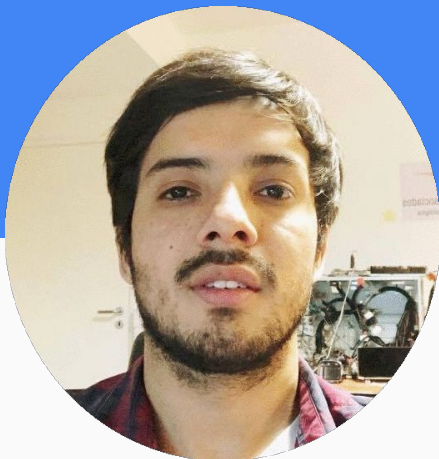




Automated Testing

¿Quienes somos?



Esteban Ocampo

QA Engineer Automation



Gaston Galante

QC Analyst

Contenido



Introducción

XPATH

Selenium IDE

Maven & Selenium

Xpaths & Selenium

Puliendo Xpath

Puliendo Selenium

Suites

Examen Final

CRONOGRAMA

Introducción
A la automatización

Selenium IDE

Xpath +
Selenium

Puliendo
Selenium

Examen
Final



Xpath

Maven & Selenium

Puliendo
Xpath

Suites

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark blue and orange, with scattered clouds. The city lights are visible, and the Empire State Building stands out prominently in the center. The text '¿Qué es el Testing?' is overlaid in a large, white, sans-serif font.

¿Qué es el Testing?

¿Qué es el Testing?

- Es un proceso de evaluación de un sistema o ciertos componentes, con la intención de chequear si se satisfacen los requerimientos.
- A su vez, el testing se realiza con la intención de buscar issues o bugs, algún requerimiento olvidado.
- Por lo general, se aconseja empezar a realizar testing en los primeros ciclos de vida de un software

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark blue and orange, with scattered clouds. The city is densely packed with skyscrapers, many of which are illuminated with their lights. The Empire State Building is prominent in the center, with its top lit in red and green. The Hudson River is visible on the left, and the East River is on the right. The title 'Tipos de pruebas' is overlaid in the center in a large, white, sans-serif font.

Tipos de pruebas

Tipos de pruebas

- Manual Testing
- Automated Testing

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark blue and orange, with scattered clouds. The city is densely packed with skyscrapers, many of which are illuminated with their lights. The Empire State Building is prominent in the center, with its top lit in red and green. The Hudson River is visible on the right side of the image. The text "Automated Testing" is overlaid in the center in a large, white, sans-serif font.

Automated Testing

Automated Testing

The background image shows a laptop screen displaying a dashboard. At the top, there is a line chart with a blue line showing an upward trend. Below the line chart, there is a pie chart with a green slice and a blue slice. The text 'New Visitor' and 'Returning Visitor' is visible next to the pie chart. The laptop keyboard is visible at the bottom of the frame.

- Hace referencia a ejecutar una prueba sin intervención humana.
- En términos prácticos, puede consistir en una prueba de unidad, una prueba de UI usando cierta herramientas.

An aerial photograph of a city skyline at dusk or dawn. The sky is a mix of dark blue and orange, with some clouds. The city is densely packed with skyscrapers, many of which are lit up with lights. The text '¿Porque automatizar las pruebas?' is overlaid in the center in a large, white, sans-serif font.

¿Porque automatizar las pruebas?

¿Porque automatizar las pruebas?

- La razón principal es el tiempo, ya que con las pruebas automatizadas se puede reducir el tiempo de las pruebas.
- A su vez, el objetivo es minimizar el esfuerzo de testing como sea posible a través de scripts.

An aerial photograph of a city skyline at dusk or dawn. The sky is a mix of dark blue, purple, and orange. The city is densely packed with skyscrapers, many of which are illuminated with lights. The Empire State Building is prominent in the center, with its top lit in red and green. The text 'Ventajas de Automated Testing' is overlaid in a large, white, sans-serif font, centered horizontally and vertically.

Ventajas de Automated Testing

Ventajas

- **Fiabilidad:** Las pruebas ejecutan las mismas operaciones cada vez que se ejecutan, eliminando así el error humano.
- **Repetición:** Es ejecutado en forma determinista, por ende no hay variación de resultados.

Ventajas

- **Rapidez:** Las herramientas de testing automatizado corren las pruebas significativamente más rápido que los testers humanos.
- **Reducción de costos:** Los recursos necesarios para ejecutar regresiones son menores.

Ventajas

- **Reusabilidad:** Se pueden usar las pruebas en diferentes versiones de la aplicación.
- **Software de mejor calidad:** Se pueden ejecutar más pruebas en menos tiempo y con menos recursos.

An aerial photograph of a city skyline at dusk or dawn. The sky is a mix of dark blue, purple, and orange. The city is densely packed with skyscrapers, many of which have their lights on. The Empire State Building is prominent in the center. The text 'Desventajas de Automated Testing' is overlaid in a large, white, sans-serif font.

Desventajas de Automated Testing

Desventajas

- El esfuerzo inicial es mayor. La creación del script puede ser más costoso que la creación del caso de prueba para ejecución manual.
- Tiempo y esfuerzo para el mantenimiento.



LOCATOR S (XPATH)



LOCATORS (XPATH)

Se utilizan para hacer referencia a un elemento. Por ejemplo, podría ser para clicar a un botón

¿Qué es XPath?

XPath (XML Path Language) es un lenguaje que permite hacer referencia a nodos a partir de ciertas expresiones. La idea es parecida a las expresiones regulares. XPath permite buscar y seleccionar en base a la estructura y/o propiedades.

Nodos

Un nodo es un elemento, el cual por lo general se lo encuentra entre los símbolos “menor a” y “mayor a” (< y >). Y generalmente son llamados por el tipo de nodo, el cual esta dado por la primer palabra dentro del mismo (eg: <node_type attr1="val1" attr2="val2">). Es importante poder diferenciar los atributos únicos de un nodo para poder crear un buen xpath.

