# Taller de SELENIUM

Vamos a cacharrear un rato

documentados
consultora Drupal

# ¿Quienes somos?

Documentados es un pequeño proyecto que lleva más de 5 años trabajando de forma seria y profesional con Drupal.

El núcleo duro de Documentados:

@nachenko

@oskarcalvo

llo - asocial.

# ¿Qué es Selenium?

Selenium es un framework que nos permite crear pruebas funcionales en diferentes lenguajes de programación, incluyendo php que es el que nos interesa a nosotros.

# Antes de correr, tipos de pruebas ;)

Prueba unitaria:

En programación, una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado....La idea es escribir casos de prueba para cada función no trivial o método en el módulo de forma que cada caso sea independiente del resto.

fuente: http://es.wikipedia.org/wiki/Prueba_unitaria

# Tipos de pruebas

Prueba funcional:

Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. La pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático.

Fuente: http://es.wikipedia.org/wiki/Pruebas_funcionales

# Tipos de pruebas

Aunque parezcan similares la verdad es que no lo son.

Las pruebas unitarias se basan en el código, y que este funcione correctamente, aunque no sea lo definido en las especificaciones del proyecto.

Las pruebas funcionales no evaluan el código, sino el resultado final, esto es si se muestra la opción de menú correcta, el enlace correcto, etc...

# ¿Pero qué nos ofrece Selenium?

Selenium se puede dividir en cuatro componentes:

- Selenium IDE.
- Selenion server.
- Selenium Client Driver.
- Selenium Grid.

# Selenium IDE

El IDE de selenium es una serie de addons de Firefox que hay que instalar para poder utilizarlo.

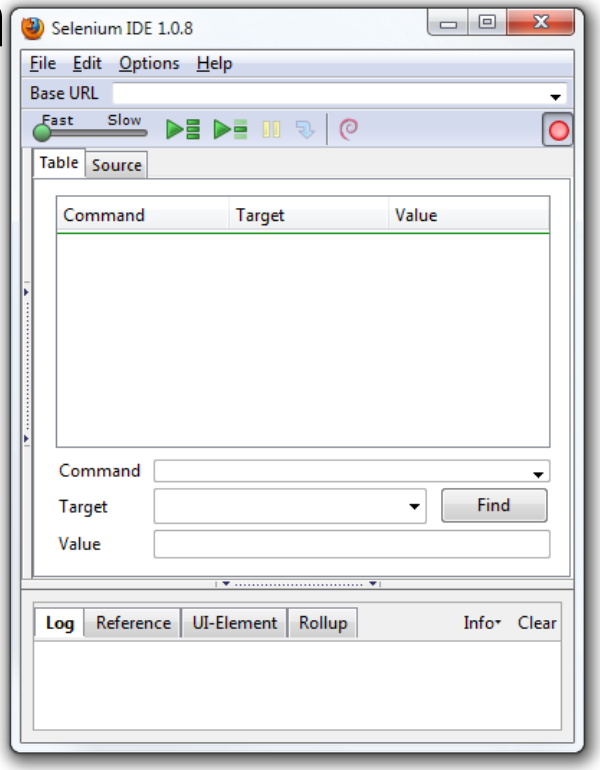
https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/

Además del IDE hay que instalar el addons que permite exportar a php los test de selenium.

https://addons.mozilla.org/en-US/firefox/addon/selenium-ide-php-formatters/?src=search

# Selenium IDE - que nos ofrece

La documentación de Selenium nos expica todas las opciones que su IDE nos ofrece:




http://seleniumhq.org/docs/02_selenium_ide.html

# Test Suits vs Test Case

El IDE de Selenium nos permite crear dos elementos.

- Test Case
- Test Suits

La documentació ofical del IDE la podemos encontrar en la web de Selenium:
Documentación oficial
Por defecto Selenium lo guarda todo en HTML.

# Test Suit

Un **Test Suit es un conjunto de Test Case** que se pueden ejecutar de forma conjunta.

```
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <title>Sample Selenium Test Suite</title>
</head>
<body>
   <table cellpadding="1" cellspacing="1" border="1">
      <thead>
         <tr><td>Test Cases for De Anza A-Z Directory Links</td></tr>
      </thead>
   <tbody>
      <tr><td><a href="./a.html">A Links</a></td></tr>
      <tr><td><a href="./b.html">B Links</a></td></tr>
      <tr><td><a href="./c.html">C Links</a></td></tr>
      <tr><td><a href="./d.html">D Links</a></td></tr>
   </tbody>
   </table>
</body>
</html>
```

# Test Suit

Tener uno o varios Test Suits es la mejor escusa para irse a por un "Piti" o "Coffe".

