

# INTEGRACIÓN E INTEROPERABILIDAD

---

## Práctica 3: *Wrapper* con Selenium

Integración e Interoperabilidad  
ETS Ingeniería Informática  
DSIC – UPV

*Curso 2017-2018*

## Índice

- Objetivos
- Referencias
- Arquitectura de aplicaciones Web
- Introducción a Selenium
- Selenium Web Driver
  - Localizar elementos en las páginas Web
  - Interacción con formularios
- Utilización desde Java
- Descripción de la entrega 2
- Ejemplo guiado



## Objetivos

- Presentar una librería para realizar *testing* funcional sobre aplicaciones Web
- Utilizar la librería para programar un *Wrapper* de extracción de información a partir de páginas Web.

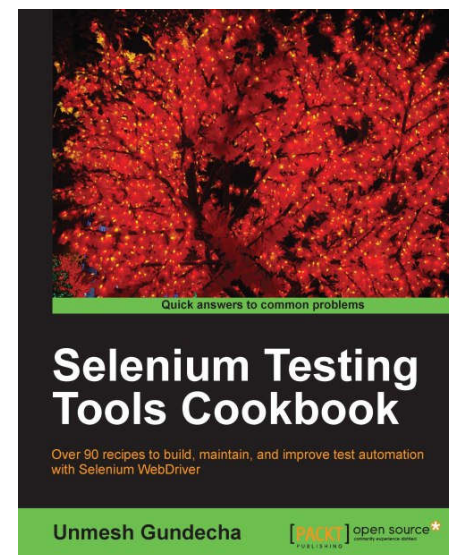
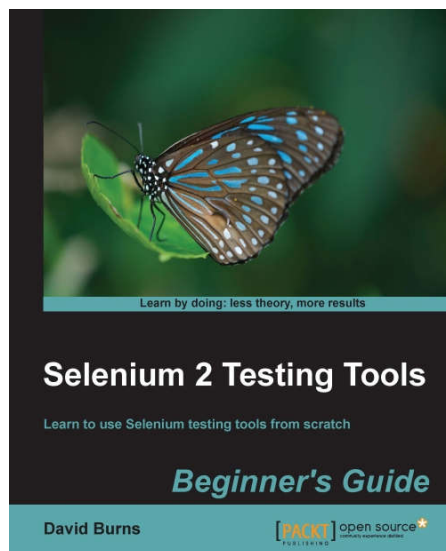
## Referencias

- El material está extraído de direcciones Web:
  - <http://www.guru99.com/accessing-forms-in-webdriver.html>
  - <http://www.qaautomation.net/?cat=7>
  - Stack Overflow
- Una buenas referencias bibliográficas son:

GURU<sup>99</sup>

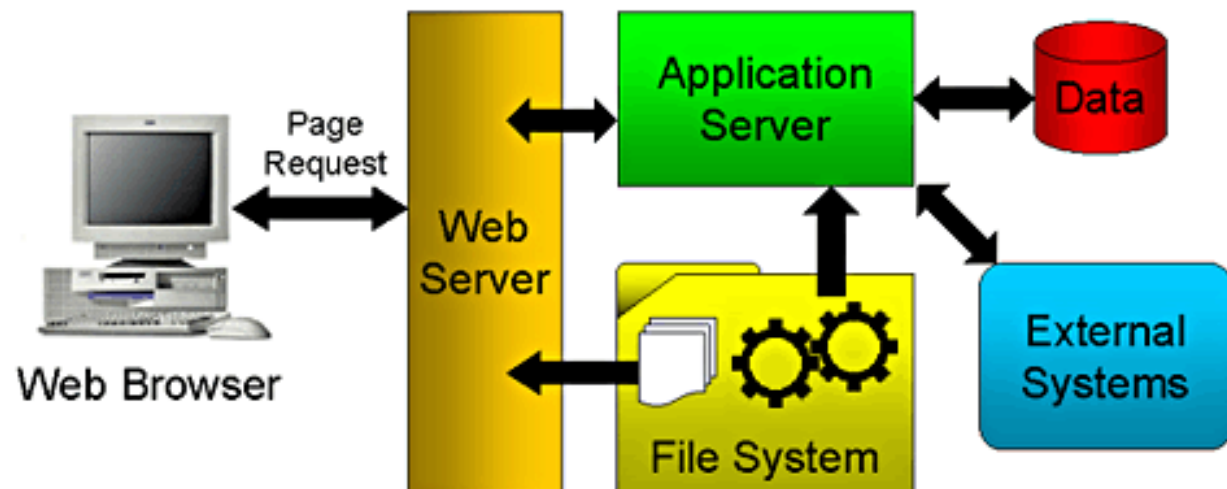
QA Automation

Improving software quality with automation.



## Arquitectura de aplicaciones Web

- Sistemas heterogeneos
  - *Front end*
    - *Browser*: IE, Firefox, Chrome, Safari...
  - *Servidor*
    - Servidor de aplicaciones
    - Servidor de base de datos
    - Sistema de archivos
    - .....



# Sistemas heterogeneos

- *Front end*
  - HTML, JavaScript, Adobe Flash.....

Página en el navegador

facebook

Email or Phone

☐ Keep me logged

Sign Up

It's free and a

First Nam

Last Nam

Your Ema

Re-enter Ema

New Passwor

I ar

Birthda

```
1 <html lang="en" id="facebook" class="no_js">
2 <head><meta charset="utf-8" /><script>function envFlush(a){function b(c){for(var d i
3 envFlush({"ffid1":"AcEhdku5-jWdFFPkIj0WbBGHn8S8s8ufqddxwgALSia8Ws7KKbRXQpmA96UUqScwbs , f11uz : AcEmC
4 <link rel="stylesheet" href="https://s-static.ak.fbcdn.net/rsrc.php/v2/yE/r/5k4qh71cfe.css" />
5 <link rel="stylesheet" href="https://s-static.ak.fbcdn.net/rsrc.php/v2/y8/r/ALKDrJ6G0nl.css" />
6 <link rel="stylesheet" href="https://s-static.ak.fbcdn.net/rsrc.php/v2/yE/r/u0J-X2jz094.css" />
7 <link rel="stylesheet" href="https://s-static.ak.fbcdn.net/rsrc.php/v2/y0/r/8e04id9zUP-.css" />
8 <link rel="stylesheet" href="https://s-static.ak.fbcdn.net/rsrc.php/v2/yY/r/u8iA3kXb8V1.css" />
9
10
11 <script src="https://s-static.ak.fbcdn.net/rsrc.php/v2/y1/r/GqLzR9PwXc2.js"></script>
12
13 <script>window.Bootloader && Bootloader.done(["j2Y1E"]);</script><script>new (require("ServerJS"))().
14 <script>Bootloader.setResourceMap({"zmU4p":{"type":"css","src":"https://s-static.ak.fbcdn.net/rsrc
15 Bootloader.enableBootload({"legacy:async-signal":{"resources":["JU6Q1"],"AsyncDOM":{"resources":["JU6
16 <script>require("InitialJSLoader").loadOnDOMContentLoadedReady(["JU6Q1","MX2xp","T8gbI","Abmov","UjXIQ"]);
17 <script>
18 Bootloader.configurePage(["zmU4p","M7bAm","jg0IV","D721J","VuNPD"]);
19 Bootloader.done(["jDr+c"]);
20
21
22 new (require("ServerJS"))().handle({"instances":["m187486_2",["Autosize","m187486_1"],[{"__m":"m187486_1"}]});
23
24 onloadRegister_DEPRECATED(function () {useragentcm();});
25 onloadRegister_DEPRECATED(function () {
26   if (typeof RegUtil == 'undefined') {
27     RegUtil = function() {};
28   }
29   RegUtil.captcha_class="ReCaptchaCaptcha";
30   RegUtil.recaptcha_public_key="6LfDxsYSAAGGLBGaRurayNbn1G8";
31 });
32 onloadRegister_DEPRECATED(function () {RegistrationBootload();});
33 onloadRegister_DEPRECATED(function () {startFormTypeABTest();});
34 onloadRegister_DEPRECATED(function () {$("#upfspt3").value = tz_calculate(1352931874);});
```

HTML

JavaScript

Código subyacente

# Sistemas heterogéneos

## • Servidor

- Puede estar escrito en PHP, Java, C#...
- Se comunica con el servidor de base de datos en SQL

ManageUsers.php

PHP Script

PHP

SQL

HTML

SQL

```
1 <?php
2
3 // Add the new user if one is being added //
4 if($_POST["adduser"] == 1 && $_POST["password"] != "" && $_POST["type"] != "")
5 {
6     $query = mysql_query(
7         "SELECT username FROM users"
8     )
9     or die("ManageUsers.php: Unable to get list of users - " . mysql_error());
10
11 // Make sure that the supplied username does not already exist in the database //
12 while( $userlist = mysql_fetch_row($query) )
13 {
14     if($_POST["username"] == $userlist[0])
15     {
16         die(
17             "<br><br><h1 align='center'><font color='red'>Username already exists!<br>
18             <br>
19             <form name='uhoh' action='./index.php' method='POST'>
20             <center><input type='button' value='&nbsp;Back&nbsp;' onClick='document.uhoh.page2.value=10;document.uhoh.submit();'></center>
21             <input type='hidden' name='page2' value='$page2'>
22             <input type='hidden' name='logout'>
23             <input type='hidden' name='page' value='$page'>
24             </form>"
25         );
26     }
27 }
28
29 // If all is good, insert the new user into the database //
30 $query = mysql_query(
31     "INSERT INTO users VALUES('' , '$_POST[username]', '$_POST[password]', '$_POST[type]'"
32 ) or die("ManageUsers.php: Unable to insert new user - " . mysql_error());
33 }
```

## *Testing el front end*

- *Testing* automatizado de aplicaciones web
  - El robot es capaz de:
    - Proporcionar valores de entrada
    - Simular acciones de usuario
- Selenium
  - Un conjunto de herramientas que automatizan el *testing* de aplicaciones web sobre diversas plataformas.
  - Puede simular interacciones de usuario en el navegador



# Selenium

## •Proyectos

### Selenium Projects

Selenium has many projects that combine to make a versatile testing system.

