

CALL TECNICO ETB

MANUAL GUIA INTALACION Selenium
(GUIA DE INTALACION)

MARZO DE 2025

Tabla de Contenido

1. Introducción

- 1.1 ¿Qué es Selenium?
- 1.2 ¿Por qué usar Selenium con PHP y XAMPP?
- 1.3 Requisitos previos

2. Preparación del Entorno

- 2.1 Instalación y Configuración de XAMPP
- 2.2 Instalación de Google Chrome
- 2.3 Instalación de Composer

3. Configuración de Selenium

- 3.1 Creación del Proyecto PHP
- 3.2 Instalación de la Biblioteca Selenium WebDriver
- 3.3 Descarga e Instalación de ChromeDriver
- 3.4 Configuración de la Variable de Entorno (Opcional)

4. Ejecución de Selenium

- 4.1 Creación del Script de Prueba
- 4.2 Inicio del Servidor Selenium Standalone
- 4.3 Ejecución del Script de Prueba
- 4.4 Interpretación de Resultados

5. Consideraciones Avanzadas

- 5.1 Manejo de Excepciones y Errores
- 5.2 Automatización de Otros Navegadores
- 5.3 Integración con Frameworks de Pruebas (PHPUnit)
- 5.4 Mejores Prácticas de Automatización

6. Resolución de Problemas Comunes

- 6.1 Problemas con ChromeDriver
- 6.2 Errores de Conexión con el Servidor Selenium
- 6.3 Problemas con la Biblioteca WebDriver

7. Recursos Adicionales

8. Conclusión

1. Introducción

1.1 ¿Qué es Selenium?

Selenium es un conjunto de herramientas de código abierto para automatizar la interacción con navegadores web. Permite a los desarrolladores simular acciones de usuario, como hacer clic en botones, rellenar formularios y navegar por páginas web, todo ello a través de código.

1.2 ¿Por qué usar Selenium con PHP y XAMPP?

- **Automatización de Pruebas:** Selenium facilita la creación de pruebas automatizadas para aplicaciones web, lo que mejora la calidad y reduce el tiempo de desarrollo.
- **Web Scraping:** Permite extraer datos de sitios web de manera automatizada.
- **Compatibilidad:** Funciona con múltiples navegadores y sistemas operativos.
- **Integración con XAMPP:** XAMPP proporciona un entorno de desarrollo local completo, lo que facilita la configuración de Selenium.

1.3 Requisitos Previos

- XAMPP instalado y funcionando.
- Google Chrome instalado.
- Conocimientos básicos de PHP.
- Conocimientos básicos de la línea de comandos.

2. Preparación del Entorno

2.1 Instalación y Configuración de XAMPP

- Descarga XAMPP desde el sitio web oficial (apachefriends.org).
- Ejecuta el instalador y sigue las instrucciones.
- Asegúrate de que los servicios Apache y MySQL estén en ejecución.

2.2 Instalación de Google Chrome

- Descarga e instala Google Chrome desde el sitio web oficial (google.com/chrome).
- Mantén Chrome actualizado.

2.3 Instalación de Composer

- Descarga Composer desde getcomposer.org.
- Ejecuta el instalador y sigue las instrucciones.
- Verifica la instalación con `composer --version`.

3. Configuración de Selenium

3.1 Creación del Proyecto PHP

- Crea una carpeta para tu proyecto en `C:\xampp\htdocs`.
- Abre la línea de comandos y navega a la carpeta del proyecto.

3.2 Instalación de la Biblioteca Selenium WebDriver

- Ejecuta `composer require php-webdriver/webdriver`.

3.3 Descarga e Instalación de ChromeDriver

- Verifica la versión de Chrome en `chrome://version/`.
- Descarga ChromeDriver desde `chromedriver.chromium.org`.
- Coloca `chromedriver.exe` en una ubicación accesible (por ejemplo, `C:\xampp`).

3.4 Configuración de la Variable de Entorno (Opcional)

- Agrega la ubicación de `chromedriver.exe` a la variable `PATH` para ejecutarlo desde cualquier directorio.

4. Ejecución de Selenium

4.1 Creación del Script de Prueba

- Crea `test.php` con el código de ejemplo.

4.2 Inicio del Servidor Selenium Standalone

- Descarga el archivo `.jar` de Selenium desde `selenium.dev/downloads`.
- Ejecuta `java -jar selenium-server-<version>.jar`.

4.3 Ejecución del Script de Prueba

- Abre `http://localhost/tu_proyecto/test.php` en el navegador.

4.4 Interpretación de Resultados

- Verifica que el navegador se abre, realiza la búsqueda y se cierra.

5. Consideraciones Avanzadas

5.1 Manejo de Excepciones y Errores

- Utiliza bloques `try-catch` para manejar errores.

5.2 Automatización de Otros Navegadores

- Descarga los controladores correspondientes (GeckoDriver, EdgeDriver).

5.3 Integración con Frameworks de Pruebas (PHPUnit)

- Utiliza PHPUnit para organizar y ejecutar pruebas.

5.4 Mejores Prácticas de Automatización

- Utiliza selectores robustos (ID, CSS).
- Implementa esperas explícitas e implícitas.
- Organiza tu código en funciones y clases.

6. Resolución de Problemas Comunes

- **6.1 Problemas con ChromeDriver**
 - Verifica la versión de ChromeDriver y Chrome.
 - Asegúrate de que ChromeDriver esté en la ubicación correcta.
- **6.2 Errores de Conexión con el Servidor Selenium**
 - Verifica que el servidor esté en ejecución.
 - Comprueba la configuración del host y el puerto.
- **6.3 Problemas con la Biblioteca WebDriver**
 - Asegúrate de que las dependencias estén instaladas.
 - Verifica la versión de la biblioteca.

