenium - Automação

Cronograma

- Selenium – O que é?
- Instalando o Selenium IDE
- Instalando o Firebug
- Noções Básicas de HTML
- Estrutura do Selenium IDE
- Comandos Básicos
- Expressões Regulares
- Exercícios

Selenium – O que é?

Selenium é um conjunto de ferramentas para automação de browser web.

Atualmente é composto por 2 grupos:

Selenium WebDriver

Selenium IDE

Selenium WebDriver


Selenium WebDriver aceita comandos (Java, PHP, Ruby) e os envia para um navegador. Esses comandos são enviados aos navegadores através de driver específico podendo obter respostas da ação no navegador utilizado.

Características:

- Suíte de automação de testes de regressão
- Escalar e distribuir scripts entre vários ambientes
- Suporte a vários navegadores:
  - Internet Explorer
  - Safari
  - Opera
  - Chrome
  - Firefox

Selenium IDE

Selenium IDE é um ambiente de desenvolvimento integrado como extensão do Firefox onde permite gravar, editar e depurar os testes desejados.

Características:

- Permite gravar e reproduzir os testes em ambiente real de forma rápida
- Seleção de campos através de IDs, names ou expressão regular
- Possibilidade de salvar os testes em diversos formatos
- Plugin suporta apenas Firefox

Documentação

Selenium:

- http://docs.seleniumhq.org/docs/
- http://docs.seleniumhq.org/support/

- https://seleniumhq.wordpress.com/
- http://www.seleniumhq.org/docs/02_selenium_ide.jsp
- http://release.seleniumhq.org/selenium-core/1.0.1/reference.html

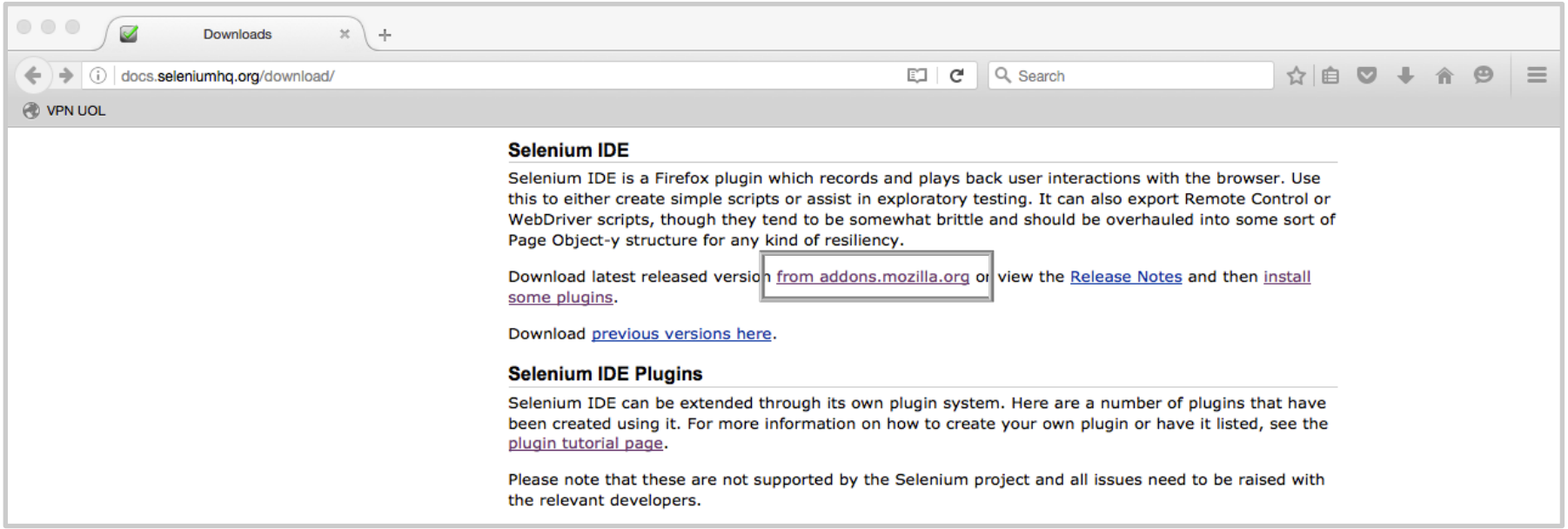
Expressão Regular:

