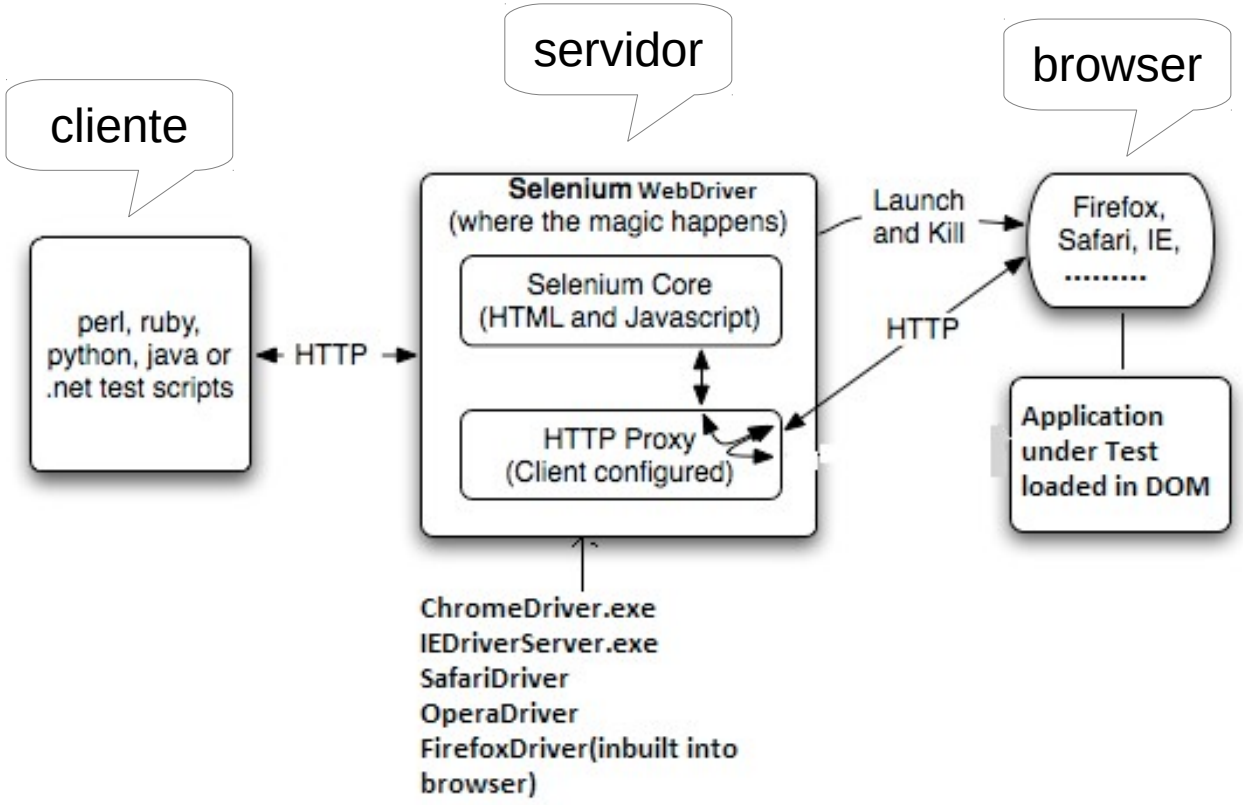
Selenium

- Teste de GUI para aplicações web
  - Mais famosa ferramenta de teste de GUI
- ***Selenium WebDriver***
  - *Codificar testes automatizados para GUI de aplicações Web*
  - *Várias linguagens suportadas*
    - *Java + JUnit*
- ***Selenium IDE***
  - Ferramenta de capture/replay
  - Chrome e Firefox add-on
  - Ver material complementar

https://www.selenium.dev/

Selenium Webdriver

- **Arquitetura**



Selenium Webdriver

- **Instalação**
- Passo-a-passo no Moodle

Selenium Webdriver

- **Primeiro Teste**

```
@Test
public void test01() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://www.google.com.br/");
    WebElement searchInput = driver.findElement( By.name("q") );
    searchInput.sendKeys("teste de software");
    searchInput.submit();

    WebDriverWait wait = new WebDriverWait(driver, 10);
    wait.until(new ExpectedCondition<Boolean>() {
        @Override
        public Boolean apply(WebDriver d) {
            return d.getTitle().toLowerCase().startsWith("teste");
        }
    });

    assertTrue(driver.getTitle().startsWith("teste de software"));
    driver.close();
}
```

