

Temario SeU Certified Selenium Engineer (CSE)

Versión lanzada en el 2018

Selenium United



[Aviso de derechos de autor](#)

Este documento puede ser copiado parcialmente o en su totalidad, si se reconoce la fuente.

Todos los materiales de Selenium United, incluyendo este documento, son propiedad © de Selenium United (en adelante denominado SeU).

Los autores del material y los expertos internacionales involucrados en la creación de los recursos de la SeU transfieren los derechos de autor a Selenium United (SeU). Los autores del material, los expertos internacionales y la SeU han acordado las siguientes condiciones de uso:

- Cualquier persona o empresa de capacitación puede utilizar este temario como base para un curso de capacitación si SeU es reconocido como la fuente y propietario de los derechos de autor del temario, y sólo después de haber sido reconocida oficialmente por SeU. Más información sobre el reconocimiento está disponible en: <https://www.selenium-united.com/recognition>.
- Cualquier individuo o grupo de individuos puede usar este temario como base para artículos, libros u otros escritos derivados si SeU es reconocido como la fuente y propietaria de los derechos de autor del temario.

Gracias a los principales colaboradores

Rahul Verma, Vipul Kocher, Chandra Mouli Maddala and Jayapradeep Jiothis

Historial de revisiones

Versión	Fecha	Observaciones
SeU 2018	Agosto 2018	Primer lanzamiento oficial

Índice de contenidos

<i>Finalidad de este documento</i>	Error! Bookmark not defined.
<i>Recursos de SeU</i>	4
<i>¿Qué es Selenium?</i>	4
<i>Acerca del SeU Certified Selenium Engineer (CSE)</i>	5
<i>Resultados de las actividades</i>	Error! Bookmark not defined.
<i>Objetivos del aprendizaje/Niveles cognitivos del conocimiento</i>	5
<i>Prerequisitos de conocimiento en Java</i>	6
<i>Capítulo 1 – Automatización IU Web</i>	Error! Bookmark not defined.
<i>Capítulo 2 – Introducción a Selenium</i>	Error! Bookmark not defined.
<i>Capítulo 3 – Automatización de IU Web con Selenium</i>	11
<i>Capítulo 4 – Más allá de la simple construcción del código Selenium</i>	13
<i>Capítulo 5 – Elaboración de un marco básico</i>	16
<i>Referencias:</i>	17

Propósito de este documento

Este temario constituye la base de Selenium United en lo que respecta a su certificación de Certified Selenium Engineer (CSE). Este documento define lo que usted necesita saber para pasar la prueba de certificación para Certified Selenium Engineer (CSE), del cual los derechos de autor son propiedad de Selenium United. El examen de certificación sólo cubrirá los conceptos y conocimientos que se describen en este documento.

Recursos de SeU

Una visión general de los recursos de SeU, así como toda la información relevante sobre la certificación SeU y otros tipos de certificaciones SeU están disponibles en www.selenium-united.com — el sitio web oficial de Selenium United. La información en www.selenium-united.org incluye:

- Una lista completa de los proveedores de capacitación reconocidos de la SeU y de los cursos disponibles. Tenga en cuenta que se recomienda la capacitación, pero no es necesaria, para realizar el examen de certificación SeU CSE.
- El temario SeU (este documento) para descargar.
- Una muestra completa de un conjunto de 40 preguntas de SeU CSE con respuestas, para fines de entrenamiento.
- Nuestro objetivo es que los documentos estén disponibles en otros idiomas lo antes posible. Para ver las versiones en los idiomas actualmente disponibles, por favor visite www.selenium-united.com

¿Qué es Selenium?

Selenium es un conjunto de diferentes herramientas destinadas a la automatización de navegadores, incluyendo Selenium WebDriver y Selenium Grid. Aunque no se limita a ser utilizado para probar aplicaciones web, se ha convertido en la herramienta más buscada por los testers de software. Con soporte para todos los navegadores ampliamente utilizados, también ha crecido más allá de ser otra biblioteca de pruebas de software. Es la columna vertebral de innumerables herramientas de automatización de navegadores, interfaces de programación de aplicaciones APIs y marcos de trabajo. La interfaz WebDriver introducida por Selenium WebDriver está siendo revisada para convertirse en un estándar W3C, lo que aumentaría aún más su importancia en el espacio de la automatización de pruebas.