Ejemplo

<A1>

<B1>

<C1>

<D1></D1>

</C1>

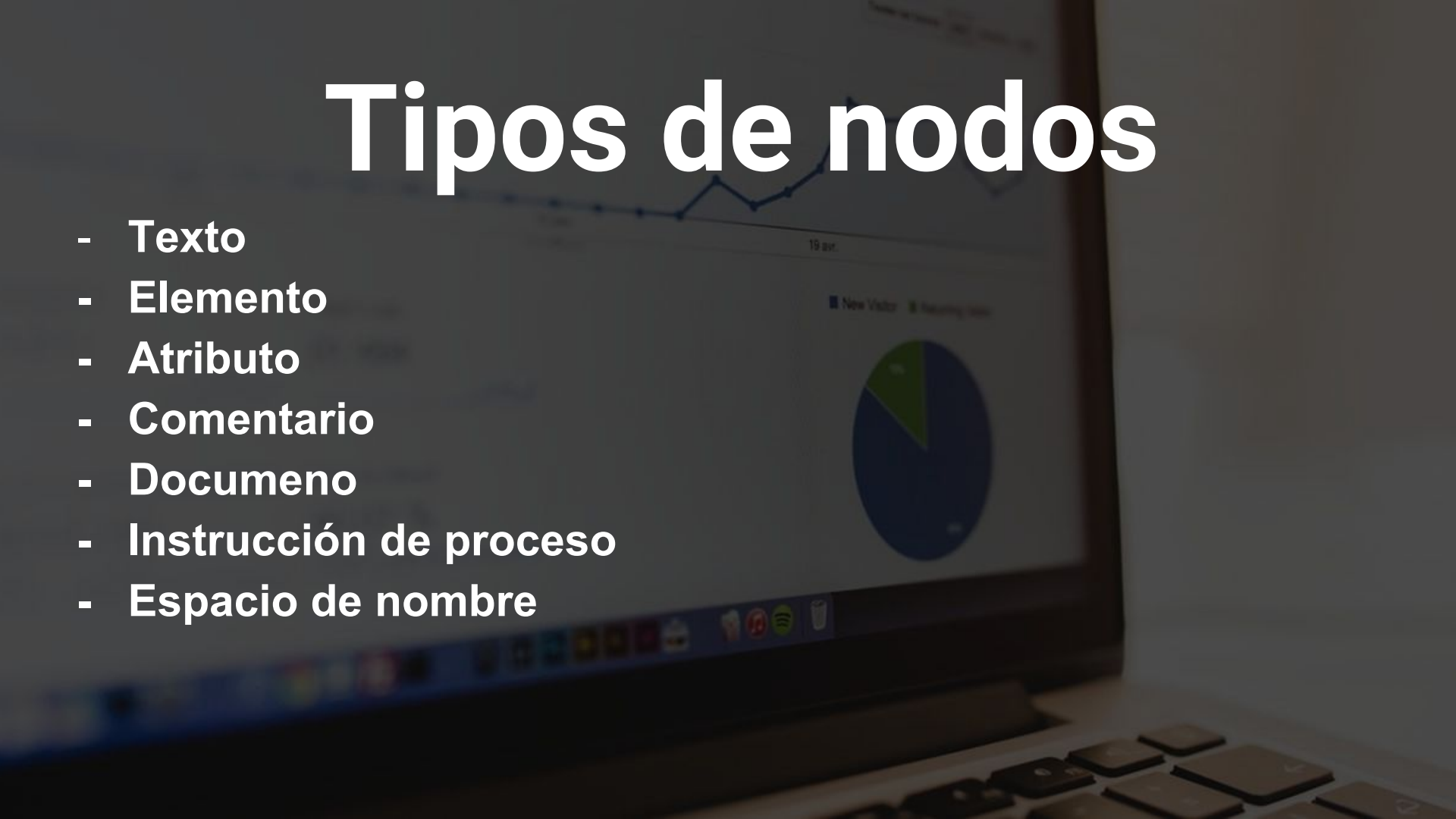
</B1>

</A1>

<A2></A2>

Tipos de nodos

- Texto
- Elemento
- Atributo
- Comentario
- Documeno
- Instrucción de proceso
- Espacio de nombre



Tipos Nodos



Propiedades

En html los nodos en general son llamados tags, y cada tag dependiendo de que tipo de tag sea puede tener distinto atributos, llamados propiedades. Por ejemplo un tag de `` puede tener un atributo “src” el cual especifica la ubicación de la imagen ``

http://www.way2tutorial.com/xml/xpath_node_test_examples.php

Relaciones entre Nodos

Padres

Cada elemento y atributo tiene un padre.

En el siguiente ejemplo; el elemento book es el padre de title, author, year, y price:

```
<book>
```

```
  <title>Harry Potter</title>
```

```
  <author>J K. Rowling</author>
```

```
  <year>2005</year>
```

```
  <price>29.99</price>
```

```
</book>
```

Relaciones entre Nodos

Hijos

Los elementos nodos pueden tener cero, uno o más hijos.

En el siguiente ejemplo; los elementos title, author, year, y price son todos hijos del elemento book:

```
<book>  
  <title>Harry Potter</title>  
  <author>J K. Rowling</author>  
  <year>2005</year>  
  <price>29.99</price>  
</book>
```


Relaciones entre Nodos

Hermanos

Son los nodos que tienen el mismo padre.

En el siguiente ejemplo; los elementos title, author, year, y price son todos hermanos:

```
<book>  
  <title>Harry Potter</title>  
  <author>J K. Rowling</author>  
  <year>2005</year>  
  <price>29.99</price>  
</book>
```

Relaciones entre Nodos

Ancestros

Un nodo padre, padre de padre, etc.

En el siguiente ejemplo; el ancestro del elemento title es el elemento book y el elemento bookstore:

```
<bookstore>
```

```
<book>
```

```
<title>Harry Potter</title>
```

```
<author>J K. Rowling</author>
```

```
<year>2005</year>
```

```
<price>29.99</price>
```

```
</book>
```

```
</bookstore>
```

Relaciones entre Nodos

Descendiente

Un nodo hijo, hijo de hijo, etc.

En el siguiente ejemplo; los descendientes del elemento bookstore son los elementos book, title, author, year, y price:

```
<bookstore>  
  <book>  
    <title>Harry Potter</title>  
    <author>J K. Rowling</author>  
    <year>2005</year>  
    <price>29.99</price>  
  </book>  
</bookstore>
```

Práctica

```
<p> </p>
```

```
<xml>
```

```
  <property>
```

```
    <name type="String">Juan</name>
```

```
    <lastName type="String">Cho</lastName>
```

```
    <age type="int">32</>
```

```
  </property>
```

```
</xml>
```

A close-up photograph of a person's hand holding a pen, poised to write on a document. The image is partially obscured by a solid blue vertical bar on the right side. The word "REPASO" is written in large, white, bold, sans-serif capital letters across the center of the image, overlapping both the photograph and the blue bar.