- https://pt.wikipedia.org/wiki/Express%C3%A3o_regular
- http://regexr.com/

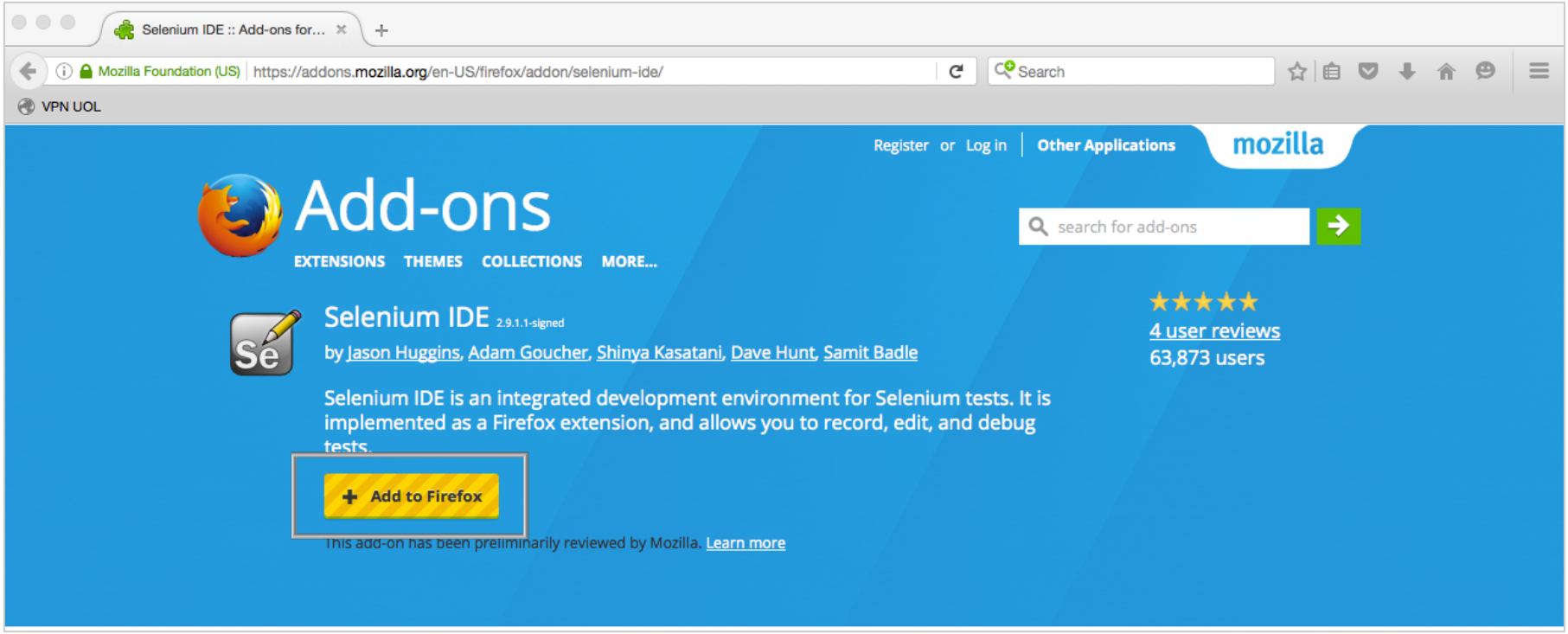
Instalando o Selenium IDE

Plugin disponível apenas para Firefox

http://docs.seleniumhq.org/download/



Instalando o Selenium IDE



Após adicionar a extensão ao Firefox, é necessário reiniciar o mesmo.

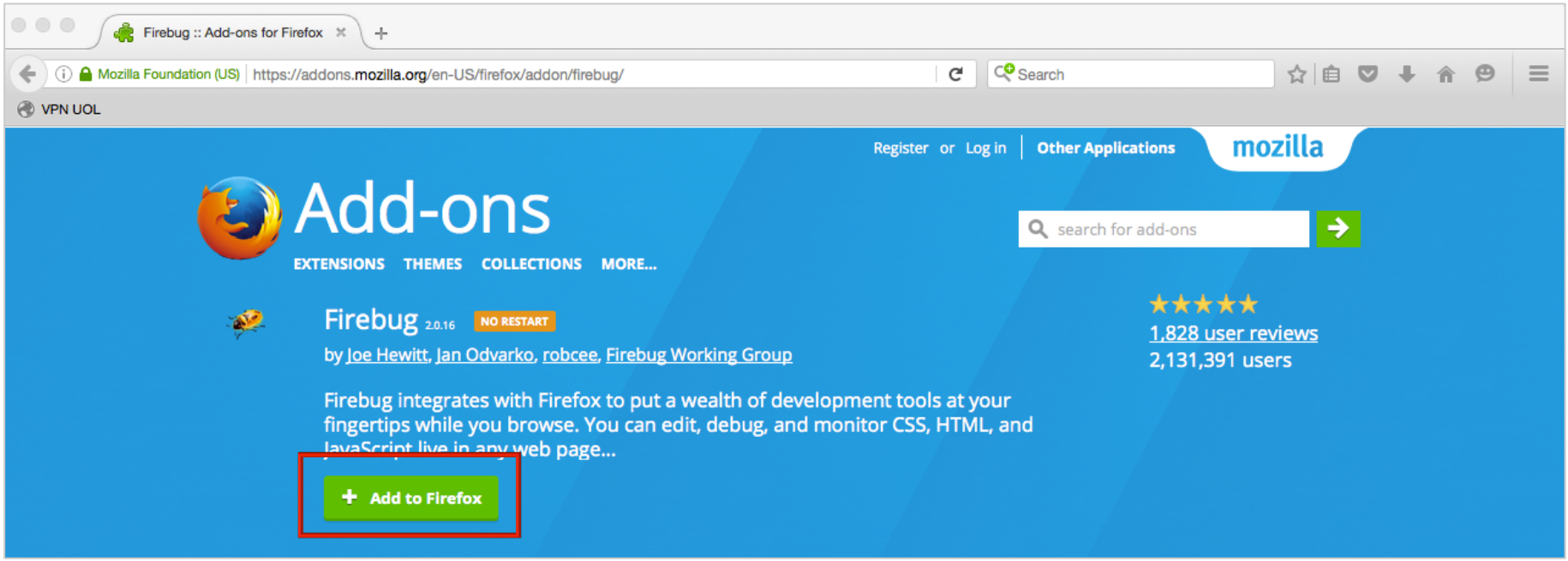
Instalando o Firebug

Ferramenta de desenvolvimento web onde é possível editar e monitorar CSS, HTML e JavaScript de uma página web.

Será utilizado apenas uma funcionalidade do Firebug para ajudar a identificar os elementos desejados na automação.