Acerca del SeU Certified Selenium Engineer (CSE)

SeU Certified Selenium Engineer (CSE) es un curso de nivel profesional para testers involucrados en la automatización de pruebas web. El curso cubre a Selenium como una biblioteca de automatización de navegadores desde cero. Las construcciones y el diseño de automatización de pruebas se mantienen al mínimo para centrarse más en las construcciones de código que permiten el uso de Selenium de una manera correcta. Este curso, centrado en el diseño, proporciona una experiencia de Selenium no diluida, enfocándose más en los conceptos críticos de Selenium, lo que permite una mejor implementación en el trabajo diario de los participantes.

Resultados de las actividades (RAs)

RA 1	Adaptar la experiencia y el conocimiento existente en pruebas y automatización de pruebas para desarrollar pruebas automatizadas para aplicaciones web utilizando Selenium.
RA 2	Utilizar Selenium para crear pruebas de automatización para aplicaciones Web.
RA 3	Depurar pruebas automatizadas basadas en Selenium para comprobar la funcionalidad correcta.

Objetivos de aprendizaje/Niveles cognitivos de conocimiento

Los objetivos de aprendizaje (OAs) son declaraciones breves que describen lo que se espera que usted sepa después de estudiar cada capítulo. Los OAs se definen como se indica a continuación:

- K1: Recordar
- K2: Entender
- K3: Aplicar

La lista principal de los OAs para la certificación SeU CSE:

OA1	Comprender la importancia de la cobertura del navegador y distinguir varias opciones para probar la interfaz de usuario (IU) de las aplicaciones web. (K2)
OA2	Comprender la relación de la interfaz web con el HTML correspondiente y JavaScript con la inspección DOM. (K2)

OA3	Recordar la Historia de Selenium y varias herramientas en su paquete, junto con su propósito. (K1)
OA4	Comprender la arquitectura de Selenium en términos de enlaces de lenguaje, protocolos de comunicación y controladores (Drivers). (K2)
OA5	Reconocer el propósito y la interfaz de programación de aplicaciones API de la automatización de la interfaz de usuario (IU) web a niveles del navegador, de la página y de los elementos. (K2)
OA6	Comprender y explicar el propósito de las indicaciones, elementos fijos y afirmaciones de JUnit. (K3)
OA7	Aplicar diferentes estrategias de identificación para los elementos de la IU. (K3)
OA8	Aplicar diferentes estrategias de investigación e interacción para los elementos de la IU. (K3)
OA9	Aplicar varias construcciones de automatización de Selenium más profundas, que establecen las bases para una automatización más avanzada. (K3)
OA10	Automatizar escenarios de usuario final utilizando buenas prácticas de codificación y principios orientados a objetos para colocar el código de Selenium en diferentes clases y métodos. (K3)
OA11	Solucionar problemas y describir el alcance de las mejoras para la implementación de la automatización que se muestran en ejemplos de códigos cortos. (K3)

Prerequisitos de conocimiento en Java

El facilitador del taller informará sobre los conceptos de Java a medida que se vayan utilizando en el curso. Sin embargo, estos conceptos sólo se mencionarán y no se explicarán, a menos que se añada un día al taller para explicar esta cobertura adicional.

Los participantes que estén bien familiarizados con los siguientes conceptos podrán enfocarse en los conceptos de Selenium de una manera mucho mejor, sin desviar su atención para entender las construcciones de Java.

No se incluirán en el examen preguntas directas sobre el contenido básico de Java.

- Concepto del método principal()
- Compilación y ejecución del código Java
- Tipos de datos primitivos
- Contrapartes de clase de tipos de datos primitivos (por ej. “Integer” para “int”)
- Arreglos
- Colecciones básicas: Lista, Mapa
- Formateo y manipulación
- “System.out” (salida) y “System.err” (error)
- Estructuras de control condicional: si/otra/o si, cambiar
- Estructuras de control de ciclo: para (los 2 estilos), en tanto que
- Manejo de Excepciones – lanzamiento, lanzamientos, “try-catch-fin” (prueba y atrapa), Jerarquía de Excepciones
- Escribir una clase con “getter”, “setter” y otros métodos
- Constructores e instanciación de clases
- Variables de clase vs variables de objeto
- Modificadores de acceso – público, privado, protegido
- Métodos de clase frente a métodos de objeto
- Enum
- Paquetes
- Sobrecarga de métodos
- Sucesiones
- Clases abstractas y métodos abstractos
- Anulación de métodos
- Composición del objeto