REPASO

A close-up photograph of a person's hand using a stylus to interact with a tablet. The background is blurred, showing some bokeh light effects. A solid blue vertical bar covers the right half of the image. The text 'SELENIUM IDE' is overlaid in white, bold, sans-serif font across the center.

SELENIUM IDE



Selenium IDE

- Es un entorno de desarrollo integrado para scripts de selenium.
- Se implementa como una extensión de Firefox, y permite grabar, editar y debugger Tests.

An aerial photograph of a city skyline at dusk or dawn. The sky is a mix of dark blue, purple, and orange. The city is densely packed with skyscrapers, many of which have their lights on. The Empire State Building is prominent in the center. The title 'Características de Selenium IDE' is overlaid in large white text.

Características de Selenium IDE

Características de Selenium IDE

1 - Fácil de grabar y reproducir un test

Características de Selenium IDE

- 1 - Fácil de grabar y reproducir un test
- 2 - Inteligente selector de ID, nombre o xpath

Características de Selenium IDE

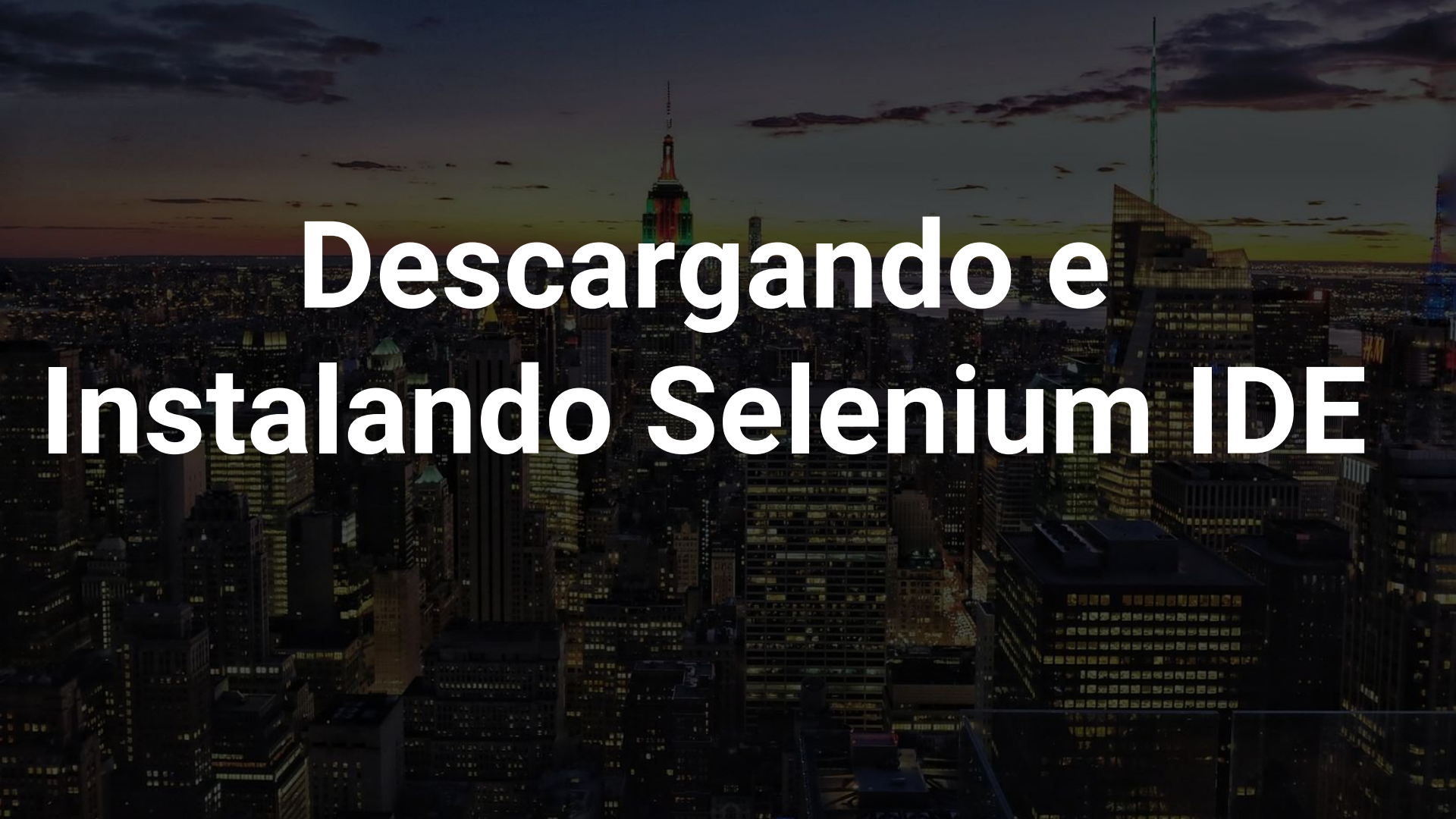
- 1 - Fácil de grabar y reproducir un test
- 2 - Inteligente selector de ID, nombre o xpath's
- 3 - Autocompletación de para los comandos de Selenium IDE

Características de Selenium IDE

- 1 - Fácil de grabar y reproducir un test
- 2 - Inteligente selector de ID, nombre o xpath's
- 3 - Autocompletación de para los comandos de Selenium IDE
- 4 - Exportar Test como HTML, JAVA, entre otros.

Características de Selenium IDE

- 1 - Fácil de grabar y reproducir un test
- 2 - Inteligente selector de ID, nombre o xpaths
- 3 - Autocompletación de para los comandos de Selenium IDE
- 4 - Exportar Test como HTML, JAVA, entre otros.
- 5 - Opción para verificar automáticamente el título de cada página