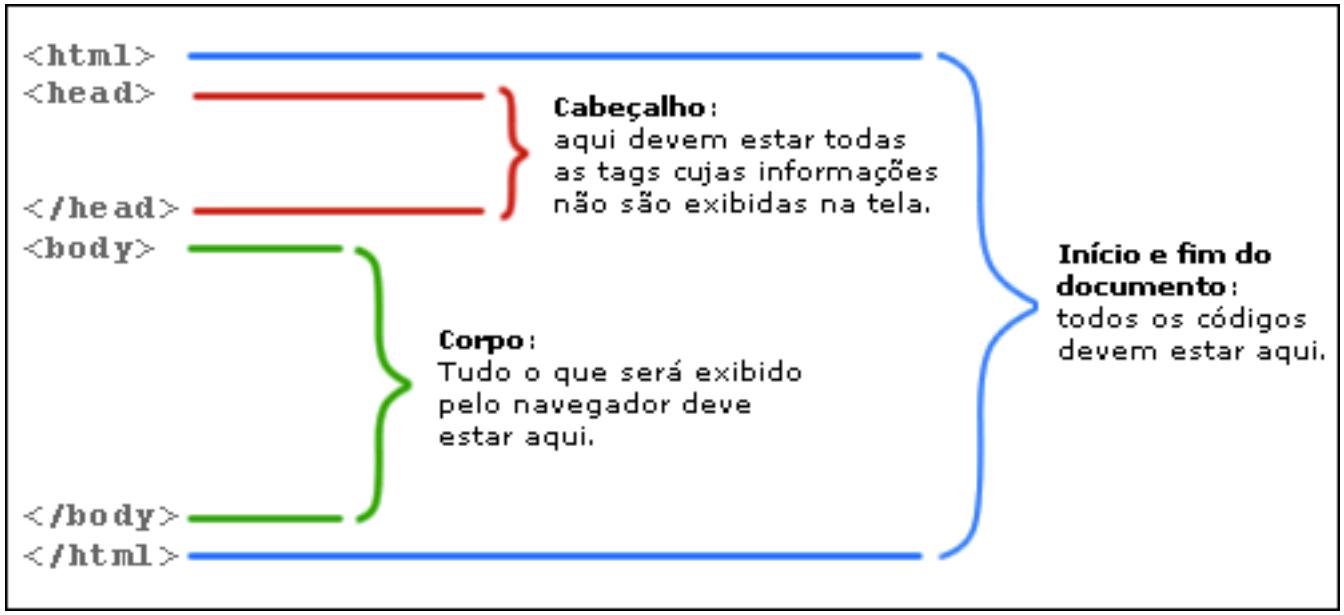
Instalando o Firebug

Acessar a página: https://addons.mozilla.org/en-US/firefox/addon/firebug/



Noções Básicas de HTML

A estrutura básica de um documento HTML é composta por três tags (html, head e body), deve estar presente em todos os documentos, e seguir exatamente a seguinte sequência:



Fonte: http://www.educacaopublica.rj.gov.br/oficinas/informatica/html/estrutura.htm

Noções Básicas de HTML

Tags

- *<html>* - identifica que o documento trata-se de um HTMl e não documento texto.
- *<head>* - cabeçalho do documento onde são inseridas dados e informações que não são exibidas na tela pelo navegador.
- *<title>* - título exibido pelos navegadores.
- *<body>* - corpo do documento. As informações abaixo dessa tag são exibidas pelos navegadores.
- *<p>* - parágrafo a ser exibido pelos navegadores.
- * e * - lista não ordenada (representado por bullets).
- * e * - lista ordenada (representado por 1,2,3 ou A,B,C).
- *<h1> até <h6>* - título de níveis.
- *<img>* - usado para exibição de imagens pelos navegadores.
- *<a>* - marca o link. Pode conter texto ou imagem.
- *<table>* - exibe uma tabela pelos navegadores.
  - *<tr>* - representa a linha da tabela.
  - *<td>* - representa a coluna da tabela.
- * * - agrupador de tags

Noções Básicas de HTML

Tags

- *<form>* - os campos abaixo dessa tag serão enviados para o destino configurado.
- *<input>* - define campos e botões nas páginas.
  - Atributos:
    - *type="text"* – campo de texto em uma única linha.
    - *type="password"* – representa um campo de senha.
    - *type="checkbox"* – campo de opções, mais de um item pode ser selecionado.
    - *type="radio"* – campo de escolha única.
    - *type="hidden"* – campo não visivel na tela, mas a informação contida no mesmo é enviado junto com os demais campos.
    - *type="submit"* – botão com função de envio do formulário.
    - *type="reset"* – botão com função de limpar todos campos do formulário.
- *<textarea>* - campo de texto com multiplas linhas.
- *<select>* e *<option>* - campo de seleção. <option> representa os itens.

Noções Básicas de HTML

Atributos: são informações para manipular o comportamento de uma tag.

Exemplos:

<a **href**="http://www.tqi.com.br">Clique Aqui

href – representa o destino quando o usuário clicar no texto "Clique Aqui"

<input **type**="text" **name**="usuario">

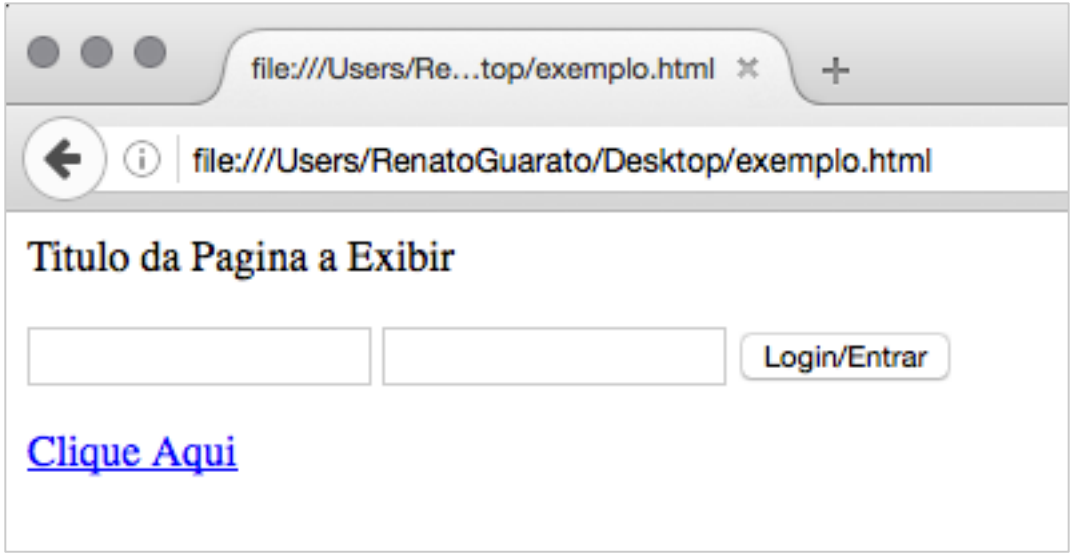
type – define o tipo do campo input que será exibido pelos navegadores.
name – representa o nome do campo a ser enviado pelo formulário.