Selenium Webdriver

- **Primeiro Teste**

```
@Test
public void test01() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://www.google.com.br/");
    WebElement searchInput = driver.findElement( By.name("q") );
    searchInput.sendKeys("teste de software");
    searchInput.submit();

    WebDriverWait wait = new WebDriverWait(driver, 10);
    wait.until(new ExpectedCondition<Boolean>() {
        @Override
        public Boolean apply(WebDriver d) {
            return d.getTitle().toLowerCase().startsWith("teste");
        }
    });

    assertTrue(driver.getTitle().startsWith("teste de software"));
    driver.close();
}
```

* Espefica onde está o Chrome driver
* Instancia a classe que interage com o browser

Selenium Webdriver

- **Primeiro Teste**

```
@Test
public void test01() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://www.google.com.br/");
    WebElement searchInput = driver.findElement( By.name("q") );
    searchInput.sendKeys("teste de software");
    searchInput.submit();

    WebDriverWait wait = new WebDriverWait(driver, 10);
    wait.until(new ExpectedCondition<Boolean>() {
        @Override
        public Boolean apply(WebDriver d) {
            return d.getTitle().toLowerCase().startsWith("teste");
        }
    });

    assertTrue(driver.getTitle().startsWith("teste de software"));
    driver.close();
}
```

* Espefica o timeout implícito que o selenium espera quando tenta recuperar um elemento via findElement(..)

Selenium Webdriver

- **Primeiro Teste**

```
@Test
public void test01() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://www.google.com.br/");
    WebElement searchInput = driver.findElement( By.name("q") );
    searchInput.sendKeys("teste de software");
    searchInput.submit();

    WebDriverWait wait = new WebDriverWait(driver, 10);
    wait.until(new ExpectedCondition<Boolean>() {
        @Override
        public Boolean apply(WebDriver d) {
            return d.getTitle().toLowerCase().startsWith("teste");
        }
    });

    assertTrue(driver.getTitle().startsWith("teste de software"));
    driver.close();
}
```

* vai para a página
* busca pelo elemento HTML com name igual a “q”
* preenche o campo com uma string
* envia o formulário

Selenium Webdriver

- **Primeiro Teste**

```
@Test
public void test01() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUr

    driver.get("https://www.google.com.br/");
    WebElement searchInput = driver.findElement( By.n
    searchInput.sendKeys("teste de software");
    searchInput.submit();

    WebDriverWait wait = new WebDriverWait(driver, 10);
    wait.until(new ExpectedCondition<Boolean>() {
        @Override
        public Boolean apply(WebDriver d) {
            return d.getTitle().toLowerCase().startsWith("teste");
        }
    });

    assertTrue(driver.getTitle().startsWith("teste de software"));
    driver.close();
}
```

* para verificar se a busca foi realizada, vamos checar o título da página
* precisamos esperar pelo elemento para que ele apareça