#### Selenium IDE



[Selenium IDE](#) is a Firefox add-on that makes it easy to record and playback tests in Firefox 2+. You can even use it generate code to run the tests with Selenium Remote Control.

#### Selenium Remote Control



[Selenium Remote Control](#) is a client/server system that allows you to control web browsers locally or on other computers, using almost any programming language and testing framework.

#### Selenium WebDriver



[Selenium WebDriver](#) can drive a browser natively either locally or on remote machines.

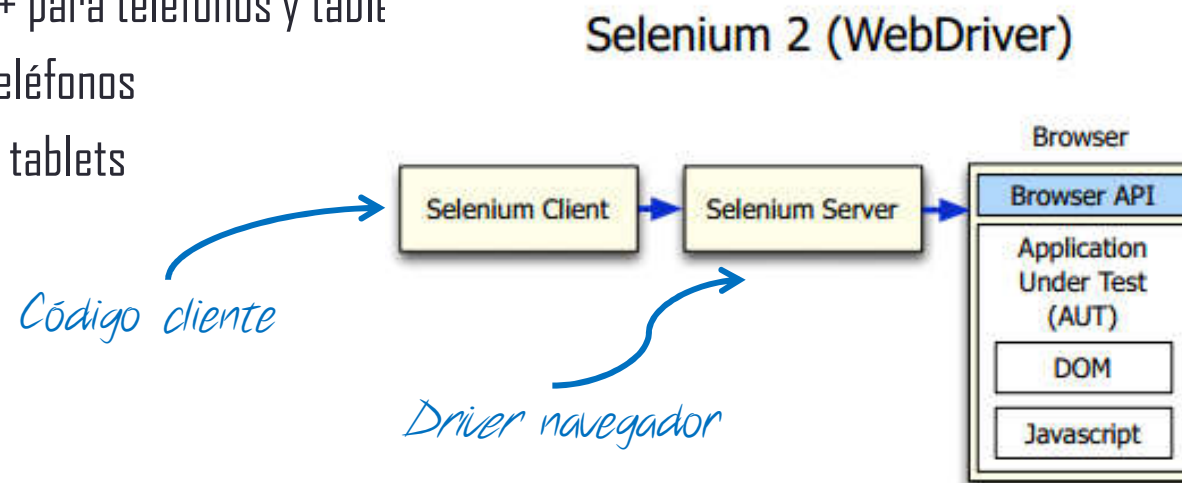
#### Selenium Grid



[Selenium Grid](#) takes Selenium Remote Control to another level by running tests on many servers at the same time, cutting down on the time it takes to test multiple browsers or operating systems.

## Selenium WebDriver (Selenium 2)

- Selenium-WebDriver soporta diferentes navegadores en multiples plataformas
  - Google Chrome 12.0.712.0+
  - Internet Explorer 6+
  - Firefox 3.0+
  - Opera 11.5+
  - Android – 2.3+ para teléfonos y tablets
  - iOS 3+ para teléfonos
  - iOS 3.2+ para tablets



## Selenium WebDriver

- WebDriver proporciona una interface de programación simple y uniforme
  - Los mismos scripts de WebDriver pueden ser ejecutados sobre diferentes plataformas
- Soporta diversos lenguajes de programación
  - Java, C#, Python, Ruby, PHP, Perl...
- Es eficiente
  - WebDriver hace uso del soporte nativo de cada navegador para implementar la automatización.

## Utilización de Selenium WebDriver

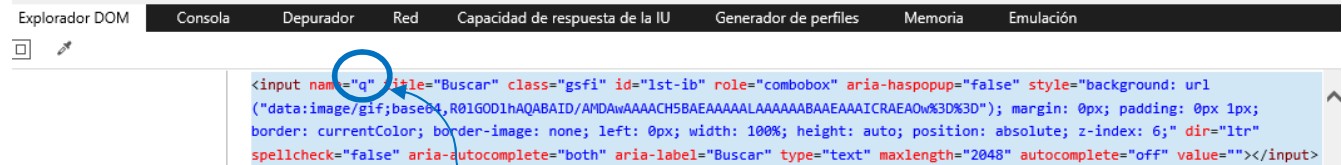
- (1) Ir a la página
- (2) Localizar un elemento
- (3) Realizar alguna acción con ese elemento
- .....
- (i) Localizar un elemento
- (i+1) Realizar alguna acción con ese elemento
- (i+2) Verificar / Assert el resultado

## Demo: Verificar título

Código incompleto

```
public static void main( String[] args )
{
    // Crear una nueva instancia del driver de Firefox
    WebDriver driver = new FirefoxDriver();
    // (1) Ir a la página
    driver.get("http://www.google.com");
    // (2) Localizar an elemento
    WebElement element = driver.findElement(By.name("q"));
    // (3-1) Introducir la cadena de búsqueda
    element.sendKeys("Universidad Politécnia de Valencia");
    // (3-2) Enviar el formulario
    element.submit();
    // (3-3) Esperar 10 segundos para una condición
    WebDriverWait waiting = new WebDriverWait(driver, 10);
    waiting.until( ExpectedConditions.presenceOfElementLocated( By.id("pnnext") ) );
    // (4) Comprobar el título de la página de respuesta
    if( driver.getTitle().equals("Universidad Politécnica de Valencia- Buscar con Google") )
        System.out.println("PASA");
    else System.err.println("FALLA");
    // Cerrar el navegador
    driver.quit();
}

<title>Universidad Politécnica de Valencia - Buscar con Google</title>
```



Elemento a buscar



```
<a class="pn" id="pnnext" style="text-align: left;" href="/search?q=universidad+polit%C3%A9cnica+de+valencia&biw=1600&bih=503&ei=5T_7V4_-M8qbgAbh5ZbIAw&sqi=2&start=10&sa=N">
  <span class="csb ch" style="background: url("/images/nav_logo242.png") no-repeat -96px 0px;
  <span style="margin-left: 53px; display: block;">Siguiente</span>
```

## Localizar elementos en páginas web

- Los localizadores le dicen a WebDriver los elementos (TextBox, Buttons, etc) sobre los que operará.
- La localización es imprescindible para poder crear el código de automatización.
- Selenium permite los siguientes tipos de localizadores:

- ID 
- Abrir la página para buscar el localizador del elemento*

- Name
- Class
- Link Text
- CSS Selector


- Tag e ID
- Tag y class
- Tag y atributo
- Tag, class, y atributo
- Inner text

- ~~DOM (Document Object Model)~~

- ~~• getElementById~~
- ~~• getElementsByName~~
- ~~• dom:name~~
- ~~• dom: index~~

- XPath

```
<form name="loginForm">
<label for="username">UserName: </label> <input
type="text"
id="userName" /><br/>
<label for="password">Password: </label> <input
type="password" id="password" /><br/>
<input name="login" type="submit" value="Login" />
</form>
```



*Código cliente Java de acceso al elemento*

```
WebElement username = driver.findElement(By.id("userName"));
```