<div **id**="componenteXPTO">

id – identificador único de uma tag na página. Não deve existir duas ou mais tags com o mesmo id.

Noções Básicas de HTML

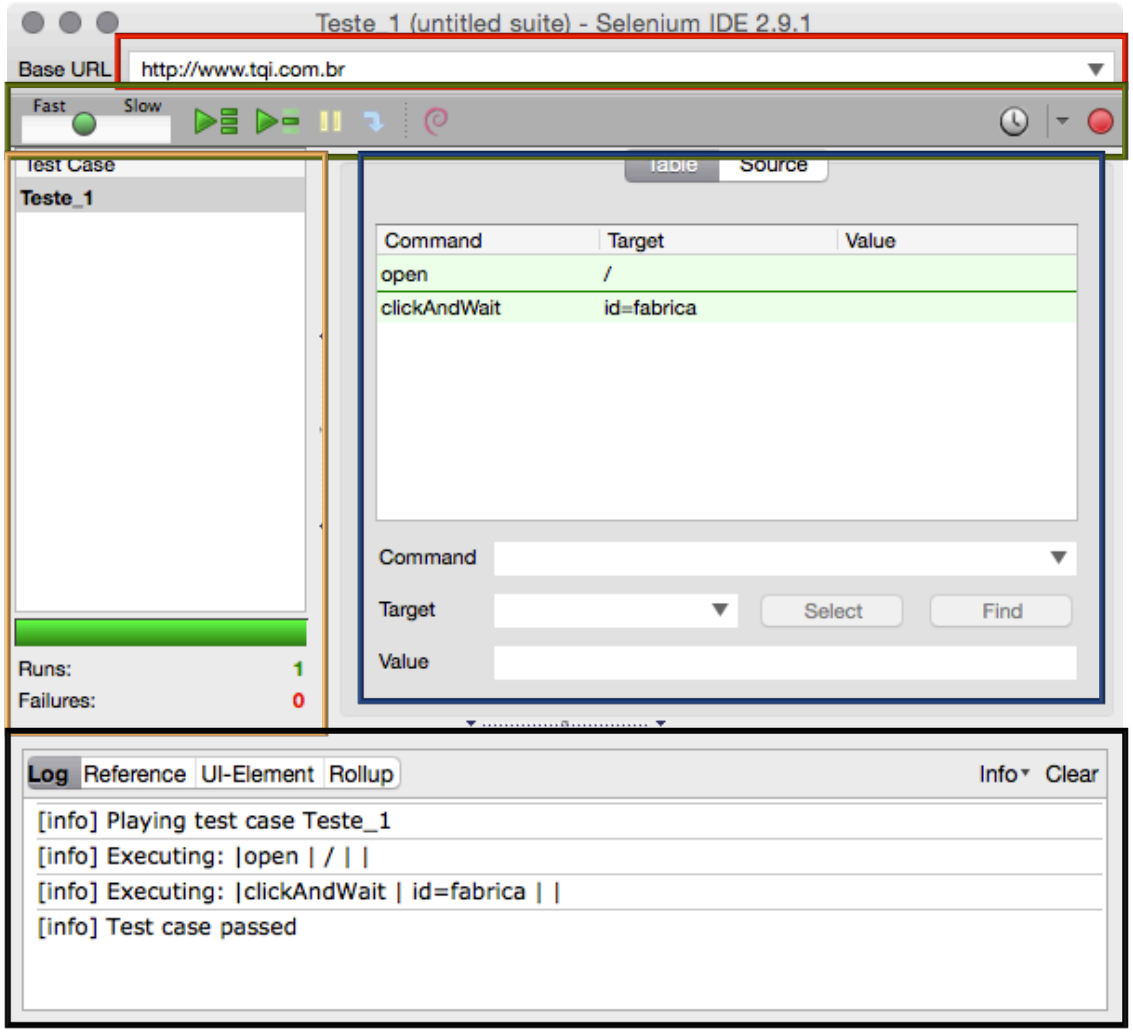
```
1  <html>
2      <body>
3          <p>Titulo da Pagina a Exibir</p>
4          <form id="formulario">
5              <input class="required" type="text" name="usuario"/>
6              <input class="required passfield" type="password" name="senha"/>
7              <input type="submit" name="login" value="Login/Entrar"/>
8          </form>
9          <a href="continue.html">Clique Aqui</a>
10     </body>
11 </html>
```



Estrutura do Selenium IDE

Barra de Menu

Suíte de Teste



URL Inicial

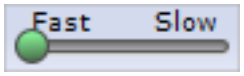
Estrutura e edição do script

Informações gerais

Estrutura do Selenium IDE

- **Base URL:** início da execução do script.
- **Barra de Menu:** controla a velocidade do script, inicia a execução do script. Usada em caráter de visualização das ações do script.
- **Test Case:** lista todos os scripts criados e onde é possível agrupar diversos scripts e salvar uma suíte de testes.
- **Estrutura de Script:** área onde é visualizado os comandos que estão sendo gravados.
- **Informações gerais:** apresenta log de execução, referência de comandos entre outras informações.

Estrutura do Selenium IDE



Controla a velocidade de execução do teste.

Executa todos os casos de teste da suíte.

Executa o caso de teste selecionado.

Interrompe a execução do caso de teste.