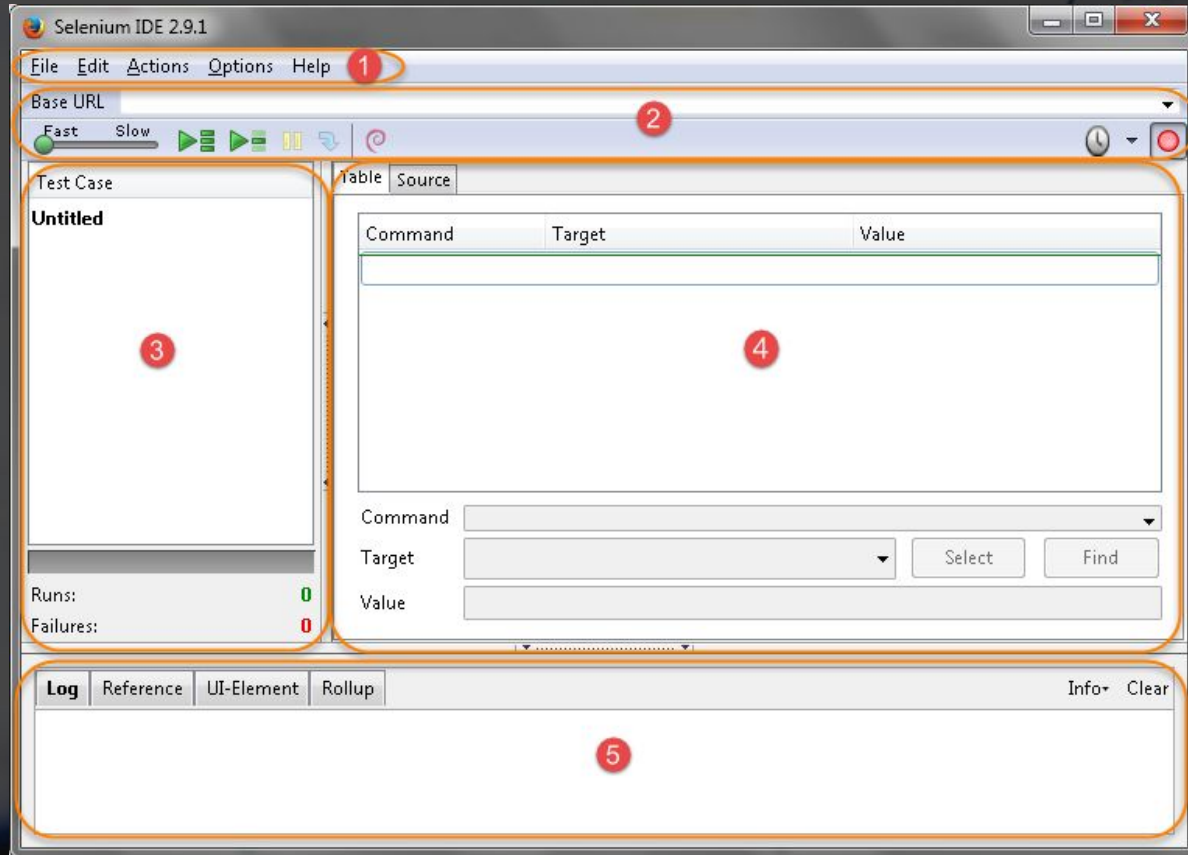


Descargando e Instalando Selenium IDE

Descargando e Instalando Selenium IDE

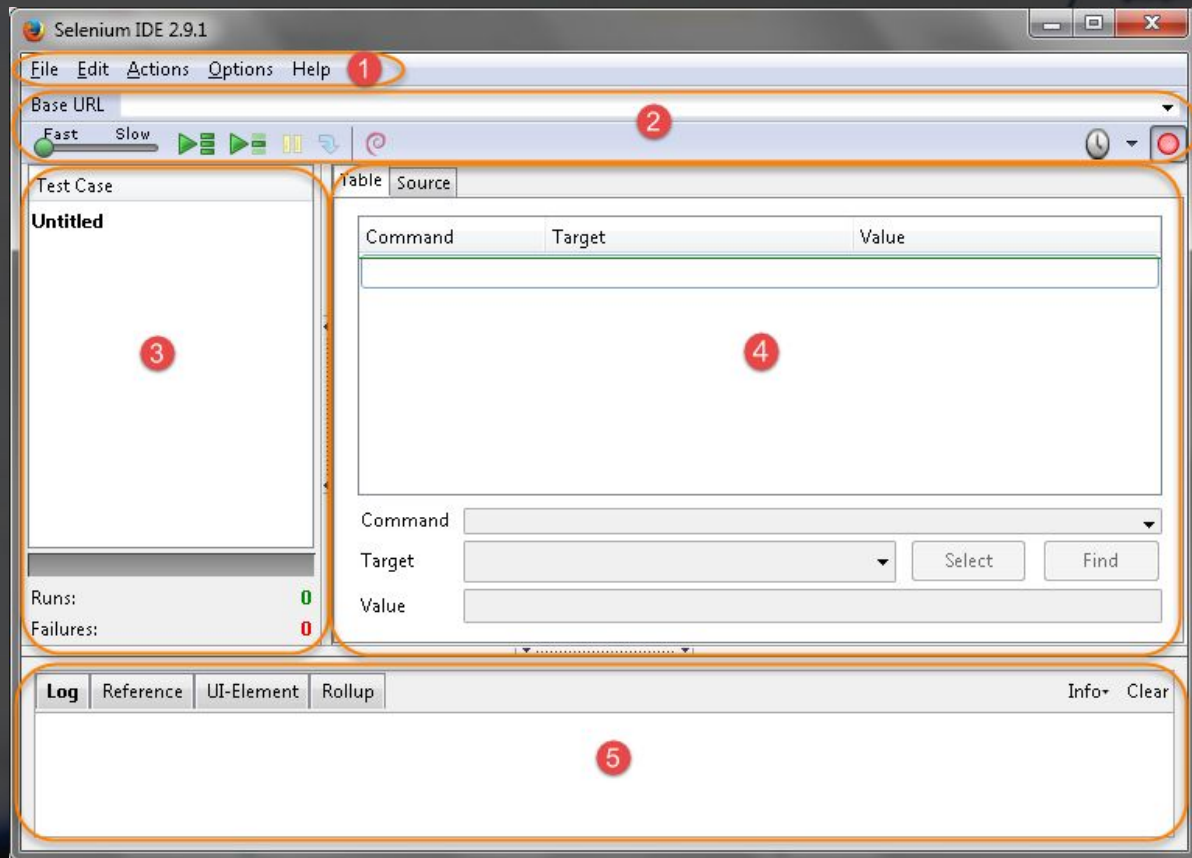
- 1 - Abrir el navegador Mozilla Firefox. (Instalarlo en caso de no tenerlo)
- 2 - Escribir <https://goo.gl/kchwPV>
- 3 - Click en Add to Firefox.
- 4 - Una vez que se haya abierto un popup, click en Install.
- 5 - Una vez instalado, restart Firefox
- 6 - Ir a Tools -> Selenium IDE