Capítulo 1 - Automatización de la interfaz de usuario (IU) Web

OA1	Comprender la importancia de la cobertura del navegador y distinguir varias opciones para probar la IU de las aplicaciones web. (K2)
OA2	Comprender la relación de la interfaz de usuario (IU) Web con el HTML básico y JavaScript con la inspección DOM. (K2)

Resumen:

- Introducción
- Automatización de la IU con navegadores reales
- Automatización de la IU con navegadores reales con simulación del tamaño de la pantalla
- Uso de los navegadores “Headless” (sin cabezal)
- IU Web: Punto de vista Usuario vs. Navegador

Antecedentes:

Las aplicaciones Web deben estar disponibles para una variedad de usuarios que utilizan una variedad de dispositivos. Muchos de ellos emplean interfaces fluidas para que las aplicaciones se rendericen de forma diferente para los distintos navegadores.

Los propios navegadores se comportan de forma diferente en determinadas situaciones, ya que tienen diferentes tamaños de lienzo y tecnología subyacente.

Esto hace que la cobertura del navegador sea un aspecto esencial en las pruebas de la interfaz de la aplicación IU web, tanto si se realiza con un código humano como con un fragmento de código. Un ingeniero de automatización de pruebas de Selenium debe entender cómo utilizar Selenium para automatizar diferentes navegadores junto con diferentes opciones para imitar a un navegador. Por razones de velocidad y eficiencia, el uso de un navegador sin cabezal en la automatización es otra opción que el ingeniero debe tener en cuenta.

Los navegadores muestran los controles de la interfaz de usuario que están representados en HTML para cargarlos en un objeto DOM y representarlos visualmente en su lienzo. Un ingeniero de automatización de pruebas debe ser capaz de asociar lo que se ve en la pantalla a su definición en HTML/DOM utilizando la inspección DOM. Esta información se utiliza como entrada al

código de automatización de pruebas para la identificación e interacción con dichos elementos.

Las aplicaciones web tienen diferentes tipos de elementos de interfaz de usuario que soportan diferentes tipos de acciones que un usuario puede realizar, por ejemplo, puede hacer clic en un botón, introducir texto en un cuadro de texto, seleccionar una opción de una lista desplegable, etc. Un usuario también puede verificar visualmente el estado de un elemento de la interfaz de usuario en particular, por ejemplo, si está habilitado, qué texto contiene, etc. Un ingeniero de automatización de pruebas debe ser capaz de realizar todas estas investigaciones y tomar todo tipo de medidas en una prueba automatizada.

Un ingeniero de automatización de pruebas también debe estar al tanto del estilo CSS y de las partes de HTML en las que se puede especificar. Esto también es cierto para JavaScript, que es el principal lenguaje de scripting del lado del cliente utilizado por las aplicaciones web modernas.

Capítulo 2 – Introducción a Selenium

OA3	Revisar la historia Selenium y varias herramientas en su gama junto con su propósito. (K1)
-----	--

Resumen:

- Historia
- El poder de Selenium
- Gama ("Suite") Selenium
- Arquitectura simplificada de Selenium.

Antecedentes:

Con sus inicios en “ThoughtWorks” y las contribuciones posteriores de diferentes personalidades de la comunidad, Selenium ha pasado a la vanguardia de la automatización de la IU web. La API Selenium WebDriver, que se ha convertido en un estándar del W3C, es utilizada por varias herramientas que no son de Selenium, como Appium, para proporcionar la interfaz a su implementación básica..

Es bueno conocer la historia de Selenium y mantenerse al día con su plan de desarrollo para que los ingenieros de automatización de pruebas sepan cuándo y por qué se introdujeron ciertas características y hacia dónde se dirige el futuro de Selenium y de las herramientas relacionadas.

Selenium permite enlaces de múltiples lenguajes debido a la arquitectura que separa las responsabilidades entre los enlaces de cliente específicos del lenguaje y un componente de controlador específico del navegador que emplea el protocolo "JSON Wire". Entender esta arquitectura ayuda a un ingeniero de automatización de pruebas a saber qué componentes proporcionan qué funciones.