## Localizar elementos en páginas web

- Herramientas para encontrar localizadores
- Firebug



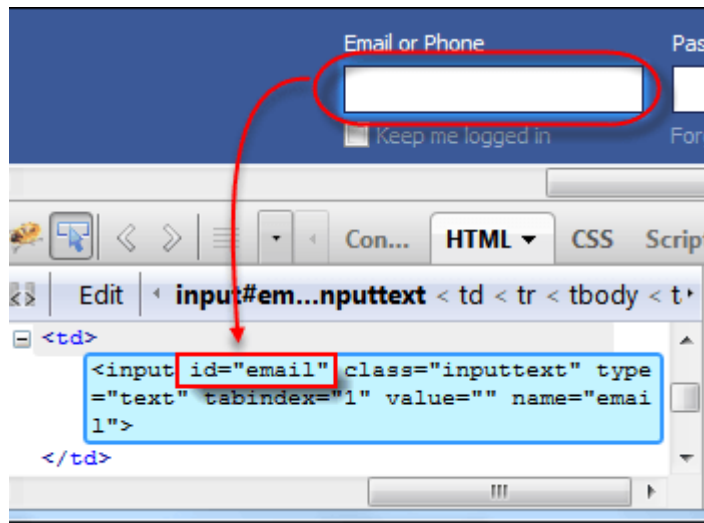
- O en el navegador activar las herramientas de desarrollo



- Firefinder <https://addons.mozilla.org/firefox/addon/firefinderfor-firebug>. Para testear CSS y Xpath en la página

## Localizar elementos en páginas web

- Localizar por **ID**: la forma más usual ya que se supone que el id es único




- A veces no existe el id o bien es generado dinámicamente, con lo cual debe buscarse una alternativa



# Localizar elementos en páginas web

- Localizar por **Name**



The screenshot shows the Mercury Tours website. On the left is a navigation menu with links: Home, Flights, Hotels, Car Rentals, Cruises, Destinations, and Vacations. The main content area features a 'Featured Destination' for Aruba, including a globe and a beach image. To the right, there's a 'Find A Flight' section with a sign-in form. The form has fields for 'User Name' and 'Password', and a 'Sign' button. A red arrow points from the 'User Name' input field to its HTML code in a separate window.

Oct 10, 2016

**Find A Flight**

Registered users can **sign-in** here to find the lowest fare on participating airlines.

User Name:

Password:

**Sign**

```
<td align="right">  
<td width="112">  
  <input name="userName" size="10" type="text">  
</td>  
</tr>
```

```
WebElement username =  
    driver.findElement(By.name("userName"));
```

## Localizar elementos en páginas web

- Localizar por **Class**

```
<form name="loginForm">
  <label for="username">UserName: </label> <input type="text"
  class="username" /></br>
  <label for="password">Password: </label> <input
  type="password" class="password" /></br>
  <input name="login" type="submit" value="Login" />
</form>
```

- Podemos utilizar el atributo class para localizer por código el element

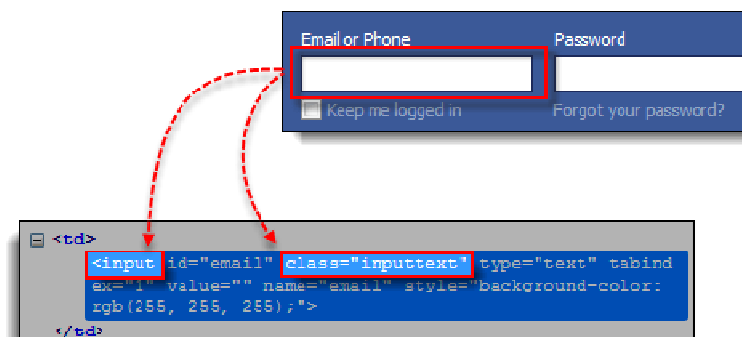
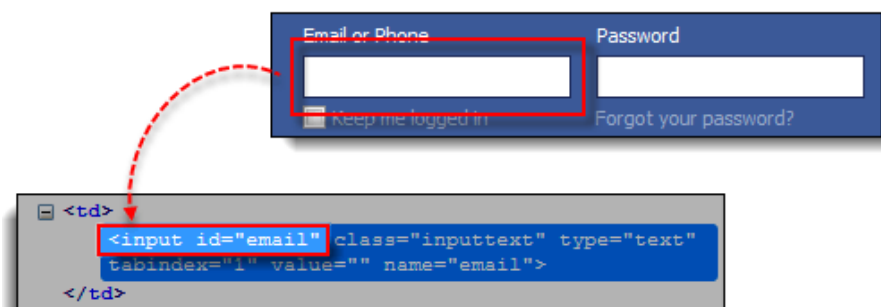
```
WebElement username =
    driver.findElement(By.className("username"));
WebElement password =
    driver.findElement(By.className("password"));
```

- Si existe un **enlace** en la página podemos acceder por el texto que se muestra en la misma.

```
WebElement gmailLink = driver.findElement(By.linkText("GMail"));
```

## Localizar elementos por CSS

- Podemos acceder por el **tag html** y por el **id** del elemento o bien por el **tag** y la **clase**



- Por tag y por id:

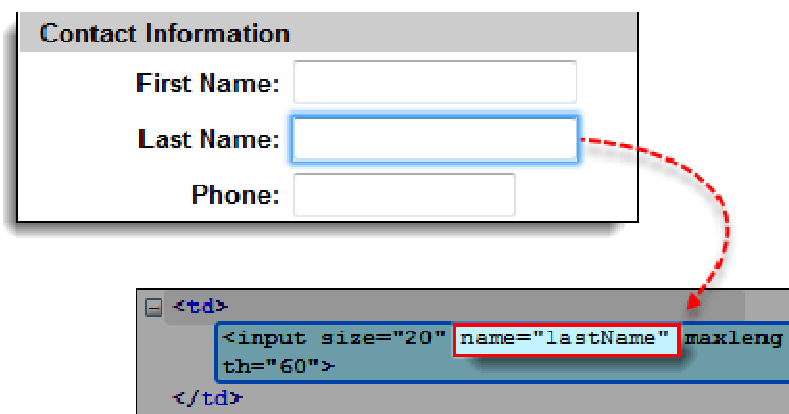
```
WebElement userName =  
    driver.findElement(By.cssSelector("input#email"));
```

- Por tag y clase

```
WebElement userName =  
    driver.findElement(By.cssSelector("input.inputtext"));
```

## Localizar elementos por CSS

- Tag y atributo: Podemos utilizar un **tag** de html y un par **atributo – valor** del elemento.



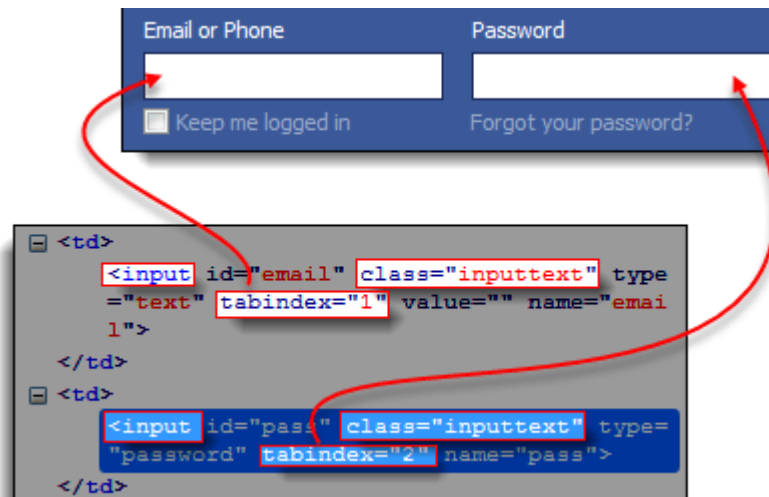
```
WebElement userName =  
    driver.findElement(By.cssSelector("input[name=lastName]"));
```

- A veces es necesario utilizar combinaciones de atributo- valor

```
WebElement previousButton =  
    driver.findElement(By.cssSelector("input[type='submit'][value='Login']"));
```

## Localizar elementos por CSS

- Tag, clase y atributo

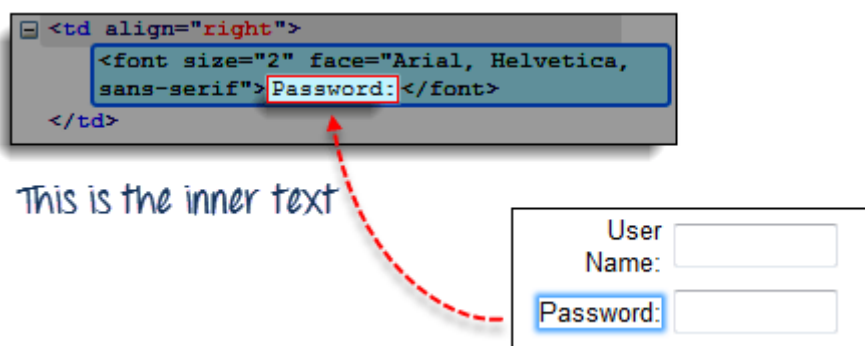


- Para acceder por código al selector se utiliza

```
WebElement previousButton =  
driver.findElement(By.cssSelector("input.inputtext[tabindex=1]"));
```

## Localizar elementos por CSS

- Texto interno del elemento: podemos acceder a las etiquetas de texto que muestra la página HTML



- Desde el código cliente se puede utilizar

```
WebElement cssele = driver.findElement(By.cssSelector("font:contains('Password')"));
```