Selenium Webdriver

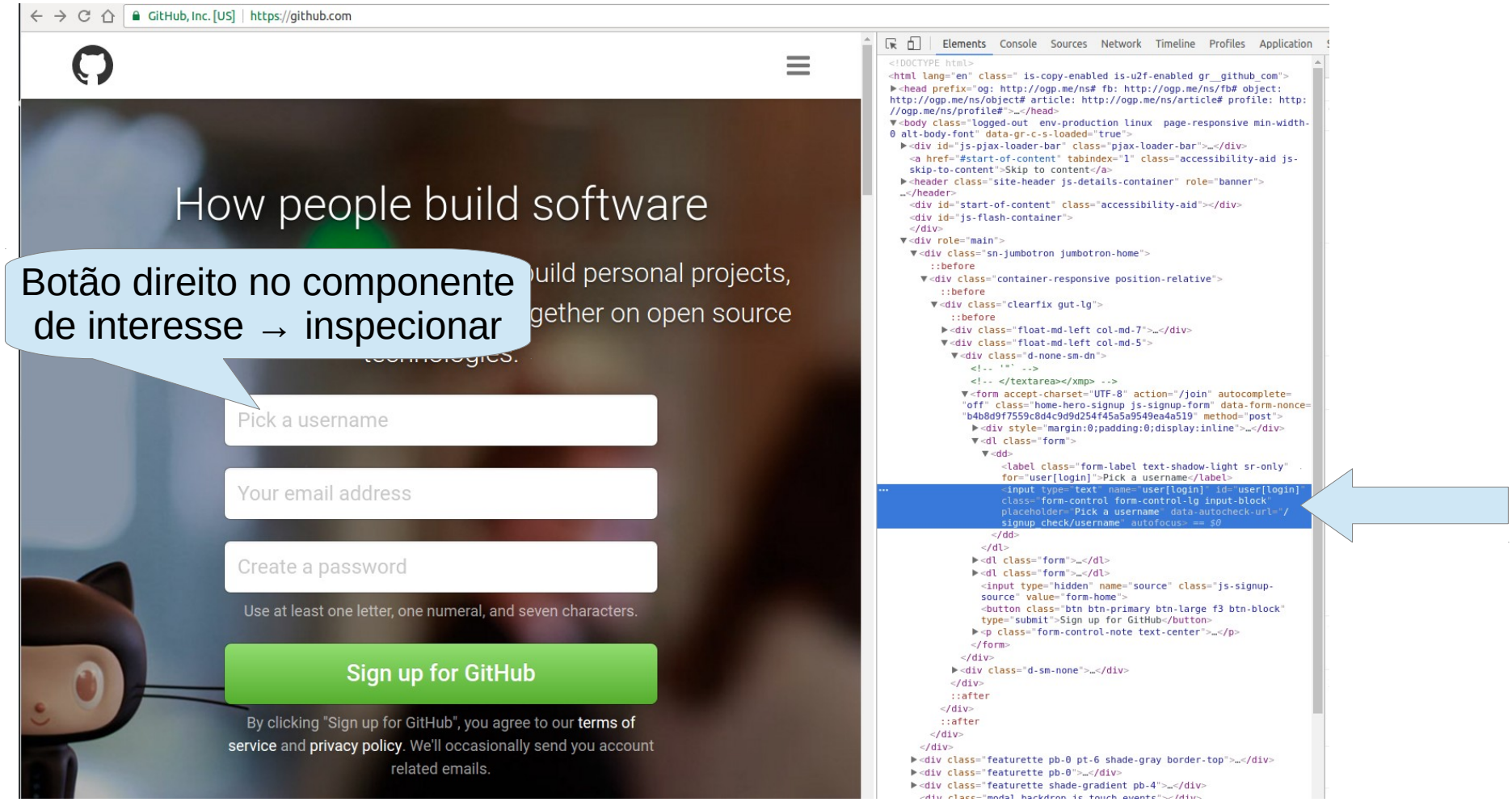
- **Primeiro Teste**

```
@Test
public void test01() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://www.google.com.br/");
    WebElement searchInput = driver.findElement( By.name("q") );
    searchInput.sendKeys("teste de software");
    searchInput.submit();

    WebDriverWait wait = new WebDriverWait(dri
    wait.until(new ExpectedCondition<Boolean>(
        @Override
        public Boolean apply(WebDriver d) {
            return d.getTitle().toLowerCase().startsW
        }
    });

    assertTrue(driver.getTitle().startsWith("teste de software"));
    driver.close();
}
```

* faço uma assertiva do JUnit, verificando se o título da página começa com a string fornecida.
* fecho a conexão (fecha o browser)