Capítulo 3 - Autmatizando la “IU Web” con Selenium

OA4	Comprender la arquitectura de Selenium en términos de enlaces de lenguaje, protocolos de comunicación y controladores. (K2)
OA5	Reconocer el propósito y la API de la automatización de la IU web a nivel de navegador, página y elemento. (K2)
OA6	Comprender y explicar el propósito de las anotaciones, accesorios y afirmaciones de JUnit. (K3)
OA7	Aplicar diferentes estrategias de identificación para los elementos IU. (K3)
OA8	Aplicar diferentes estrategias de investigación e interacción para los elementos de IU. (K3)

Resumen:

- Introducción
- Automatización del nivel del navegador
 - Lanzamiento/cierre de diferentes navegadores
 - Navegación
 - Consultar la Ventana y la información de la URL
- Automatización del nivel de página
 - Consultar información de nivel de página
 - Identificación de elementos a profundidad
 - ID
 - Nombre
 - Nombre de la clase
 - Texto del enlace
 - Texto parcial del enlace
 - Selectores CSS - cobertura de diferentes variantes
 - Xpath – cobertura de diferentes variantes
- Automatización del nivel de elementos
 - Consulta de estado
 - Acciones básicas

Antecedentes:

Para la automatización de la IU web, se necesita una API de automatización en 3 niveles:

1. Nivel de Navegador - Estas acciones no dependen de una página web en particular y son proporcionadas por la interfaz en Selenium.

2. Nivel de página - Estas acciones dependen de la página actual cargada en el navegador y también son proporcionadas por la interfaz de WebDriver.

3. Nivel de elemento IU: estas acciones están relacionadas con un elemento de interfaz de usuario en particular. La mayoría de estos son proporcionados por la interfaz de WebElement. Las acciones específicas del menú desplegable son proporcionadas por "Select class", como se discutirá más adelante en el capítulo de construcciones de automatización avanzada.

Un ingeniero de automatización de pruebas debe saber qué acciones soporta Selenium a través de su API en todos estos niveles. La expectativa no es que el ingeniero recuerde toda la API de memoria. Sin embargo, el ingeniero debe estar equipado para referir la API de Selenium y reconocer el propósito de las diferentes llamadas a la API y debe ser capaz de utilizarlas mientras escribe un código de automatización de pruebas.

En el proceso, el ingeniero debe saber cómo utilizar un motor de automatización de pruebas para utilizar diferentes características estándar, tales como representación de pruebas, dispositivos de prueba, aserciones, para utilizar junto con el código de Selenium a fin de escribir una prueba automatizada.

Para la identificación de elementos, Selenium soporta varias estrategias de localización a través de su objeto "By"(por):

- By (por) nombre
- By (por) ID
- By (por) Nombre de clase
- By (por) Nombre de etiqueta
- By (por) Texto del enlace y texto parcial del enlace
- By (por) Selector CSS
- By (por) XPath

Un ingeniero de automatización de pruebas debe conocer el propósito de cada una de estas estrategias de localización. El/ella debe ser capaz de elegir las estrategias de localización correctas para una situación dada. Los selectores CSS y XPath son extremadamente potentes y proporcionan múltiples formas de definir una estrategia de localización. El ingeniero de automatización de pruebas debe entender en profundidad los conceptos de CSS Selector y XPath, así como las implicaciones de flexibilidad y rendimiento de las diferentes estrategias.

Las interacciones con los elementos IU también necesitan consultas previas y posteriores para comprobar si se puede realizar una acción y si ésta se ha llevado a cabo con éxito.

Cápítulo 4 – Más allá de la simple construcción del código Selenium

OA9	Aplicar varias construcciones de automatización de Selenium más profundas, que establecen las bases para una automatización más avanzada. (K3)
-----	--

Resumen:

- Mejor espera
- Manejo de listas desplegadas
- Combinación de varios elementos
- Manipulación de elementos anidados
- Subir un archivo
- Ejecución de JavaScript
- Manejo de Ventanas/Pestañas
- Manipulación de marcos
- Manejo de alertas
- Cómo tomar capturas de pantalla
- Cadenas de acción
- Acciones del teclado
- Manejo de Cookies
- Navegación sin encabezado con HtmlUnitDriver

Antecedentes:

Debido a la lentitud de la red y otras operaciones del lado del cliente, es posible que la web IU no se encuentre en un estado en el que se pueda llevar a cabo la acción deseada. Por ejemplo, es posible que desee hacer clic en un botón; sin embargo, ese botón aún no se ha renderizado, o en el botón aún no se puede hacer clic. Una prueba automatizada, en la práctica, contiene el código para esperar el estado deseado antes de tomar una acción. Esta espera también es necesaria antes de que se pueda identificar un elemento para esperar a que se cargue la página. En lugar de utilizar esperas codificadas, las pruebas deben utilizar un mecanismo de espera basado en sondeo proporcionado por el mecanismo "Explicit Wait"(espera explícita) en Selenium, utilizando el objeto WebDriverWait.