## Localizar elementos por CSS

- CSS admite expresiones del siguiente tipo

Sintaxis	Descripción
<code>^=</code>	Encuentra el elemento que empieza con el valor pasado
<code>\$=</code>	Encuentra el elemento que termina con el valor pasado
<code>*=</code>	Encuentra el elemento que contiene al elemento pasado

```
ArrayList<WebElement> resultsDiv =  
    (ArrayList<WebElement>)  
driver.findElements(By.cssSelector("*[class^='col-xs-6 col-sm-4 col-md-4  
col-lg-4']"));
```

## Localizar elementos por XPath

- XPath se utiliza para localizar elementos dentro de un archivo XML. XPath permite realizar consultas sobre DOM como si fuese un archivo XML.
- Encontrar elemento por **camino absoluto**

```
WebElement userName =  
    driver.findElement(By.xpath("html/body/div/div/form/input"));
```

- Encontrar elemento por **camino relativo**

```
WebElement userName = driver.findElement(By.xpath("//input"));
```

- Encontrar elementos por **índice**

```
WebElement userName = driver.findElement(By.xpath("//input[2]"));
```

- Encontrar elemento por **valores de atributos**

```
WebElement userName =  
    driver.findElement(By.xpath("//input[@id='username']"));
```



## Localizar elementos por XPath

- En algunas situaciones es necesario combinar los **valores de varios atributos**

```
WebElement previousButton = driver.findElement(By.xpath  
    ("//input[@type='submit'][@value='Login']"));
```

- Xpath admite los operadores and y or

```
WebElement previousButton = driver.findElement(By.xpath  
    ("//input[@type='submit' and @value='Login']"));
```

```
WebElement previousButton = driver.findElement(By.xpath  
    ("//input[@type='submit' or @value='Login']"));
```

- Ejemplos adicionales de localización por código en:

[http://www.seleniumhq.org/docs/03\\_webdriver.jsp](http://www.seleniumhq.org/docs/03_webdriver.jsp)

- Java API WebDriver

<http://seleniumhq.github.io/selenium/docs/api/java/>

<https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/WebDriver.html>

## Interacción con formularios

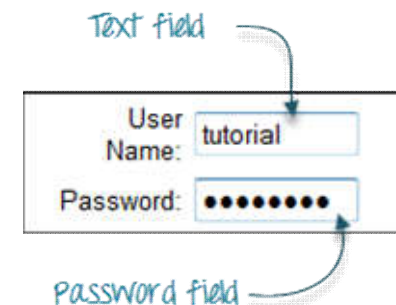
- InputBox (Text field, password field)
- El método `sendKeys` se utiliza para introducir valores en el campo de un formulario

this will type the text "tutorial" onto the element with name="username"

```
driver.findElement(By.name("username")).sendKeys("tutorial");
```



User Name: tutorial



User Name: tutorial

Password: .....

Text field

password field

- Para limpiar información de campos `username`

```
driver.findElement(By.name("username")).clear();
```

- El elemento tiene que estar visible en el navegador para poder interactuar

## Interacción con formularios

- Radio Botón: Para seleccionar un radio botón se utiliza el método click

```
driver.findElement(By.cssSelector("input[value='Business']")).click();
```

Service Class:

☐ Economy class

☒ Business class

☐ First class

- Check Box: se hace de igual forma que antes con el método click

```
public static void main(String[] args) {  
    WebDriver driver = new FirefoxDriver();  
    String baseURL = "http://www.facebook.com";  
  
    driver.get(baseURL);  
    WebElement chkFBPersist = driver.findElement(By.id("persist_box"));  
    for(int i=0; i<2; i++){  
        chkFBPersist.click();  
        System.out.println(chkFBPersist.isSelected());  
    }  
    driver.quit();  
}
```

First click - checkbox  
was toggled on

Console Problems

<terminated> myclass [Java Appli

true

false

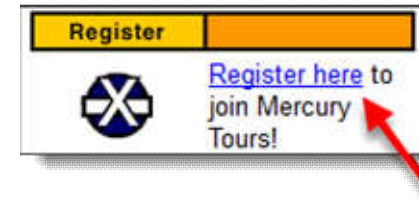
Second click - checkbox  
was toggled off

Email or Phone

☒ Keep me logged in

## Interacción con formularios

- **Enlaces:** se accede a ellos con el método click
- El acceso puede ser con el nombre completo del enlace o con un substring del mismo.



```
driver.findElement(By.linkText("Register here")).click();
```

```
driver.findElement(By.partialLinkText("here")).click();
```

- **DropBox:** necesita código adicional para contralar las interacciones de usuario

Country: UNITED STATES

```
<td>  
  <select size="1" name="country">  
</td>
```

```
import org.openqa.selenium.support.ui.Select;
```

```
Select drpCountry = new Select(driver.findElement(By.name("country")));
```

```
drpCountry.selectByVisibleText("ANTARCTICA");
```

## Interacción con formularios

- Para realizar selecciones múltiples en elementos que lo permitan podemos utilizar `SelectByVisibleText` o `SelectByIndex`

page source

```
<select id="fruits" multiple="">
  <option value="banana">Banana</option>
  <option value="apple">Apple</option>
  <option value="orange">Orange</option>
  <option value="grape">Grape</option>
</select>
```

HTML page



```
public static void main(String[] args) {
    WebDriver driver = new FirefoxDriver();

    driver.get("http://jsbin.com/osebed/2");
    Select fruits = new Select(driver.findElement(By.id("fruits")));
    fruits.selectByVisibleText("Banana");
    fruits.selectByIndex(1);
}
```



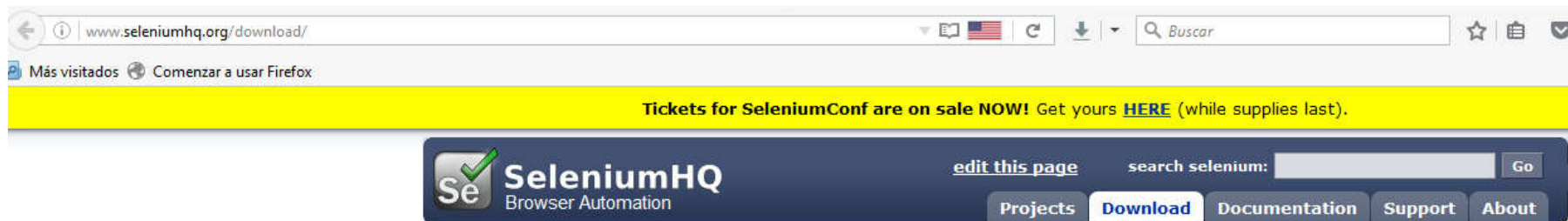
- Envío de un formulario: El método `submit` es una alternativa a click sobre el botón enviar

```
driver.findElement(By.name("userName")).sendKeys("tutorial");
driver.findElement(By.name("password")).sendKeys("tutorial");
driver.findElement(By.name("password")).submit();
```

*submit() was used on the password text box instead of on the sign-in button.*

## Utilización desde Java (Eclipse, NetBeans, IntelliJ IDEA)

- Para programar un cliente en Java debemos bajar las librerías de Selenium



### Selenium Client & WebDriver Language Bindings

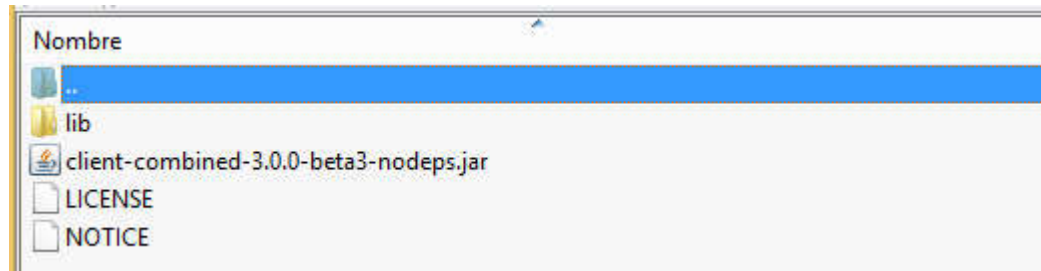
In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

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	3.6.0	2017-09-27	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">Javadoc</a>
C#	3.6.0	2017-09-25	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Ruby	3.6.0	2017-09-24	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Python	3.6.0	2017-09-25	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>
Javascript (Node)	3.6.0	2017-10-06	<a href="#">Download</a>	<a href="#">Change log</a>	<a href="#">API docs</a>

## Utilización desde Java

- Descomprimir el zip en un directorio del disco duro



- Descargaremos ahora los Webdrivers para los navegadores desde enlaces de la página de selenium

### The Internet Explorer Driver Server

This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver. Please make sure that this is available on your \$PATH (or %PATH% on Windows) in order for the IE Driver to work as expected.