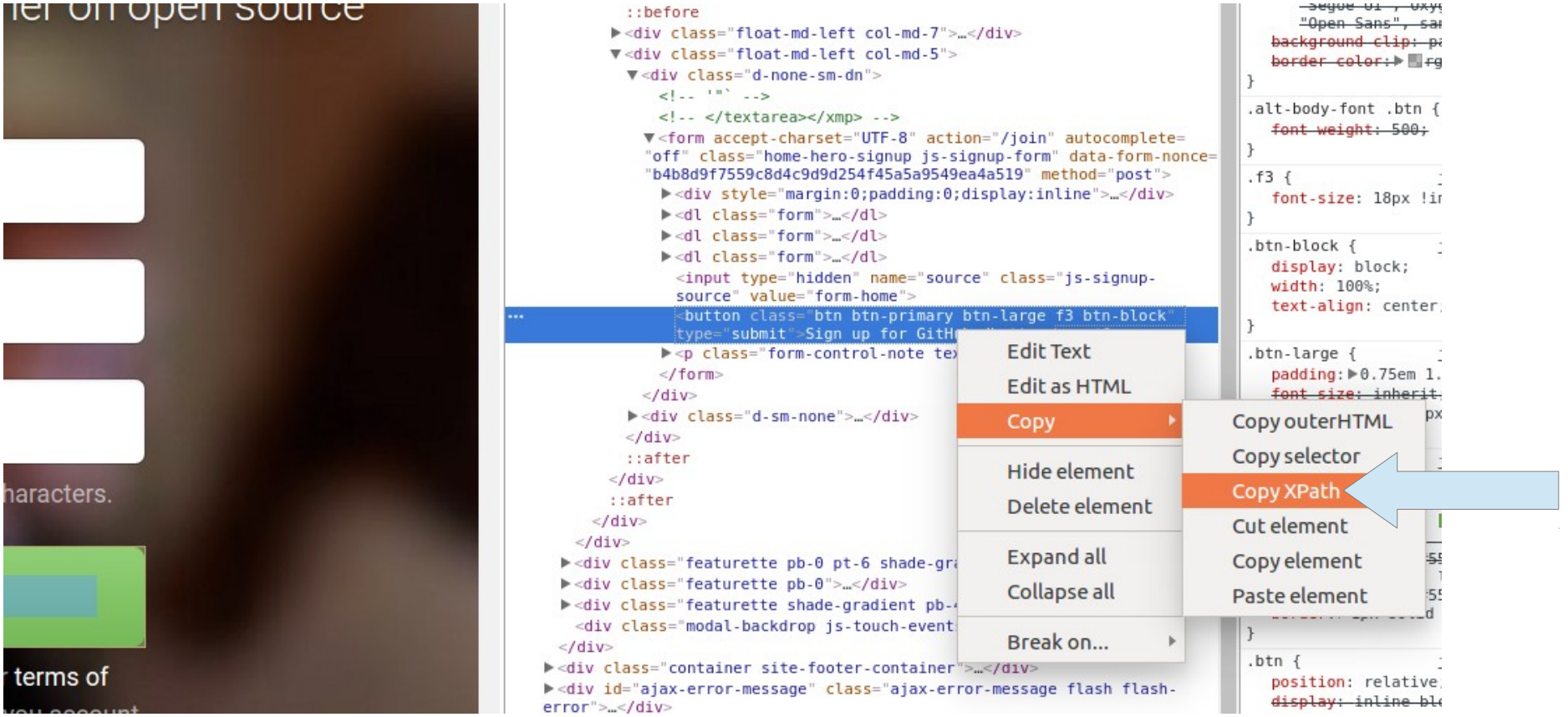
Selenium Webdriver

- **Como identificar os elementos (no Chrome)**
  - TagName, id, name, Xpath, ...