Selenium IDE



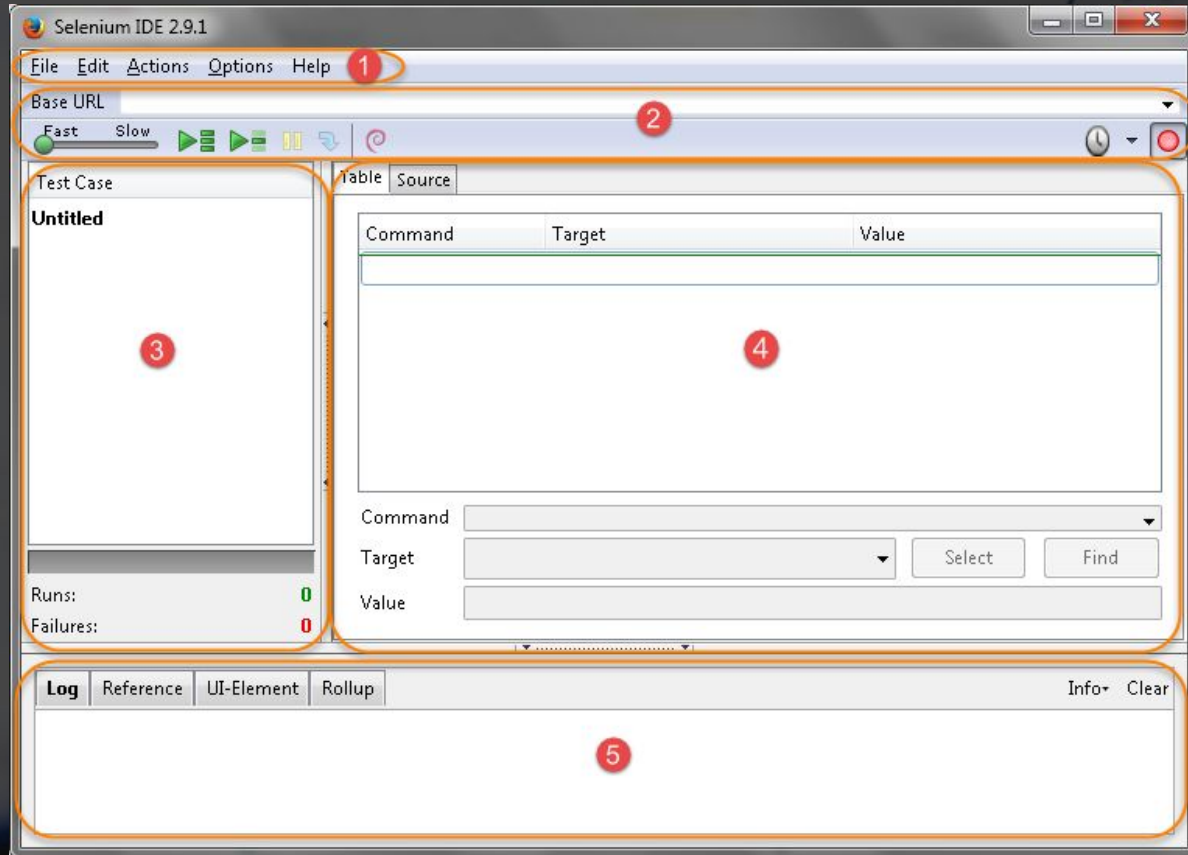
- Base URL muestra donde se están ejecutando las pruebas.
- Panel de Test Cases Muestra los casos de prueba abiertos.
- Panel de tabla Muestra una lista de acciones grabadas como parte de la prueba

Selenium IDE



- Fast Slow, configura la velocidad de la reproducción, rápida o lenta para la prueba grabada.
- El botón Play reproduce la prueba.

Selenium IDE



- El panel inferior muestra un registro de acciones y una lista de referencias
- El botón rojo es el interruptor de Record para grabar las interacciones de registro en cualquier sitio web.

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark blue and orange, with scattered clouds. The city is densely packed with skyscrapers, many of which are illuminated with their interior lights. The Empire State Building is prominent in the center, with its top lit in red and green. Other notable buildings include the Chrysler Building and the United Nations Secretariat Building. The text "Creando una prueba" is overlaid in a large, white, sans-serif font across the middle of the image.

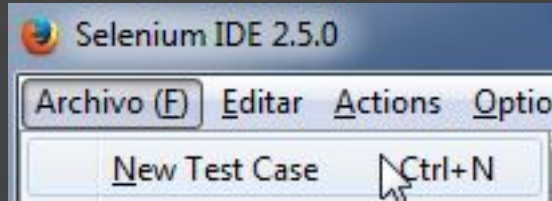
Creando una prueba

Creando una prueba

A modo de prueba, vamos a grabar un test que haga una búsqueda en Google.

1 - Crear uno nuevo Test de Prueba.

Archivo -> Nuevo Test Case



Creando una prueba

2 - Empezamos a grabar nuestra prueba. Para ello, pulsa el interruptor rojo.



2.1 - Abre un navegador de Firefox y escribe en “<https://www.google.es>” en la barra de direcciones.

2.2 - En la página principal de google.es, busca cualquier palabra. Para este ejemplo, voy a consultar la palabra “helloWorld java”.

2.3 - Una vez que la página de resultados se abre, detén la grabación haciendo clic en el botón rojo.

Creando una prueba

2 - Ahora que hemos grabado la prueba vamos a reproducir el test nuevamente.

Observar que hay un fallo.

¿Alguna idea de porque puede ser?

The screenshot shows the Selenium IDE 2.9.1 interface. The 'Test Case' tab is active, showing a test suite named 'Untitled *'. The test suite contains three commands: 'open' (target: /webhp?hl=es-419&sa=...), 'type' (target: id=lst-ib, value: hello wold java), and 'clickAndWait' (target: link=HelloWorld.java - I...). The 'clickAndWait' command is highlighted in orange. Below the commands, the 'Command' dropdown is set to 'clickAndWait', the 'Target' is 'link=HelloWorld.java - Int', and the 'Value' is empty. The 'Runs' section shows 1 run and 1 failure. The 'Log' section shows an error message: '[error] Element link=HelloWorld.java - Introduction to Programming in Java not found'. Below the error message, it says '[info] Test case failed', '[info] Test suite completed: 1 played, 1 failed', and '[info] Executing: [clickAndWait | link=HelloWorld.java - Introduction to Programming in Java |'.