Download version 2.53.1 for (recommended) [32 bit Windows IE](#) or [64 bit Windows IE](#)  
[CHANGELOG](#)

- Descomprima el zip sobre un directorio del disco duro

## Utilización desde Java

- Para otros navegadores descargue el zip

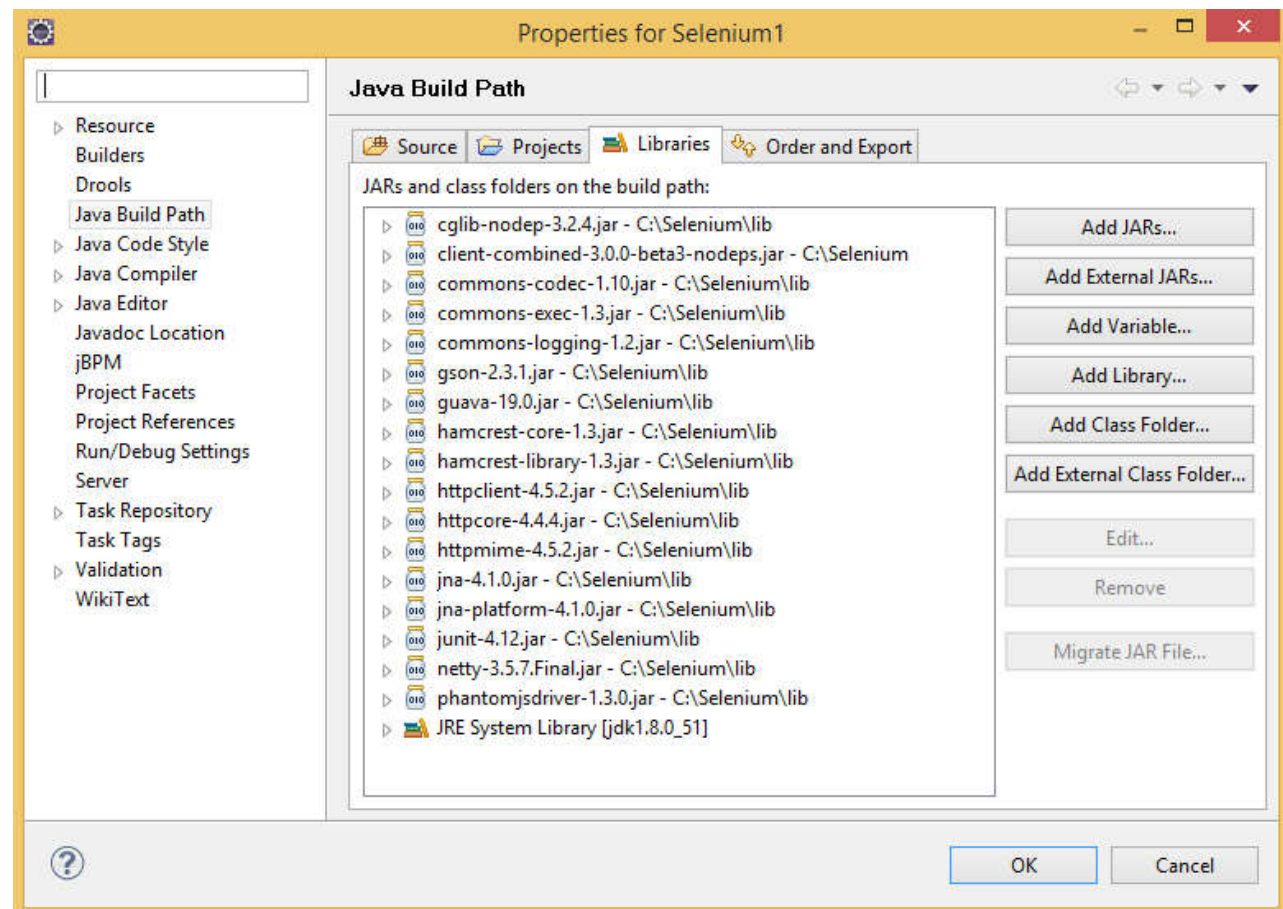
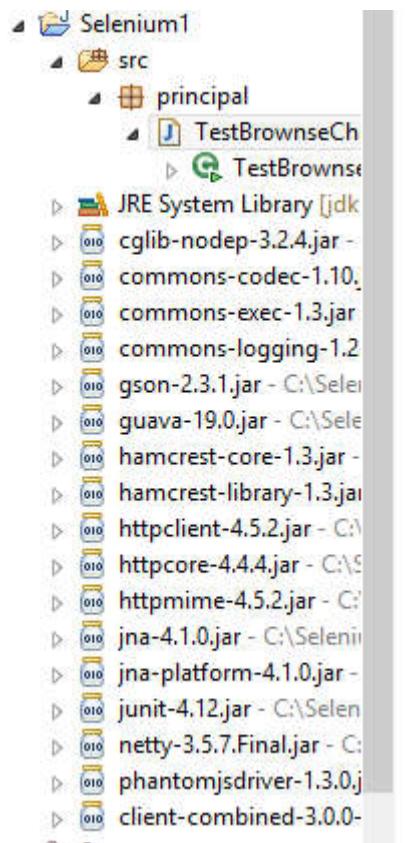
Third Party Browser Drivers **NOT DEVELOPED** by seleniumhq

Browser					
<a href="#">Mozilla GeckoDriver</a>	<a href="#">0.19.0</a>	<a href="#">change log</a>	<a href="#">issue tracker</a>	<a href="#">Implementation Status</a>	Released 2017-09-16
<a href="#">Google Chrome Driver</a>	<a href="#">2.33</a>	<a href="#">change log</a>	<a href="#">issue tracker</a>	<a href="#">selenium wiki page</a>	Released 2017-10-03
<a href="#">Opera</a>	<a href="#">2.29</a>		<a href="#">issue tracker</a>	<a href="#">selenium wiki page</a>	Released 2017-06-27
<a href="#">Microsoft Edge Driver</a>			<a href="#">issue tracker</a>	<a href="#">Implementation Status</a>	
<a href="#">GhostDriver</a>	(PhantomJS)		<a href="#">issue tracker</a>	<a href="#">SeConf talk</a>	
<a href="#">HtmlUnitDriver</a>	<a href="#">2.27</a>		<a href="#">issue tracker</a>		Released 2017-06-04
<a href="#">SafariDriver</a>			<a href="#">issue tracker</a>		
<a href="#">Windows Phone</a>			<a href="#">issue tracker</a>		
<a href="#">Windows Phone</a>	<a href="#">4.14.028.10</a>		<a href="#">issue tracker</a>		Released 2013-11-23
<a href="#">SeleniumAndroid</a> - Selenium for Android			<a href="#">issue tracker</a>		
<a href="#">ios-driver</a>			<a href="#">issue tracker</a>		
<a href="#">BlackBerry 10</a>			<a href="#">issue tracker</a>		Released 2014-01-28
<a href="#">Appium</a>			<a href="#">issue tracker</a>		
<a href="#">CrossWalk</a>			<a href="#">issue tracker</a>		Released 2014-05-05
<a href="#">QtWebDriver</a>	<a href="#">1.3.1</a>	<a href="#">change log</a>	<a href="#">issue tracker</a>	<a href="#">wiki page</a>	Released 2015-06-17
<a href="#">jBrowserDriver</a>			<a href="#">issue tracker</a>		
<a href="#">Winium.Desktop</a>	<a href="#">latest</a>	<a href="#">change log</a>	<a href="#">issue tracker</a>	<a href="#">wiki, talks &amp; demos</a>	
<a href="#">Winium.StoreApps</a>	<a href="#">latest</a>	<a href="#">change log</a>	<a href="#">issue tracker</a>	<a href="#">wiki, talks &amp; demos</a>	
<a href="#">Winium.StoreApps.CodedUi</a> (Early stage WIP)	<a href="#">latest</a>		<a href="#">issue tracker</a>	<a href="#">talks &amp; demos</a>	



## Utilización desde Java

- Desde cualquier entorno de desarrollo Java se crea un nuevo proyecto Java y se añaden los jar de Selium.
- En Eclipse



## Utilización desde Java

- En el main() del proyecto llamamos al WebDriver del navegador correspondiente.

```
public static void Chrome()  
{ String exePath = "C:\\Selenium\\chromedriver_win32\\chromedriver.exe";  
System.setProperty("webdriver.chrome.driver", exePath);  
ChromeOptions options = new ChromeOptions();  
options.addArguments("--start-maximized");  
WebDriver driver = new ChromeDriver(options);  
driver.get("http://www.pccomponentes.com");  
// No funciona en Chrome driver.manage().window().maximize();  
driver.quit();  
}
```

```
public static void InternetExplorer()  
{ String driverPath = "C:\\Selenium\\IEDriverServer_x64_2.53.1\\";  
System.setProperty("webdriver.ie.driver", driverPath+"IEDriverServer.exe");  
DesiredCapabilities capabilities = DesiredCapabilities.internetExplorer();  
capabilities.setCapability(InternetExplorerDriver.INTRODUCE_FLAKINESS_BY_IGNORING_SECURITY_DOMAINS, true);  
WebDriver driver = new InternetExplorerDriver();  
driver.manage().window().maximize();  
//driver.navigate().to("http://www.pccomponentes.com");  
driver.get("http://www.pccomponentes.com");  
driver.quit();  
}
```