Selenium Webdriver

- **Pegando a expressão em XPath**



Selenium Webdriver

- **Github**

```
@Test
public void test02() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.get("https://github.com/");

    WebElement signUpButton = driver.findElement( By.xpath("____________________") );
    signUpButton.click();

    //check msg: "There were problems creating your account."
    WebElement errorMsg = driver.findElement( By.xpath("____________") );
    assertEquals("There were problems creating your account.", errorMsg.getText().trim());

    //check msg: "Login can't be blank"
    WebElement errorMsg02 = driver.findElement( By.xpath("____________") );
    assertEquals("Login can't be blank", errorMsg02.getText().trim());

    //fill the username
    WebElement username = driver.findElement( By.id("_____________") );
    username.sendKeys("andreendo22");

    //click on button "create account"
    WebElement caButton = driver.findElement( By.id("_____________") );
    caButton.click();

    try{
        errorMsg02 = driver.findElement( By.xpath("//*[@id=\"signup-form\"]/dl[1]/dd[2]") );
        fail();
    }
    catch(NoSuchElementException e) { }
    driver.close();
}
```

Selenium Webdriver

- **Github**

```
@Test
public void test02() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.get("https://github.com/");

    WebElement signUpButton = driver.findElement( By.xpath("/html/body/div[4]/div[1]/div/div/div[2]/div[1]/form/button") );
    signUpButton.click();

    //check msg: "There were problems creating your account."
    WebElement errorMsg = driver.findElement( By.xpath("//*[@id=\"signup-form\"]/div[2]") );
    assertEquals("There were problems creating your account.", errorMsg.getText().trim());

    //check msg: "Login can't be blank"
    WebElement errorMsg02 = driver.findElement( By.xpath("//*[@id=\"signup-form\"]/dl[1]/dd[2]") );
    assertEquals("Login can't be blank", errorMsg02.getText().trim());

    //fill the username
    WebElement username = driver.findElement( By.id("user_login") );
    username.sendKeys("andreendo22");

    //click on button "create account"
    WebElement caButton = driver.findElement( By.id("signup_button") );
    caButton.click();

    try{
        errorMsg02 = driver.findElement( By.xpath("//*[@id=\"signup-form\"]/dl[1]/dd[2]") );
        fail();
    }
    catch(NoSuchElementException e) { }
    driver.close();
}
```

Selenium Webdriver

- **IMC**

```
@Test
public void test03() {
    System.setProperty("webdriver.chrome.driver", "/home/utfpr/install/chromedriver");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
    driver.get("http://www.calcule.net/imc.calculo.indice.de.massa.corporal.a.php");

    WebElement altura = driver.findElement( By.id("___") );
    altura.sendKeys("180");

    WebElement peso = driver.findElement( By.id("___") );
    peso.sendKeys("7000");

    Select sexo = new Select( driver.findElement( By.name("___") ) );
    sexo.selectByVisibleText("____");

    WebElement calcButton = driver.findElement( By.name("___") );
    calcButton.click();

    WebElement resposta = driver.findElement( By.xpath("//*[@id=\"conteudo3\"]/table/tbody/tr/td[3]/p[4]/span/b[5]") );

    assertEquals("Normal", resposta.getText().trim());
    driver.close();
}
```

Bibliografia

- [Pfleeger07] S. L. Pfleeger, “Engenharia de Software: Teoria e Prática”, 2007.
- [Pressman11] R. S. Pressman, “Engenharia de Software: uma abordagem profissional”, 2011.
- [Sommerville03] I. Sommerville, “Engenharia de Software”, 2003.
- [Brooks87] “No Silver Bullet: Essence and Accidents of Software Engineering”, 1987.
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1663532
- [IEEE90] “IEEE Standard Glossary of Software Engineering Terminology”, 1990.
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342

Bibliografia

- [Myers] G. J. Myers, T. Badgett, C. Sandler, “The art of software testing”, 2012.
- [Pezze] M. Pezze, M. Young, “Teste e análise de software: Processos, princípios e técnicas”, 2008.
- [DMJ07] DELAMARO, Márcio Eduardo; MALDONADO, José Carlos; JINO, Mario. Introdução ao teste de software. Rio de Janeiro, RJ: Elsevier, 2007. 394 p. ISBN 9788535226348.
- [UUU] Materiais didáticos elaborados pelos grupos de engenharia de software do ICMC-USP, DC-UFSCAR e UTFPR-CP.

Bibliografia

- http://startingwithseleniumwebdriver.blogspot.com.br/2013/11/here-we-try-to-configure-selenium-web.html