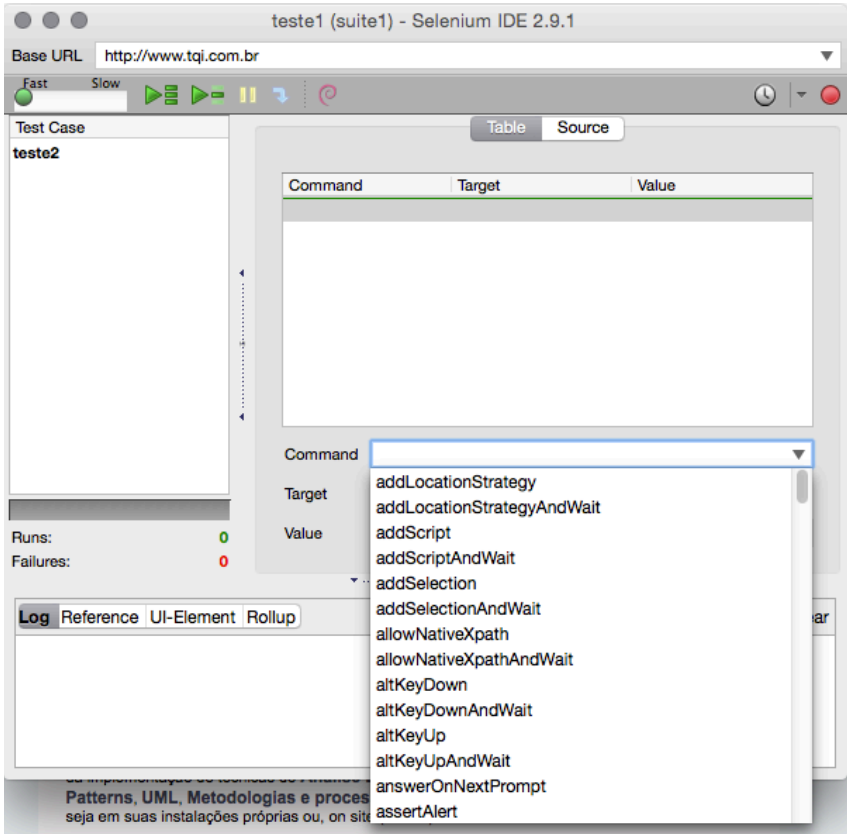
Inicia a gravação do script de teste.

Comandos Básicos

Existem uma série de comandos no Selenium IDE já prontos.

Os comandos do Selenium IDE se dividem em três grupos:

- Garantia ou verificação
- Ação
- Gerais



Comandos Básicos

Garantia ou Verificação

Esses comandos podem ser identificados iniciando com **verify…** ou **assert…**

Ex: **assertText** – garante que um determinado texto está contido no elemento.

Asserts: garante que uma certa informação existe no elemento, caso contrário, a execução do script é encerrada como falha.

Verify: verifica que uma certa informação existe no elemento, caso contrário, marca o comando como erro e continua a execução do script.

Comandos Básicos

Ação

Esses comandos mudam o comportamento da página.

Ex:

- **click** – simula o clique em botão, link ou qualquer outro elemento da página.
- **sendKeys** – simula a digitação letra por letra em qualquer campo texto.

Todos os comandos que terminam com **AndWait** aguardam até que a página esteja 100% carregada ou que o tempo total de espera seja ultrapassado quando esse comando é executado.

Tempo total de espera pode ser configurado no seguinte menu do Selenium IDE:

Options ➔ Options ➔ Default timeout value

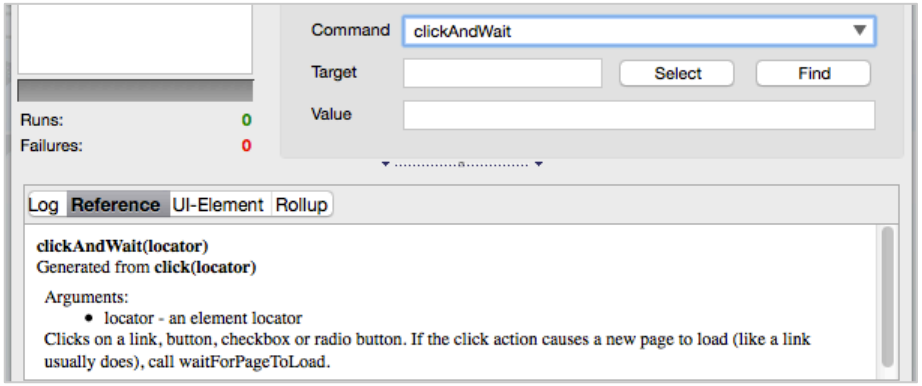
O valor configurado é em milisegundos. Ex: 30s = 30000ms

Comandos Básicos

Gerais

São os demais comandos dentro do Selenium IDE.

Sempre que for necessário obter informação de qualquer comando, basta selecioná-lo através do campo **Command** e visualizar a descrição do mesmo na aba **Reference**.



Comandos Básicos

Command	Target	Value
open	http://www.tqi.com.br	
Abre a url informada no campo **target**		
verifyText	Id=Extranet	Extranet
Verifica se o conteúdo do elemento com id=Extranet é igual a Extranet. Caso não seja, o comando é marcado com erro e o script continua com a execução.		
click	id=contato	
Executa a ação de click no elemento com id=contato.		
verifyElementPresent	link=Oportunidades	
Verifica a existência do elemento com link=Oportunidades na página.		

Comandos Básicos

Command	Target	Value
type	css=strong.locality	São Paulo
Incluir a informação de **Value** no elemento localizado em **Target**.		
verifyTitle	TQI	
Verifica se o título da página é o informado em **Target**.		
clickAndWait	id=contato	
Executa a ação de click no elemento com id=contato e aguarda a página ser totalmente carregada.		
captureEntirePageScreenshot	C:\\imagem.png	
Cria um screenshot da página em exibição salvando no arquivo informado em **Target**.		

Comandos Básicos

Command	Target	Value
store	Usuario	valor1
Cria uma variável com o nome **valor1** e coloca o valor **usuario** na mesma.		
storeValue	name=usuario	Valor2
Cria uma variável com o nome **valor2** e coloca o valor do campo no elemento de **Target**		
verifyEval	storedVars['valor1']==storedVars['valor2']	true
Realiza a comparação do conteúdo de duas variáveis: **valor1** e **valor2**. O resultado é comparado com **Value**.		

Comandos Básicos

```
1  <html>
2      <body>
3          <p>Titulo da Pagina a Exibir</p>
4          <form id="formulario">
5              <input class="required" type="text" name="usuario"/>
6              <input class="required passfield" type="password" name="senha"/>
7              <input type="submit" name="login" value="Login/Entrar"/>
8          </form>
9          <a href="continue.html">Clique Aqui</a>
10     </body>
11 </html>
```

Command	Target	Value
open	/	
type	name=usuario	usuario
store	usuario	valor1
storeValue	name=usuario	valor2
verifyEval	storedVars['valor1']==storedVars['valor2']	true

Command: verifyEval

Target: storedVars['valor1']==storedVars['valor2'] Select Find

Value: true

Comandos Básicos

Locators

São formas de identificar um elemento dentro da página web. É o principal ponto de entendimento do Selenium.