# Test Case

Los Test Case son las pruebas propiamente dichas.

Podemos hacer pruebas desde lo más básicas, detectar un elemento a una web, hasta pruebas más complejas como:

- Un proceso de registro.
- Creación de contenido.
- Etc...

# Selenium IDE

Vamos a realizar nuestro primer test case.

Abrimos la url que queremos testear, y ejecutamos el IDE.

Cada acción que se haga se guarda en el IDE como una acción/tarea.

Además, en el botón derecho del ratón tenemos opciones para agregar al test.

# Selenium IDE

Vamos a crear nuestro primer test. :)

Abrimos el navegador en google.
Activamos el IDE de Selenium
Escribimos en la caja de búsqueda "drupal".

# Un poco más de profundidad

Para entender mejor los códigos recomiendo la lectura de este documento de referencias de Selenium:

http://release.seleniumhq.org/selenium-core/0.8.0/reference.html

# Hay que instalar Phpunit

Instalamos phpunit

```
pear channel-discover pear.phpunit.de
```

```
pear install phpunit/PHPUnit
```

# Guardar los test de Selenium

Para guardar los test, la forma básica es en html, pero también podemos guardarlos en formato phpunit.

Ojo para ello hay que agregar en el archivo al principio*:

```
require_once 'PHPUnit/Extensions/SeleniumTestCase.
php';
```

Y tendremos nuestras pruebas para ser ejecutadas desde phpunit.

*http://www.phpunit.de/manual/3.0/en/selenium.html

# Selenium server

El servidor de Selenium es una app en java que permite levantar un demonio.

Por si mismo el servidor no hace nada, pero es la puerta para que podamos ejecutar nuestros test desde phpunit, o desde html.

Lo descargamos de la página oficial:

http://selenium.googlecode.com/files/selenium-server-standalone-2.21.0.jar

# Ejecutar un test desde phpunit

Levantamos el servidor:

```
java -jar selenium-server-standalone-2.18.0.jar
```

Ejecutamos la prueba que queremos valorar

```
phpunit test.php
```

# Ejecutar un test guardado en html

-htmlSuite requires you to specify:

* browserString (e.g. "*firefox")
* startURL (e.g. "http://www.google.com")
* suiteFile (e.g. "c:\absolute\path\to\my\HTMLSuite.html")
* resultFile (e.g. "c:\absolute\path\to\my\results.html")

java -jar selenium-server-standalone-2.0b3.jar -port 1234 -htmlSuite "*firefox" "http://localhost:8080/" "path to the suite" "path to the results"

# Selenium RC

Parece que es habitual que el servicio se quede "pinzado", y no deje continuar con nuestras pruebas, aparte de reiniciar la máquina podemos hace lo siguiente:
http://www.debian-administration.org/users/fugit/weblog/6
**selenium as a service**(no verificado)

Reiniciar selenium http://kirankanumuri.blogspot.com.es/2010/09/how-to-shut-down-selenium-server-if-it.html

# Selenium Client Driver

Client Driver es el siguiente paso, y es la evolución incorporada en Selenium 2.0.

Mientras que Selenium RC emulaba los procesos mediantes javascript, CD habla de forma nativa con los navegadores.

# Selenium Client Driver

La ventaja de php es que tenemos tres librerías para conectar con el CD de Selenium, y una de ellas es de Facebook.

- PHP by Chibimagic (real name unknown?)
- PHP by Lukasz Kolczynski
- PHP by facebook

# Un ejemplo de CD

```
require_once "phpwebdriver/WebDriver.php";
require("phpwebdriver/LocatorStrategy.php");

$webdriver = new WebDriver("localhost", "4444");
$webdriver->connect("firefox");
$webdriver->get("http://google.com");
$element = $webdriver->findElementBy(LocatorStrategy::
name, "q");
$element->sendKeys(array("selenium google code" ) );
$element->submit();

$webdriver->close();

//http://code.google.com/p/php-webdriver-bindings/
```

# Agradecimientos

Agradecimiento especial sobre todo a https://twitter.com/#!/albert1t0 por la presentación que hizo para la Drupalcamp Lima.

Y a Carles Climent por el asalto / derribo que hizo en twitter para que preparase la charla.

# RESPECT