Command	Target	Value
open	/webhp?hl=es-419&sa=...	
type	id=lst-ib	hello wold java
clickAndWait	link=HelloWorld.java - I...	

Command: clickAndWait
Target: link=HelloWorld.java - Int
Value:

Runs: 1
Failures: 1

[error] Element link=HelloWorld.java - Introduction to Programming in Java not found
[info] Test case failed
[info] Test suite completed: 1 played, 1 failed
[info] Executing: [clickAndWait | link=HelloWorld.java - Introduction to Programming in Java |

Creando una prueba

3 - Ahora que hemos grabado la prueba vamos a reproducir el test nuevamente.

The screenshot shows the Selenium IDE 2.9.1 interface. The 'Test Case' tab is active, showing a test suite named 'Untitled *'. The test suite contains three commands: 'open' (Target: /webhp?hl=es-419&sa=...), 'type' (Target: id=lst-ib, Value: hello wold java), and 'clickAndWait' (Target: link=HelloWorld.java - I...). The 'clickAndWait' command is highlighted in orange. Below the command list, the 'Runs' counter shows 1 and the 'Failures' counter shows 1. The 'Log' tab is active, showing an error message: '[error] Element link=HelloWorld.java - Introduction to Programming in Java not found'. Below the error message, the log shows: '[info] Test case failed', '[info] Test suite completed: 1 played, 1 failed', and '[info] Executing: [clickAndWait | link=HelloWorld.java - Introduction to Programming in Java |'.

Command	Target	Value
open	/webhp?hl=es-419&sa=...	
type	id=lst-ib	hello wold java
clickAndWait	link=HelloWorld.java - I...	

Runs: 1
Failures: 1

[error] Element link=HelloWorld.java - Introduction to Programming in Java not found

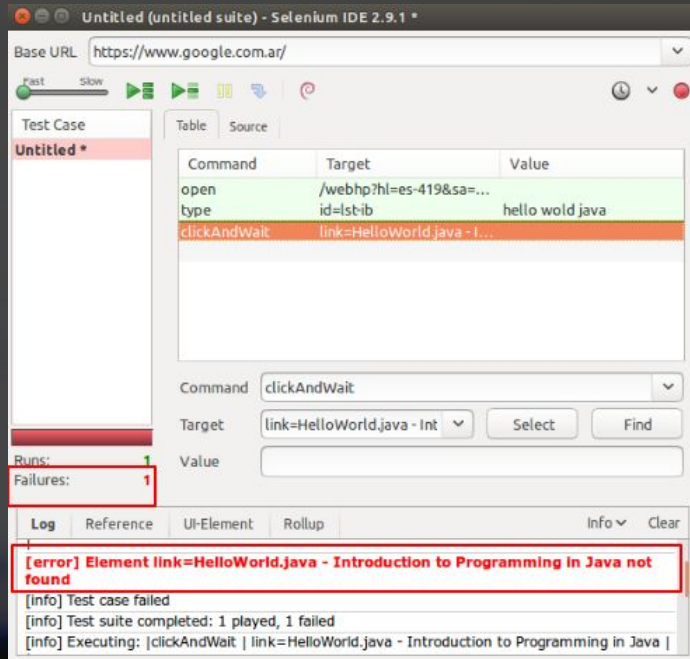
[info] Test case failed
[info] Test suite completed: 1 played, 1 failed
[info] Executing: [clickAndWait | link=HelloWorld.java - Introduction to Programming in Java |

El fallo se debe a que selenium está intentando **clickear en el link antes de que la página esté cargada**. Para solucionar esto hacer lo siguiente:

- 1 - Click derecho en el comando de ClickAndWait, y seleccionamos "Insert new command".
- 2 - Agregar el siguiente comando `waitForPageToLoad` (Target: 1000)
- 3 - Volver a Ejecutar el Test.

Creando una prueba

4 - Ahora que hemos grabado la prueba vamos a reproducir el test nuevamente.

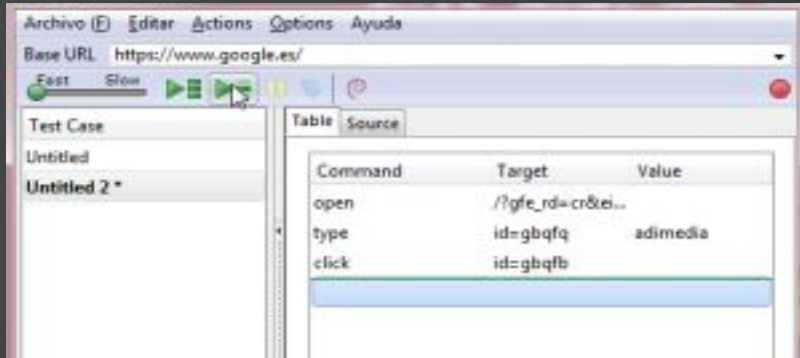


Otra posible solución

- 1 - Ir a Options -> Options
- 2 - Actualizar el campo llamado 'Default time value ...' por 50000
- 3 - Guardar los cambios
- 4 - Reducir la velocidad con la que se ejecutan los Test

Creando una prueba

5 - Ahora que hemos grabado la prueba vamos a reproducir el test nuevamente.



Este Test ejecutara 4 acciones:

- open: abre automáticamente "www.google.es"
- type: escribe "hello world java" en el campo de consulta.
- Espera 10 segundos
- click: pulsa el botón "Buscar con Google" y despliega los resultados de búsqueda.

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark purple, blue, and orange. The city lights are visible, with the Empire State Building's red and green top being a prominent feature. The word "Ejercicios" is written in a large, white, sans-serif font across the center of the image.