Tipos:

- **Id**: localização pelo ID do elemento
- **Nome**: localização pelo NOME do elemento
- **Xpath**: localização por funções ou por navegação
- **Links**: localização por um link na página
- **CSS**: localização por estilo

Comandos Básicos

Estratégia de Localização

A melhor estratégia de localização de elemento na página é:

1. Id
2. Nome
3. Link
4. CSS
5. Xpath

Embora **CSS** esteja em quarto lugar, o Selenium IDE identifica grande parte dos elementos por essa estratégia quando os mesmos não tenham **ID** ou **NOME**.

Mesmo **Xpath** seja o último da lista, existem funções que facilitam a localização de elementos na página.

Comandos Básicos

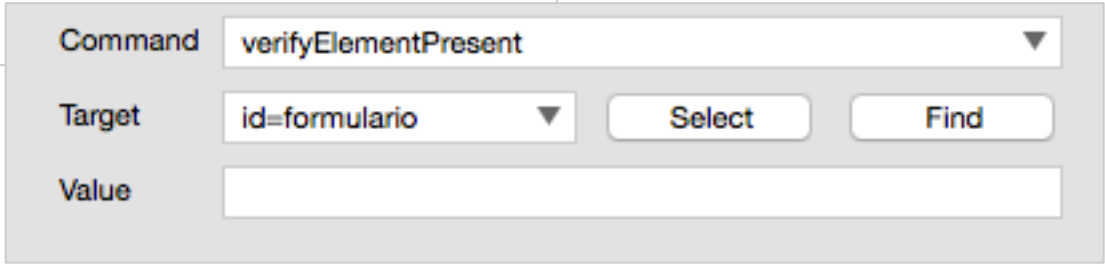
Localização por ID

Encontra o elemento pelo atributo **id**.

Ex:

id=formulario

```
1  <html>
2      <body>
3          <form id="formulario">
4              <input type="text" name="usuario"/>
5              <input type="password" name="senha"/>
6              <input type="submit" name="login" value="Login"/>
7          </form>
8      </body>
9  </html>
```



Comandos Básicos

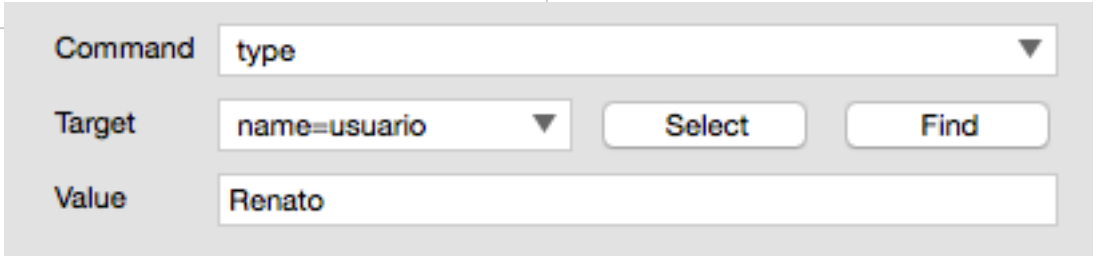
Localização por NOME

Encontra o elemento pelo atributo **name**.

Ex:

name=usuario

```
1  <html>
2      <body>
3          <form id="formulario">
4              <input type="text" name="usuario"/>
5              <input type="password" name="senha"/>
6              <input type="submit" name="login" value="Login"/>
7          </form>
8      </body>
9  </html>
```



Comandos Básicos

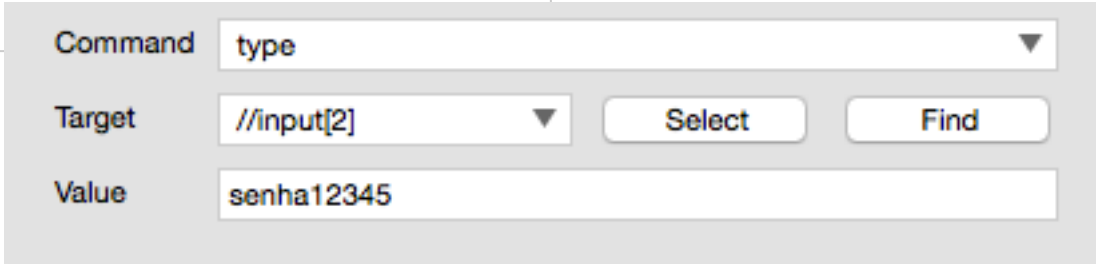
Localização por XPATH

Encontra o elemento através de navegação entre os elementos.

Ex:

//input[2] – localiza o segundo elemento do tipo **input**

```
1  <html>
2      <body>
3          <form id="formulario">
4              <input type="text" name="usuario"/>
5              <input type="password" name="senha"/>
6              <input type="submit" name="login" value="Login"/>
7          </form>
8      </body>
9  </html>
```



Comandos Básicos

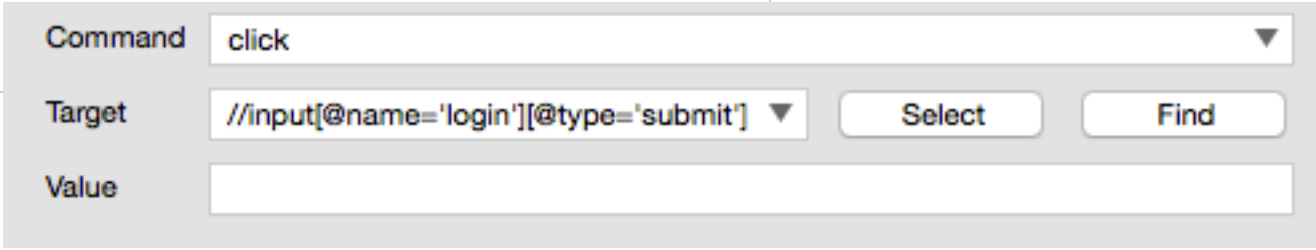
Localização por XPATH

Encontra o elemento através de funções.