## Utilización desde Java

- Por último para Firefox

```
public static void Firefox()  
{  
    String exePath = "C:\\\\Selenium\\geckodriver-v0.10.0-win64\\geckodriver.exe";  
    System.setProperty("webdriver.gecko.driver", exePath);  
    DesiredCapabilities capabilities=DesiredCapabilities.firefox();  
    capabilities.setCapability("marionette", true);  
    WebDriver driver = new FirefoxDriver(capabilities);  
    driver.get("http://www.pccomponentes.com");  
    driver.quit();  
}
```

- Desde main llamamos al driver que necesitamos

```
public static void main(String[] args) {  
    Chrome();  
    Firefox();  
    InternetExplorer();  
}
```

## Entrega 2

- En Poliformat se describe la entrega 2 de Selenium
- El ejemplo guiado no tiene nada que ver con el caso de estudio

## Ejemplo guiado

- Nos conectaremos con pccomponentes y obtendremos los teléfonos de la marca LG.
  1. Conectarnos con el sitio web y encontrar los localizadores para interactuar con la página, utilizaremos para ello Firebug
  2. Introducir la información desde el navegador para encontrar también los localizadores de respuesta, es decir los datos que devuelve la página frente a una consulta.
  3. Crear el código Java que permita la interacción y la recogida de resultados.

La página web de PCComponentes ha variado ligeramente desde que se hizo el código que viene a continuación, antes de pulsar ver más (móviles) se tiene que pulsar sobre un nuevo botón de etiqueta MARCAS. La posición de LG también ha cambiado y casualmente es la primera de la lista, el localizador es el mismo.

## Ejemplo guiado

- Una vez en la página



- Activamos Firebug y pulsamos sobre el botón de búsqueda (inspeccionar elemento)



# Ejemplo guiado

- La ventana de Firebug muestra el elemento seleccionado

The screenshot shows a web browser with the URL `https://www.pccomponentes.com/?gclid=Cj0KEQjw4fy_BRcX7b6rq_WZgl0BEiQAI78ndzPvytTerahcCoW4GdcVVVZhYo7yfdKt`. The Firebug developer tool is open, displaying the DOM tree on the left and the CSS pane on the right. The DOM tree shows the following structure:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <body id="home" class="home lang-es">
      <noscript><iframe src="//www.googletagmanager.com/ns.html?id=GTM-MCDXK2" height="0" width="0" style="display:none;visibility:hidden"></iframe></noscript>
      <script>
      <header id="cabecera-principal" class="container-fluid cabecera cabecera-principal p-y-1">
        <div class="row">
          <a href="https://www.pccomponentes.com/">
          <div id="trigger-menu" data-pcc-trigger="toggle-menu-principal" data-pcc-event="click" data-pcc-group="menus-desplegables">
            <div class="contenedor-buscador-principal">
              <form id="buscador" class="input-group input-group-lg" method="GET" action="https://www.pccomponentes.com/buscar/">
                <input id="query" class="form-control form-control" name="query" required="required" placeholder="Busca en PcComponentes..." data-url="https://www.pccomponentes.com/buscar/ajax/mini" autocomplete="off" type="text">
                <a class="buscador-principal_cierre activador-buscador">Cancelar</a>
              </form>
              <div class="buscador__superposicion-fondo"></div>
              <div id="pccom-suggestion-box" class="pccom-suggestion-box"></div>
            </div>
            <div class="posicionador-activador-buscador hidden-md-up">
            <ul id="usuario" class="pull-xs-right p-r-2 p-l-0 user-floating">
            </div>
          </div>
        </div>
      </script>
    </body>
  </html>
```

The CSS pane on the right shows the styles for the selected element, including:

- `bootstrap.css (línea 1)`: `*, ::before, ::after { box-sizing: inherit; }`
- `estilo-...pal.css (línea 1)`: `.contenedor-buscador-principal #buscador input { background-color: #ebeced; border: 2px solid #e1e1e1; border-radius: 23px; height: 44px; width: 100%; }`
- `estilo-...pal.css (línea 1)`: `.contenedor-buscador-principal #buscador input { background-color: transparent; background-image: ... }`

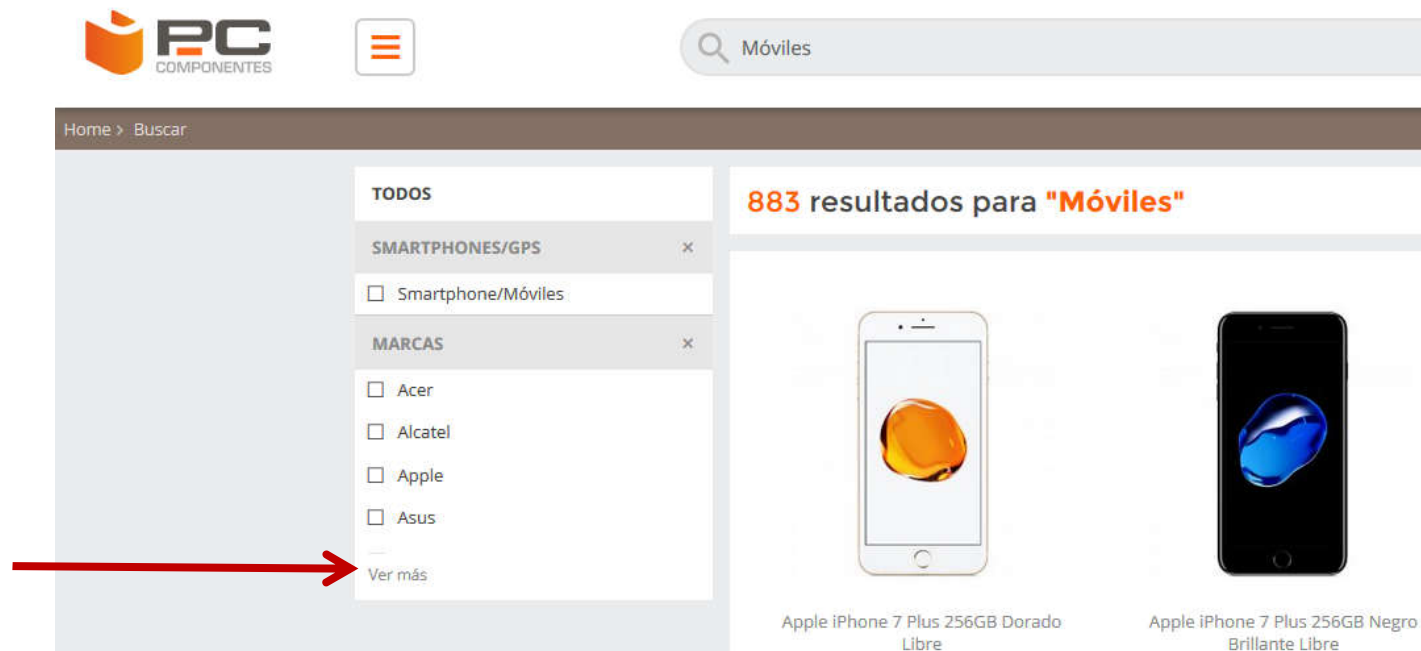
Below the browser window, the website's header is visible, featuring the PC COMPONENTES logo, a search bar, and links for "Entra o regístrate" and "Mi Carrito". A promotional banner for ASUS is also present, stating: "¡Ahora al comprar DISPOSITIVOS DE RED ASUS te llevas HASTA 50€\* DE REGALO para juegos!".

## Ejemplo guiado

- El elemento seleccionado tiene las siguientes propiedades