Ejercicios

Ejercicios

Scenario 1

- Ir a google
- Buscar <http://www.infobae.com>, hacer click en el primer link encontrado
- Ir a 'Ultimas Noticias'
- Verificar que Ultimas Noticias esta presente

Usar : `assertElementPresent`

Ejercicios

Scenario 1

- Ir a google
 - Buscar <http://www.infobae.com>, hacer click en el primer link encontrado
 - Ir a 'Ultimas Noticias'
 - Verificar que Ultimas Noticias esta presente
- Usar : `assertElementPresent`

Scenario 2 (Ejercicio_2)

- Ir a google.com
- Buscar la palabra 'bored panda', hacer click en el primer link encontrado
- Ir a 'Latest'
- Guardar el Test con el nombre de 'Ejercicio_2'

Ejercicios

Scenario 1 (Ejercicio_1)

- Ir a google
 - Buscar <http://www.infobae.com>, hacer click en el primer link encontrado
 - Ir a 'Ultimas Noticias'
 - Verificar que Ultimas Noticias esta presente
- Usar : assertElementPresent

Scenario 2 (Ejercicio_2)

- Ir a google.com
- Buscar la palabra 'bored panda', hacer click en el primer link encontrado
- Ir a 'Latest'
- Guardar el Test con el nombre de 'Ejercicio_2'

Scenario 3 (Ejercicio_3)

- Importar Ejercicio_2
- Buscar la palabra photoshop
- Ingresar al primer link encontrado

An aerial photograph of a city skyline at dusk or dawn. The sky is a mix of dark blue and orange, with some clouds. The city is densely packed with skyscrapers, many of which are lit up with lights. The title 'Tipos de Comandos de Selenium IDE' is overlaid in large, white, sans-serif font in the center of the image.

Tipos de Comandos de Selenium IDE

Tipos de comandos de Selenium IDE

- **Acciones**
- **Accesores**
- **Aserciones**

Tipos comandos de Selenium IDE

- **Acciones**

Comandos que interactúan con la aplicación modificando o ingresando algún dato.

Ejemplo :

- Click
- Type

Tipos comandos de Selenium IDE

- **Accessors**

Comandos que permiten al usuario almacenar datos en una variable definida por el usuario.

Estos valores se pueden usar luego para crear aserciones y verificaciones.

Ejemplo: `storeAllLinks`

Tipos comandos de Selenium IDE

- **Aserciones**

No interactúan con la App al igual que los Accessors. Son usados para comprobar el estado actual de la aplicación con un estado esperado

Ejemplo:

- Assert
- Verify
- waitFor

An aerial photograph of a city skyline at dusk or dawn. The sky is a mix of dark blue and orange, with some clouds. The city is densely packed with skyscrapers, many of which are lit up with their interior lights. The Empire State Building is prominent in the center. The text 'Comandos utilizados con más frecuencia' is overlaid in a large, white, sans-serif font, centered horizontally and vertically.

Comandos utilizados con más frecuencia

Comandos utilizados con más frecuencia

- **Open**

Abre una específica URL en el browser

i.e: Open - Target: <https://www.google.com.ar>

- **assertTitle, VerifyTitle**

Retorna el título de la página actual, y lo compara con el título especificado

i.e: assertTitle - Target: Google

Comandos utilizados con más frecuencia

- **assertElementPresent, verifyElementPresent**
Verify / Asserts la presencia de un elemento en una página web.
i.e: verifyElementPresent - Target: id= {someID}
- **assertTextPresent, verifyTextPresent**
Verify / Asserts la presencia de un texto en una página web.
i.e:

Comandos utilizados con más frecuencia

- **type, typeKeys, sendKeys**

Ingresa un valor(String) en el elemento web especificado.

Argumentos: 2

i.e : type - Target id={someID} - Value : {someValue}

- **Click, clickAt, clickAndWait**

Clicks en un elemento web especificado

i.e: clickAndWait - target id={someID}

Comandos utilizados con más frecuencia

- **waitForPageToLoad**

Detiene la ejecución y espera hasta que la pagina este completamente cargada.

i.e: `waitForPageToLoad - Target: 10000` (Espera 10 seg)

- **waitForElementPresent**

Detiene la ejecución y espera hasta que el elemento especificado esta presente.

i.e: `waitForElementPresent - Target: id={someID}`

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark blue and orange, with scattered clouds. The city lights are visible, and the Empire State Building stands out prominently in the center. Overlaid on the image is the text '¿Qué es Firebug?' in a large, white, sans-serif font.

¿Qué es Firebug?

¿Qué es Firebug?

- Es un addon de Mozilla
- Esta herramienta nos ayuda a identificar elementos unívocamente en una página web.
- Para instalarlo
 - Ir a <https://goo.gl/ht3co>
 - Click en Add to Firefox.
 - Continuar con la instalación.

An aerial photograph of a city skyline at dusk or dawn. The sky is a mix of dark blue, purple, and orange. The city is densely packed with skyscrapers, many of which have their lights on. The Empire State Building is prominent in the center. The text "Usando Firebug para crear Selenium Scripts" is overlaid in a large, white, sans-serif font.

Usando Firebug para crear Selenium Scripts

Usando Firebug para crear Selenium Scripts

- Open Google.com
- Verificar que el usuario se encuentra en Google
- Buscar Unilad y entrar al link encontrado
- Verificar que el usuario se encuentra en Unilad (Ejercicio_5)

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark purple, blue, and orange. The city lights are visible, with the Empire State Building's red and green top being a prominent feature. The word "Ejercicios" is written in a large, white, sans-serif font across the center of the image.

Ejercicios

Ejercicios

Scenario 1 (Ejercicio_4)

- Ir a google
- Buscar <https://accounts.google.com> , hacer click en el primer link encontrado
- Verificar que el usuario se encuentra en la página correcta
- Insertar Usuario
- Clickear en Next
- Insertar Contraseña
- Clickear en Login
- Verificar que el usuario se encuentra en la página correcta

Ejercicios

Scenario 1 (Ejercicio_4)

- Ir a google
- Buscar <https://accounts.google.com> , hacer click en el primer link encontrado
- Verificar que el usuario se encuentra en la página correcta
- Insertar Usuario
- Clickear en Next
- Insertar Contraseña
- Clickear en Login
- Verificar que el usuario se encuentra en la página correcta

Scenario 2 (Ejercicio_6)

- Ir a google
- Buscar <https://accounts.google.com> , hacer click en el primer link encontrado
- Desloguear al usuario
- Verificar que el usuario se encuentra en la página correcta
- Insertar Usuario
- Clickear en Next
- Insertar Contraseña incorrecta
- Verificar que el usuario no se ha podido loguear correctamente



Ejercicios Locales

Ejercicios

Scenario 1

- Abrir "Local Webs/busqueda/index.html"
- Buscar "gatos"
- Entrar a "Gatos"
- Elegir el segundo gato
- Verificar que se halla abierto

Scenario 2

- Abrir "Local Webs/busqueda/index.html"
- Buscar alguna cosa
- Entrar a "Wikipedia..."
- Verificar que este en la página de wikipedia correcta

Scenario 3

- Abrir "Local Webs/busqueda/index.html"
- Buscar "la comida"
- Entrar a "Noticias"
- Clickear la ultima noticia"
- Verificar que te lleve a una página (que no sea error).