Ex:

//input[@name='login'][@type='submit'] – localiza o elemento do tipo **input** com atributo **name**=login e o atributo **type**=submit

```
1  <html>
2      <body>
3          <form id="formulario">
4              <input type="text" name="usuario"/>
5              <input type="password" name="senha"/>
6              <input type="submit" name="login" value="Login"/>
7          </form>
8      </body>
9  </html>
```



Comandos Básicos

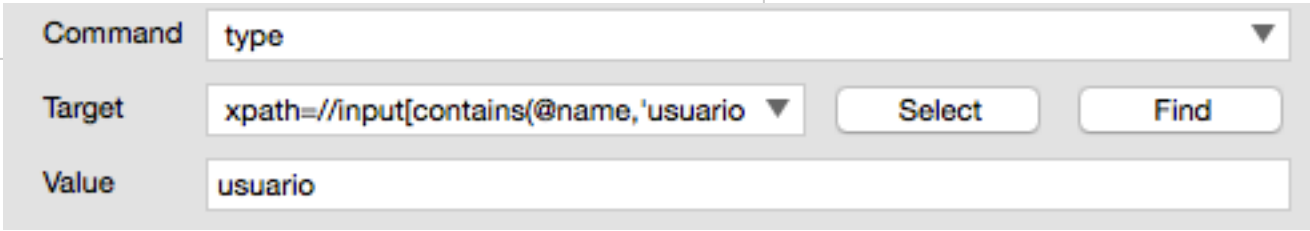
Localização por XPATH

Encontra o elemento através de funções.

Ex:

xpath=//input[contains(@name,'usuario')] – localiza o elemento do tipo **input** que contém no atributo **name** a palavra usuario

```
1  <html>
2      <body>
3          <form id="formulario">
4              <input type="text" name="usuario"/>
5              <input type="password" name="senha"/>
6              <input type="submit" name="login" value="Login"/>
7          </form>
8      </body>
9  </html>
```



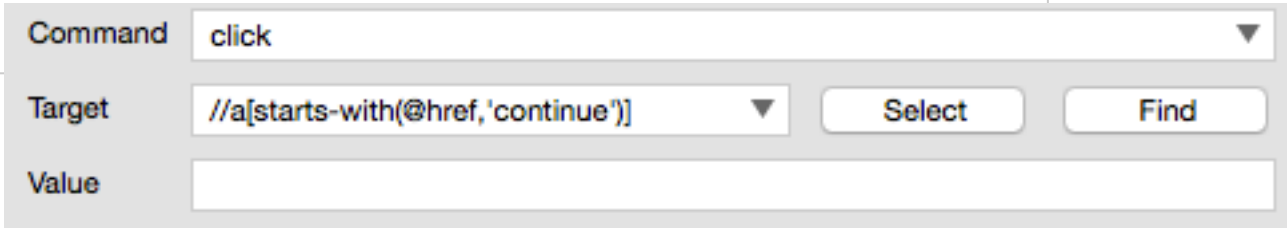
Comandos Básicos

Localização por XPATH

Para encontrar certos tipos de elementos na página podemos utilizar outra função.

starts-with: quando o elemento tem um mesmo prefixo.

```
1   <html>
2       <body>
3           <p>Titulo da Pagina a Exibir</p>
4           <form id="formulario">
5               <input class="required" type="text" name="usuario"/>
6               <input class="required passfield" type="password" name="senha"/>
7               <input type="submit" name="login" value="Login/Entrar"/>
8           </form>
9           <a href="continue.html">Clique Aqui</a>
10      </body>
11  </html>
```



Comandos Básicos

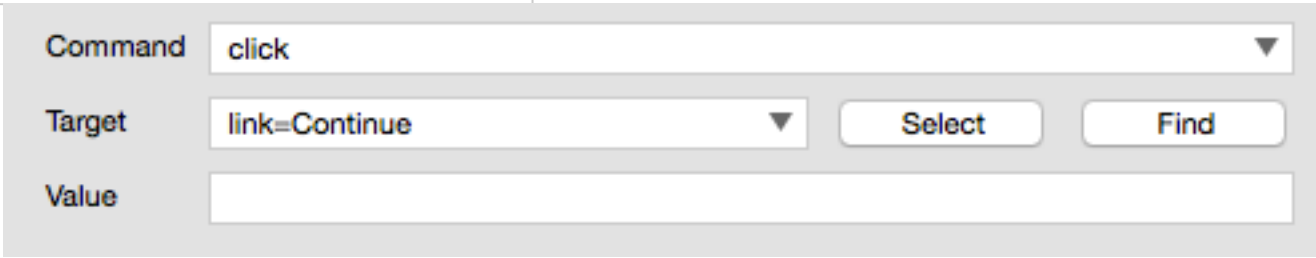
Localização por LINKS

Encontra o elemento com o atribulo **href** da página.

Ex:

link=Continue

```
1  <html>
2      <body>
3          <p>Deseja prosseguir com o cadastro?</p>
4          <a href="continue.html">Continue</a>
5          <a href="cancelar.html">Cancelar</a>
6      </body>
7  </html>
```



Comandos Básicos

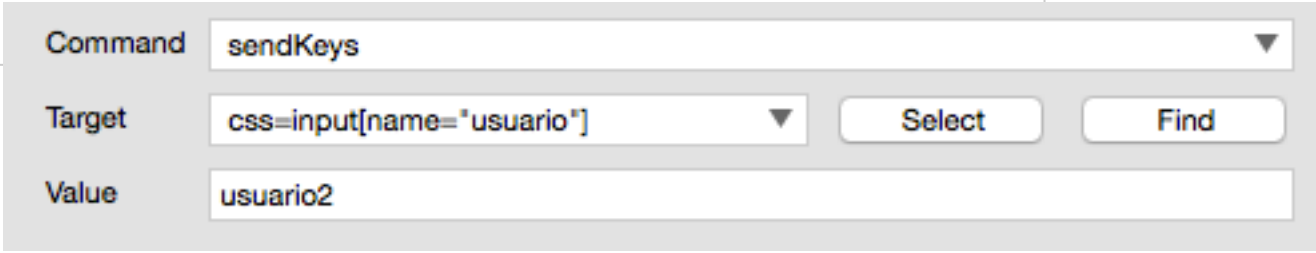
Localização por CSS

Encontra o elemento através do estilo CSS do atribulo **class**.