```
<form id="buscador" class="input-group input-group-lg" method="GET" action="https://www.pccomponentes.com/buscar/">  
  <input id="query" class="form-control form-control" name="query" required="required" placeholder="Busca en  
  PcComponentes..." data-url="https://www.pccomponentes.com/buscar/ajax/mini" autocomplete="off" type="text">  
  <a class="buscador-principal_cierre activador-buscador">Cancelar</a>  
</form>
```

- Tecleamos *Móviles* en el campo de entrada de texto y pulsamos <Enter>





## Ejemplo guiado

- Seleccionamos con Firebug inspeccionar elemento sobre la etiqueta *Ver más*
- El menú contextual nos muestra los localizadores

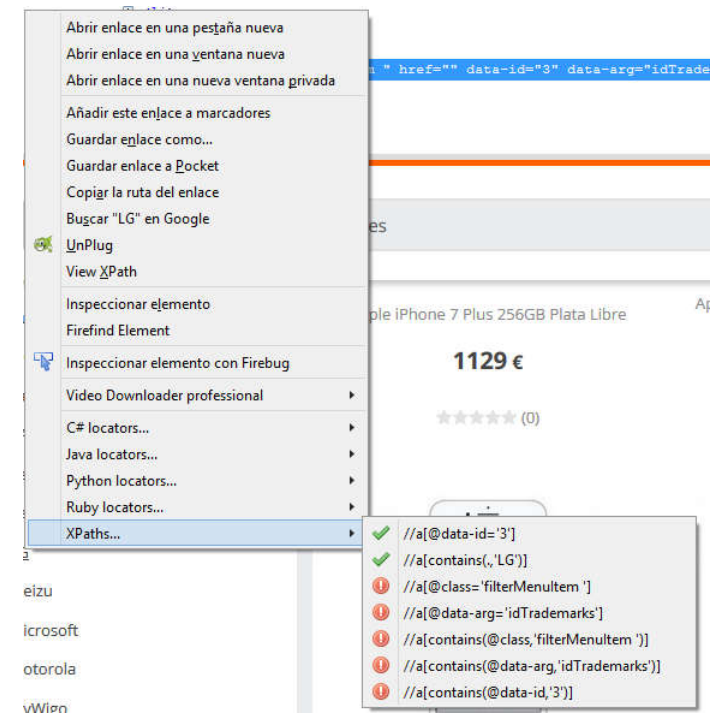
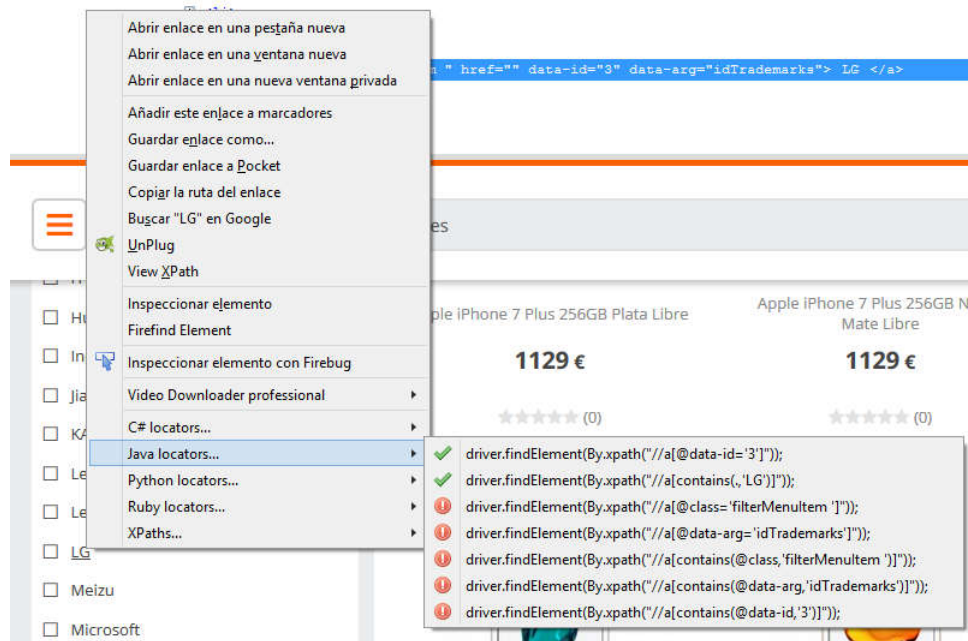
The screenshot illustrates the process of inspecting a web element using Firebug. The HTML panel shows the DOM structure, with the selected element being a link with the text "Ver más". The Firebug context menu is open, displaying various actions. The "XPaths..." option is selected, showing a list of XPath expressions that can be used to locate the element:

- ✓ `//span[@class='mas']`
- ✓ `//span[contains(., 'más')]`
- ✓ `//span[contains(@class, 'mas')]`

The background shows a mobile phone advertisement on a website, with the text "3 resultados para 'Móviles'" visible.

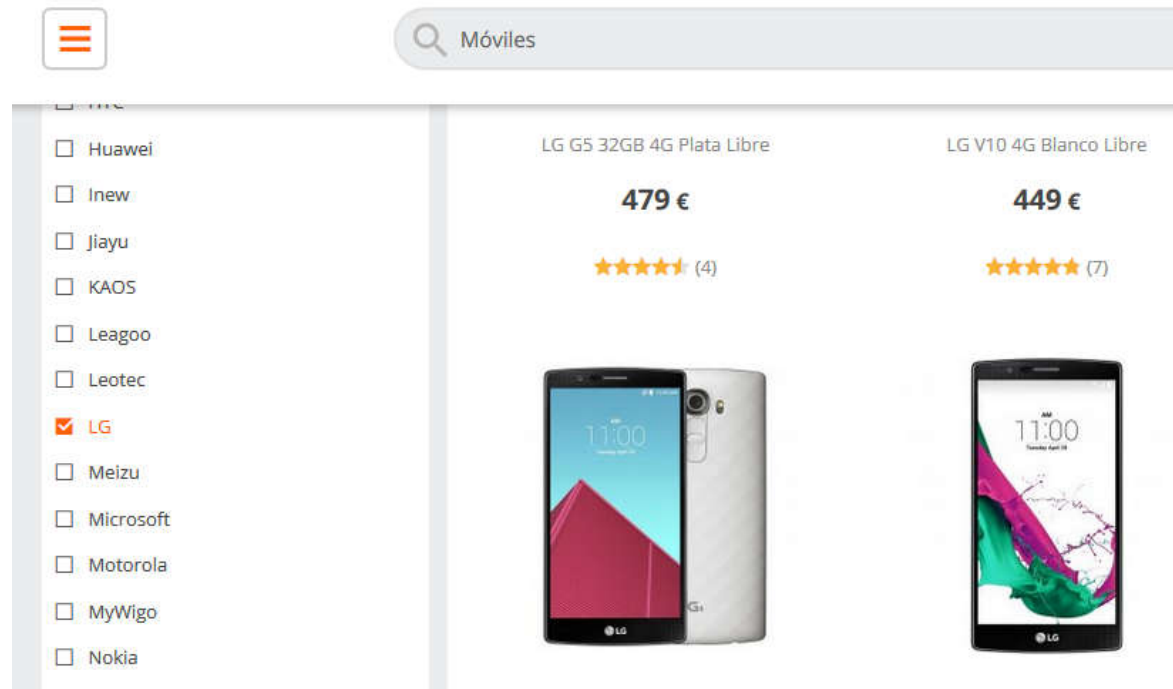
## Ejemplo guiado

- Pulsamos sobre *Ver más* para obtener la lista de todas las marcas de teléfono.
- Hacemos *scroll* en la página hasta localizar el radio botón de LG
- Al igual que antes utilizamos inspeccionar elemento con Firebug



## Ejemplo guiado

- Marcamos LG desde el navegador



- Como necesitamos procesar la respuesta de la página desde código Java tendremos que obtener los localizadores de la página para esa respuesta.

## Ejemplo guiado

- La respuesta de la página utiliza la librería de twitter bootstrap.css

The screenshot shows a web browser window with the URL `https://www.pccomponentes.com/buscar?query=Móviles#p0-m3`. The browser's developer tools are open, displaying the HTML structure of the page. The `div.white-card` element is selected, showing a grid of `col-xs-6 col-sm-4 col-md-4 col-lg-4` classes, which are Bootstrap grid classes. The CSS panel on the right shows the `bootstrap.css` file loaded, with the `.white-card` class defined in `estilo...pa.css`. The website's header includes the PC Componentes logo, a search bar with the text "Móviles", and links for "Entra o regístrate" and "Mi Carrito". The main content area displays "882 resultados para 'Móviles'" and a list of smartphones, including the LG G5 32GB 4G Titan Libre Reacondicionado for 485,55 € and two LG G5 32GB 4G Titan Libre models for 479 € each. A cookie notice is visible at the bottom of the page.

## Ejemplo guiado

- Las lista de teléfonos está en la siguiente posición de la página

The screenshot shows a web browser window with the address bar displaying `componentes.com/buscar/?query=Móviles#p0-m3`. The browser's developer tools are open, showing the HTML source code. A red box highlights a list of 20 `<div class="col-xs-6 col-sm-4 col-md-4 col-lg-4">` elements. A red arrow points from this box to the rendered page below. The rendered page shows a search bar with the text "Móviles" and a search button. Below the search bar, there is a section titled "882 resultados para 'Móviles'" and a list of mobile phones, including a blue phone, a black phone, and a white phone.

## Ejemplo guiado

- Si en la ventana de Firebug expandimos el elemento seleccionado