Selenium proporciona una clase de selección para habilitar una API de nivel superior para listas desplegables.

Hay situaciones en las que uno necesitaría identificar múltiples elementos según las estrategias de un localizador y actuar en consecuencia. Hay otras situaciones en las que los elementos no se encuentran en la raíz de DOM, sino como elementos hijos dentro de un elemento.

Cargar un archivo es una característica bastante común en las aplicaciones web y el código de automatización debería ser capaz de simular esta característica.

Selenium también proporciona una clase JavaScript Ejecutor para permitir la ejecución de JavaScript en bruto en el contexto de un navegador. Esta es una característica muy útil, mediante la cual un ingeniero de automatización de pruebas puede inyectar cualquier tipo de JavaScript personalizado en el navegador y ejecutarlo.

Otras funciones necesarias en las pruebas automatizadas son la manipulación de ventanas/pestañas, marcos y alertas..

Cuando se produce un fallo, una prueba automatizada puede tomar una captura de pantalla del contenido del navegador para que esta información ayude a la resolución de problemas. Selenium soporta esto a través de su interfaz "Toma Captura de Pantalla".

Algunas acciones están encadenadas para representar una sola acción. Por ejemplo, se puede pasar el cursor por encima de una barra de navegación, que a su vez muestra un submenú hijo y se puede hacer clic en un elemento del

submenú. Estos pueden ser simulados en Selenium usando sus clases de Acción. También se utiliza para simular acciones complejas del teclado.

Normalmente, una aplicación web gestiona su estado así como las preferencias del usuario utilizando cookies. El administrador de Selenium proporciona acceso a las cookies y las acciones relacionadas pueden ser implementadas en pruebas automatizadas.

Selenium también proporciona un navegador integrado sin cabeza `HtmlUnitDriver` para ejecutar código de Selenium. Esta es una característica muy útil que puede ser utilizada por los ingenieros de automatización para ejecutar muchos navegadores en paralelo, con muy poco uso de recursos por navegador.

C pítulo 5 – Elaboraci n de un marco b sico

OA10	Automatizar escenarios de usuario final utilizando buenas pr�cticas de codificaci�n y principios orientados a objetos para colocar c�digo de Selenium en diferentes clases y m�todos. (K3)
OA11	Solucionar problemas y describir el alcance de las mejoras para la implementaci�n de la automatizaci�n que se muestran en ejemplos de c�digos cortos. (K3)

Resumen:

- Ejercicio largo: Automatizaci n de Escenarios Finales
- Refactorizaci n del c digo para crear y utilizar una clase de WebAutomator que incluye la construcci n y las interacciones con WebDriver.
- Pr ximos pasos: (Recorrido de Alto Nivel y Demo)
 - Implementaci n del Auditor de Eventos
 - Patr n de dise o de objeto de p gina
 - Uso de las herramientas de Pagina
 - P ginas como componentes que se pueden cargar
 - Plantilla de Selenium

Antecedentes:

Sin implementar buenas pr cticas de programaci n y patrones de dise o, un c digo automatizado pronto se sobrecarga con muchas instrucciones duplicadas. Las inconsistencias de codificaci n tambi n se introducen en el c digo a medida que los diferentes ingenieros codifican en su propio estilo espec fico.

Es importante poner todas las instalaciones de uso com n en un marco central que se utiliza para escribir las pruebas. Se pueden utilizar los principios de orientaci n a objetos para crear un marco b sico muy f cilmente y continuar mejorando su conjunto de caracter sticas a medida que el proyecto avanza.

Aunque no es el tema directo de este curso, es importante entender que hay muchas medidas que un ingeniero de automatizaci n de pruebas puede tomar para armar un marco de trabajo robusto. Por lo tanto, uno debe conocer el "Page Object Model (POM)"(modelo de objeto de p gina) y componentes como

"WebDriverEventListener", "Page Factories"(herramientas de página), componentes cargables, la plantilla de Selenium etc. que pueden mejorar aún más las características y la robustez de un marco de trabajo.

Referencias:

- Sitio web de SeleniumHQ: <https://www.seleniumhq.org/>
- Todo ha sido diseñado y creado utilizando experiencias de primera mano recogidas de la industria por las organizaciones involucradas en la creación de Selenium United..