Ex:

css=input[name="usuario"]
css=input.passfield

```
1  <html>
2      <body>
3          <form id="formulario">
4              <input class="required" type="text" name="usuario"/>
5              <input class="required passfield" type="password" name="senha"/>
6              <input type="submit" name="login" value="Login"/>
7          </form>
8      </body>
9  </html>
```



Comandos Básicos

JavaScript

Comumente as páginas web utilizam funções javascript para executarem determinadas tarefas.

Existem diversos comandos com suporte a javascript:

- **addScript** – adiciona um script na página para execução do teste.
- **assertEval** – garante a avaliação de resultado positivo de um javascript.
- **assertNotEval** – garante a avaliação de resultado negativo.
- **runScript** – executa um javascript.
- **storeEval** – armazena o resultado de alguma ação via javascript.
- **waitForEval** – espera por algum resultado positivo de um javascript.

Comandos Básicos

JavaScript

Alertas são mensagens informativas em javascript que possuem apenas um botão de OK.

Confirmações são mensagens em javascript esperando uma determinada ação de "sim/não" ou "ok/cancelar".

Existem comandos para manipulação tanto de **Alertas** como **Confirmações**.

Os comandos de alertas possuem a palavra **alert** e as de confirmações possuem a palava **confirmation**.

Comandos Básicos

Alertas

- **assertAlert**: garante que um alerta apareceu validando pelo seu texto.
- **assertAlertNotPresent**: garante que um alerta não apareceu independente do seu texto.
- **assertAlertPresent**: garante que um alerta apareceu independente do seu texto.
- **storeAlert**: guarda a mensagem de um alerta.
- **storeAlertPresent**: guarda a mensagem de um alerta presente na página.
- **verifyAlert**: verifica se um alerta apareceu validando pelo seu texto.
- **waitForAlert**: espera por um alerta validando pelo seu texto.
- **waitForAlertNotPresent**: espera que um alerta não apareça na página.
- **waitForAlertPresent**: espera por qualquer alerta na página.

Comandos Básicos

Confirmações

Os mesmos comandos do Alerta servem para as Confirmações e tem as mesmas ações, porém com o nome **confirmation** ou invés de **alert**.

O único comando adicional é **chooseCancelOnNextConfirmation**, que realiza o clique no botão Cancelar.

Comandos Básicos

Popups

Os popups tem o mesmo tratamento que as janelas.

O Selenium trabalha com janelas, porém não entende a troca de abas.

Principais comandos para interagir com janelas:

- **close**: simula o clique no botão de fechar da janela ou popup.
- **deselectPopUp**: tira o foco atual de uma popup.
- **selectPopUp**: coloca o foco em uma popup.
- **selectWindow**: coloca o foco sobre uma janela ou popup.
- **waitForPopUp**: espera por uma determinada popup.
- **windowFocus**: coloca o foco sobre uma janela ou popup.

Expressões Regulares

Expressões regulares são muito úteis no Selenium para validação ou iteração com os elementos da página.

Existem três tipos de modos de uso:

- Global
- Expressões Regulares
- Exatos

Expressões Regulares

Global

Modo **Global** pode utilizar a maioria dos caracteres curinga de expressões regulares e é útil quando necessitamos interagir ou validar certos elementos que tem nomes em comum.

Utiliza o prefixo **glob** nos comandos.

```
1  <html>
2      <body>
3          <p>Titulo da Pagina a Exibir</p>
4          <form id="formulario">
5              <input class="required" type="text" name="usuario"/>
6              <input class="required passfield" type="password" name="senha"/>
7              <input type="submit" name="login" value="Login/Entrar"/>
8          </form>
9          <a href="continue.html">Clique Aqui</a>
10     </body>
11 </html>
```

Expressões Regulares

Expressões Regulares

As **Expressões Regulares** utilizam todos os recursos da expressão regular para localizar qualquer tipo de informação na página ou mesmo validá-la.

Utiliza o prefixo **regexp** nos comandos.

```
1  <html>
2      <body>
3          <p>Titulo da Pagina a Exibir</p>
4          <form id="formulario">
5              <input class="required" type="text" name="usuario"/>
6              <input class="required passfield" type="password" name="senha"/>
7              <input type="submit" name="login" value="Login/Entrar"/>
8          </form>
9          <a href="continue.html">Clique Aqui</a>
10     </body>
11 </html>
```

Expressões Regulares

Exatos

Os **Exatos** são utilizados quando interagimos com algum elemento que tenha algum tipo de caracter especial usado por expressões regulares ou globais.

Utiliza o prefixo **exact** nos comandos.

```
1  <html>
2      <body>
3          <p>Titulo da Pagina a Exibir</p>
4          <form id="formulario">
5              <input class="required" type="text" name="usuario"/>
6              <input class="required passfield" type="password" name="senha"/>
7              <input type="submit" name="login" value="Login/Entrar"/>
8          </form>
9          <div>(1 Brasil Real) * 1 U.S. dolar = 3,20</div>
10     </body>
11 </html>
```

Exercícios

Utilizar todos os conceitos abordados no treinamento.

Sugestões de sites para utilização nos exercícios:

- http://guaratorenato.wix.com/treinamento1
- http://guaratorenato.wix.com/treinamento2
- http://guaratorenato.wix.com/treinamento3
- http://guaratorenato.wix.com/treinamento4
- http://guaratorenato.wix.com/treinamento5

Dúvidas?

Renato Henrique Guarato
renato.guarato@tqi.com.br