```
<div class="col-xs-6 col-sm-4 col-md-4 col-lg-4">
  <article class="tarjeta-articulo expandible" itemscope="" itemtype="http://schema.org/Product">
    <meta itemprop="sku" content="101216">
    <meta itemprop="brand" content="LG">
    <div class="tarjeta-articulo__elementos-basicos">
      <div class="tarjeta-articulo__foto">
      <header class="tarjeta-articulo__nombre">
        <h3 itemprop="name">
          <a class="GTM-productClick enlace-disimulado" itemprop="url" href="/lg-g5-32gb-4g-titan-libre-reacondicionado" data-name=
            Reacondicionado" data-id="101216" data-price="485.55" data-brand="LG" data-category="Smartphone/Móviles" data-list=
            Reacondicionado</a>
        </h3>
      </header>
      <div class="tarjeta-articulo__precios" itemprop="offers" itemscope="" itemtype="http://schema.org/Offer">
      <div class="tarjeta-articulo__extras">
        <a class="GTM-productClick enlace-superpuesto" itemprop="url" href="/lg-g5-32gb-4g-titan-libre-reacondicionado" data-name=
          Reacondicionado" data-id="101216" data-price="485.55" data-brand="LG" data-category="Smartphone/Móviles" data-list="" data
        </div>
      <div class="tarjeta-articulo__elementos-adicionales">
        <div class="tarjeta-articulo__disponibilidad article.availability_class"> Sin fecha de entrada </div>
      </div>
    </div>
  </article>
</div>
```

data-name="LG G5 32GB 4G Titan Libre  
Reacondicionado" data-id="101216" data-price="485.55"

- La misma información está en varios sitios

## Ejemplo guiado

- Para cada uno de los elementos tenemos que extraer la información del artículo a partir del nodo `<article class =“tarjeta-articulo expandible”`
- Web Driver no funciona con nombres de clases con espacios en blanco

```
Driver.findElement(By.className("tarjeta-articulo expandible"))
```

- Debe utilizarse un selector CSS o bien Xpath.

```
ArrayList<WebElement> resultados2=  
(ArrayList<WebElement>) driver.findElements(  
By.xpath("//*[contains(@class, 'tarjeta-articulo expandible')]"));
```

## Ejemplo guiado: código cliente Java

- Para el navegador Chrome

```
public static void Chrome()
{
    String exePath = "C:\\Selenium\\chromedriver_win32\\chromedriver.exe";
    System.setProperty("webdriver.chrome.driver", exePath);
    ChromeOptions options = new ChromeOptions();
    options.addArguments("--start-maximized");
    WebDriver driver = new ChromeDriver(options);
    driver.get("http://www.pccomponentes.com");
    // Paso 1 introducir la cadena de búsqueda
    String searchText="Móviles"+ '\n';
    WebElement searchInputBox=driver.findElement(By.name("query"));
    searchInputBox.sendKeys(searchText);

    // Paso 2 esperar a los resultados de búsqueda
    WebDriverWait waiting = new WebDriverWait(driver, 10);
    waiting.until( ExpectedConditions.presenceOfElementLocated(
        By.id("resultados-busqueda") ) );
}
```



## Ejemplo guiado: código cliente Java

- Esperar que se muestre "ver más" y pulsar click

// Paso 3 buscar el elemento ver más

```
WebElement elementoMas = driver.findElement(By.xpath("//span[@class='mas']"));
elementoMas.click();
```

- Cerrar la ventana de cookies y esperar a que se muestre el radio botón de LG

// Paso 4 Cerrar la ventana de cookies

```
driver.findElement(By.cssSelector(".btn.btn-block.btn-secondary.m-t-1.accept-
cookie")).click();
```

- Esperar a que se muestre el radio botón LG y hacer scroll

// Paso 5 esperar a que salga el radio botón de LG y hacer scroll

```
waiting = new WebDriverWait(driver, 10);
waiting.until( ExpectedConditions.presenceOfElementLocated( By.xpath("//a[@data-
id='3']")) );
WebElement element = driver.findElement(By.xpath("//a[@data-id='3']"));
Actions actions = new Actions(driver);
actions.moveToElement(element);
actions.perform();
JavascriptExecutor jse = (JavascriptExecutor)driver;
jse.executeScript("window.scrollTo(0,100)", "");
```

## Ejemplo guiado: código cliente Java

- Para que se simule la pulsación el elemento tiene que estar visible

```
// Paso 6 pulsar sobre el radio botón de los teléfonos LG  
element.click();
```

- Esperar a que se muestre la lista de teléfonos LG

```
// Paso 7 esperar a que muestre los telefonos LG  
waiting = new WebDriverWait(driver, 10);  
waiting.until( ExpectedConditions.presenceOfElementLocated( By.xpath("//a[@data-  
brand='LG']") ));
```

- Obtener la ficha de todos los productos de la página

```
// Paso 8 Obtener todos los elementos que aparecen en la primera página  
ArrayList<WebElement> resultados2= (ArrayList<WebElement>)  
driver.findElements(  
By.xpath("//*[contains(@class, 'tarjeta-articulo expandible')]"));  
System.out.println("Resultados " + resultados2.size());
```

## Ejemplo guiado: código cliente Java

- Obtener la información de cada artículo, para ello iteramos sobre la lista de `WebElement`

```
// Paso 9 Iterar sobre la lista para obtener las características de los artículos
WebElement actual_Elemento, navegacion2;
for (int i=0; i< resultados2.size(); i++)
{
    actual_Elemento = resultados2.get(i); // elemento actual de la lista.
    System.out.println("Elemento: " + i);
    navegacion2 =actual_Elemento.findElement(By.xpath("./descendant::a"));
    System.out.println("Por navegación2: " + navegacion2.getAttribute("data-name").toString());
    System.out.println("Por navegación2: " + navegacion2.getAttribute("data-price").toString() );
    System.out.println("Qué nodo :" +navegacion2.toString());
    // de forma equivalente a xpath pero con css
    navegacion2 = actual_Elemento.findElement(By.cssSelector("a[class='GTM-productClick enlace-
superpuesto']"));
    System.out.println("Qué nodo :" +navegacion2.toString());
    System.out.println("Por navegación2: " + navegacion2.getAttribute("data-name").toString());
    System.out.println("Por navegación2: " + navegacion2.getAttribute("data-price").toString() );
    // si está disponible o no, se buscar en tarjeta-articulo__elementos-adicionales
    navegacion2 = actual_Elemento.findElement(By.className("tarjeta-articulo__elementos-adicionales"));
    System.out.println("Por navegación 2 " + navegacion2.getText()); // el texto indica si está
    disponible o no
    System.out.println("-----");
}
```

- El código está repetido para xpath y para css.

## Ejemplo guiado: código Java cliente

- Una traza de ejecución

```
Starting ChromeDriver 2.24.417431 (9aea000394714d2fbb20850021f6204f2256b9cf) on port 11751
Only local connections are allowed.
oct 15, 2016 11:25:15 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFORMACIÓN: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end
oct 15, 2016 11:25:16 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFORMACIÓN: Detected dialect: OSS
Encontrados elementos en la página: 24
Resultados en la nueva búsqueda 24
#####
Elemento: 0
Por navegación2: LG G5 32GB 4G Titan Libre Reacondicionado
Por navegación2: 485.55
Qué nodo :[[[ChromeDriver: chrome on WIN8_1 (3ce51cbf3df2bfa65ca4e0f27e3d08bd)] -> xpath: //*[contains(@class,
Qué nodo :[[[ChromeDriver: chrome on WIN8_1 (3ce51cbf3df2bfa65ca4e0f27e3d08bd)] -> xpath: //*[contains(@class,
Por navegación2: LG G5 32GB 4G Titan Libre Reacondicionado
Por navegación2: 485.55
Por navegación 2 Sin fecha de entrada
-----
Elemento: 1
Por navegación2: LG G5 32GB 4G Titan Libre
Por navegación2: 479
Qué nodo :[[[ChromeDriver: chrome on WIN8_1 (3ce51cbf3df2bfa65ca4e0f27e3d08bd)] -> xpath: //*[contains(@class,
Qué nodo :[[[ChromeDriver: chrome on WIN8_1 (3ce51cbf3df2bfa65ca4e0f27e3d08bd)] -> xpath: //*[contains(@class,
Por navegación2: LG G5 32GB 4G Titan Libre
Por navegación2: 479
Por navegación 2 Sin fecha de entrada
-----
```

## Ejemplo guiado: código Java cliente

- Observaciones

- En algunas búsquedas, por ejemplo en la del caso guiado, podemos pulsar al final de la página en “ver más”, el botón desaparece cuando no haya más elementos que mostrar en la búsqueda actual.
- Para recoger todos los elementos debe modificar el código de ejemplo, recogiendo la lista de la página y después pulsando “ver más” para obtener la siguiente colección.

- Referencias adicionales

- Xpath [https://msdn.microsoft.com/en-us/library/ms256115\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms256115(v=vs.110).aspx)
- Xpath <http://www.guru99.com/xpath-selenium.html>
- Xpatn <http://santoshsarmajv.blogspot.com.es/2012/07/how-to-write-xpath.html>