Ejercicios

Scenario 4


- Abrir "Local Webs/busqueda/index.html"
- Buscar algo
- Entrar a "Películas XD"
- Hacer click en el poster de ferdiland
- Verificar que se cargue google

Scenario 5

- Abrir "Local Webs/busqueda/index.html"
- Buscar alguna cosa
- Entrar a "... 100% gratis"
- Hacer click en flashget
- Verificar que te lleve a una página que no es softonic

Scenario 6

- Abrir "Local Webs/busqueda/index.html"
- Buscar "Notebook"
- Entrar a "Compra venta"
- Clickear la Notebook
- Verificar que te lleve a la página de busqueda.

A close-up photograph of a person's hand holding a stylus, poised to write on a tablet. The background is out of focus, showing some bokeh light effects. The image is partially covered by a blue overlay on the right side.

Firefox <= 35.0.1

Instalando Firefox 35.0.1

1 - Instalación: WINDOWS

- Descargar de [aquí](#).

LINUX

- Seguir tutorial [aquí](#).
- Una vez terminada la instalación, abrir una terminal y ejecutar `> sudo apt-mark hold firefox` (de esta forma evitamos que firefox upgrade)



JDK

Instalando JDK

- 1 - Descargar JDK desde [aquí](#).
 - 2 - Instalación: WINDOWS
 - Seguir tutorial [aquí](#).
- LINUX
- Seguir tutorial [aquí](#).

A close-up photograph of a person's hand holding a white marker, writing on a whiteboard. The hand is positioned in the lower-left quadrant of the frame. The whiteboard surface is visible, showing some faint, illegible markings. A blue gradient overlay covers the right half of the image, starting from a vertical line. The text "JAVA_HOME" is written in large, white, bold, sans-serif capital letters across the center of the image, partially overlapping the hand and the blue gradient.

JAVA_HOME

Seteando JAVA_HOME

- Precondición
JDK debe estar instalado
- Seteo de JAVA_HOME: WINDOWS
 - Seguir tutorial [aquí](#).

LINUX

- Seguir tutorial [aquí](#). (Chequear casi al final de la página)

A close-up photograph of a person's hand holding a pen, writing on a document. The image is split vertically: the left side shows the hand and pen in a dimly lit setting with bokeh lights in the background, while the right side is a solid blue bar. The word "MAVEN" is written in large, white, bold, sans-serif capital letters across the center, overlapping both the photograph and the blue bar.

MAVEN

Instalando Maven

1 - Descargar Maven de [aquí](#).

2 - Verificar que JAVA_HOME está configurada correctamente y que JDK está instalado correctamente.

3 - Instalar Maven : WINDOWS

- Seguir tutorial de [aquí](#).
- Seguir tutorial de [aquí](#).

LINUX

- Seguir tutorial de [aquí](#).

Maven 3.2 requiere JDK 1.6 o superior

Maven 3.0/3.1 requiere JDK 1.5 o superior



Selenium Web Driver(SWD)

IMPORTANTE: SWD es solo compatible con versión <= Firefox 35.0.1

Instalando SWD con Maven

1 - Abrir IntelliJ Idea

2 - Crear Maven project

GroupID: com.{nombre_proyecto}.selenium

artifactID: {nombre_proyecto}

3 - Abrir nombre_proyecto.xml y agregar <properties>

```
<properties>
```

```
  <jdk.level>1.7</jdk.level>
```

```
</properties>
```


Instalando SWD con Maven

4 - Agregar dependencias

```
<dependencies>
  <dependency> <!--TESTNG-->
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>6.8.8</version>
  </dependency>
  <dependency> <!--Run Selenium Test-->
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>2.53.1</version>
  </dependency>
</dependencies>
```

Instalando SWD con Maven

5 - Agregar los siguientes plugins

```
<plugin> <!-- Used to compile code-->
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.5.1</version>
  <configuration>
    <source>${jdk.level}</source>
    <target>${jdk.level}</target>
  </configuration>
</plugin>
<plugin> <!--It's used to execute tests-->
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.17</version>
  <configuration>
    <suiteXmlFiles>
      <suiteXmlFile>src/test/resources/testng.xml</suiteXmlFile>
    </suiteXmlFiles>
  </configuration>
</plugin>
```

6 - Nuestro archivo debe quedar algo similar a :

<https://pastebin.com/ygWuUJEZ>

Instalando SWD con Maven

7 - Crear un nuevo file (extension Java) en src/main/java

8 - Abrir el archivo creado y agregar la siguientes líneas

Agregar Import

```
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.testng.Assert;  
import org.testng.annotations.AfterClass;  
import org.testng.annotations.BeforeClass;  
import org.testng.annotations.Test;
```

Agregar variables

```
private WebDriver driver;  
String appURL = "http://google.com";
```

Instalando SWD con Maven

8 - Abrir el archivo creado y agregar la siguientes líneas
Agregar métodos

```
@BeforeClass
// Setting FirefoxDriver driver
public void testSetUp() { driver = new FirefoxDriver(); }
@Test
// Ejecuting Test
public void verifyGooglePageTittle() {
    driver.navigate().to(appURL);
    String getTitle = driver.getTitle();
    Assert.assertEquals(getTitle, "Googale");
}
@AfterClass
// Closing Browser when finish the test
public void tearDown() { driver.quit(); }
```

9 - Nuestro archivo debe quedar algo similar a :

<https://pastebin.com/8WnrgGmw>

Instalando SWD con Maven

10 - Crear un nuevo file llamado testng.xml en test/resources/

11 - Abrir el archivo creado y agregar la siguientes líneas

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Example test run">
  <test name="Simple Test">
    <classes>
      <class name="{nombre_archivo_creado_en_paso_7}"/>
    </classes>
  </test>
</suite>
```

12 - Compilar project -> mvn clean compile

13 - Ejecutar project -> mvn clean install

An overhead photograph of a group of people sitting on a dark wooden deck. The scene is dimly lit, with shadows cast across the planks. In the center, a large white question mark is superimposed over the image. On either side of the question mark, there are decorative elements consisting of blue squares arranged in a grid-like pattern. The people are sitting on the ground, some with their backs to the camera, and their shadows are visible on the deck.

¿Preguntas?