7. Conclusión

Selenium con PHP y XAMPP es una herramienta poderosa para la automatización web. Con esta guía, puedes configurar tu entorno y comenzar a crear scripts de automatización.

8. Recursos Adicionales

- Documentación de Selenium: selenium.dev/documentation
- Documentación de Composer: getcomposer.org/doc/
- Documentación de PHPUnit: phpunit.de/documentation/

Mejoras Adicionales

- Se han añadido secciones sobre manejo de errores, automatización de otros navegadores, integración con frameworks de pruebas y mejores prácticas.
- Se han añadido las rutas de las páginas oficiales de cada programa.
- Se han añadido puntos de resolución de problemas comunes.

Introducción

Selenium es una herramienta de automatización de código abierto que permite controlar navegadores web a través de programas. En el contexto de PHP y XAMPP, Selenium se utiliza comúnmente para realizar pruebas automatizadas de aplicaciones web, extracción de datos (web scraping) y otras tareas que requieren interacción con un navegador.

Esta guía proporciona un paso a paso detallado para configurar Selenium con PHP y XAMPP en un entorno Windows.

Requisitos Previos

- **XAMPP:**
 - Debes tener XAMPP instalado y funcionando correctamente. Si aún no lo tienes, descarga la versión más reciente desde el sitio web oficial de Apache Friends (apachefriends.org). Asegúrate de instalar la versión de PHP que prefieras.
 - Verifica que los servicios Apache y MySQL estén en ejecución desde el Panel de Control de XAMPP.
- **Google Chrome:**
 - Selenium requiere un navegador para interactuar. Asegúrate de tener Google Chrome instalado.
 - Mantén tu navegador Chrome actualizado.
- **Composer:**
 - Composer es un gestor de dependencias para PHP. Lo utilizaremos para instalar la biblioteca de Selenium.

Paso 1: Instalación de Composer

1. **Descarga de Composer:**
 - Visita el sitio web oficial de Composer (getcomposer.org) y descarga el instalador para Windows (Composer-Setup.exe).
2. **Instalación:**
 - Ejecuta el instalador. Durante la instalación, Composer detectará automáticamente tu instalación de PHP de XAMPP. Asegúrate de que la ruta de PHP sea correcta.
 - Sigue las instrucciones del instalador para completar la instalación.
3. **Verificación:**
 - Abre la línea de comandos (cmd) y ejecuta el siguiente comando para verificar que Composer se instaló correctamente:
 - `composer --version`
 - Si la instalación fue exitosa, verás la versión de Composer en la línea de comandos.

Paso 2: Creación de un Proyecto PHP

1. **Ubicación del Proyecto:**
 - Navega hasta la carpeta `htdocs` de tu instalación de XAMPP. Esta carpeta es la raíz de tu servidor web local. La ruta predeterminada es `C:\xampp\htdocs`.
2. **Creación de la Carpeta:**

- Crea una nueva carpeta dentro de `htdocs` para tu proyecto Selenium. Por ejemplo, puedes llamarla `selenium_php`.
- 3. **Navegación a la Carpeta:**
 - Abre la línea de comandos y navega hasta la carpeta de tu proyecto utilizando el comando `cd`:
 - `cd C:\xampp\htdocs\selenium_php`

Paso 3: Instalación de la Biblioteca de Selenium con Composer

1. **Ejecución del Comando:**
 - Dentro de la carpeta de tu proyecto, ejecuta el siguiente comando para instalar la biblioteca de Selenium WebDriver:
 - `composer require php-webdriver/webdriver`
 - Composer descargará e instalará la biblioteca y sus dependencias en una carpeta llamada `vendor`.

Paso 4: Descarga e Instalación de ChromeDriver

1. **Determinación de la Versión de Chrome:**
 - Abre Google Chrome y ve a `chrome://version/` para verificar la versión de tu navegador.
2. **Descarga de ChromeDriver:**
 - Visita el sitio web de ChromeDriver (chromedriver.chromium.org/downloads) y descarga la versión que coincida con tu versión de Chrome.
 - Extrae el archivo ZIP descargado. Dentro encontrarás el ejecutable `chromedriver.exe`.
3. **Ubicación de ChromeDriver:**
 - Copia el archivo `chromedriver.exe` a una ubicación accesible. Recomendamos copiarlo a la carpeta `xampp` de tu instalación de XAMPP, o agregar la ruta del archivo a la variable de entorno `PATH` de tu sistema.
 - Si optas por agregarlo a la variable `PATH` del sistema, podrás ejecutar el `chromedriver` desde cualquier ruta de la consola de comandos.

Paso 5: Creación de un Script de Prueba

1. **Creación del Archivo:**
 - Dentro de la carpeta de tu proyecto (`selenium_php`), crea un nuevo archivo PHP llamado `test.php`.
2. **Código de Ejemplo:**
 - Abre `test.php` con un editor de texto y agrega el siguiente código de ejemplo:
 -

```
PHP
<?php
require_once('vendor/autoload.php');

use Facebook\WebDriver\Remote\DesiredCapabilities;
```

```

use Facebook\WebDriver\Remote\RemoteWebDriver;
use Facebook\WebDriver\WebDriverBy;

//Ruta al chromedriver.
$host = 'http://localhost:4444/wd/hub';
$capabilities = DesiredCapabilities::chrome();
//En caso de que el chrome driver no se encuentre en el path del sistema, se
debe de colocar la ruta completa del archivo chromedriver.exe
$capabilities->setCapability('binary', 'C:\xampp\chromedriver.exe');

$driver = RemoteWebDriver::create($host, $capabilities);

$driver->get('http://www.google.com');
$element = $driver->findElement(WebDriverBy::name('q'));
$element->sendKeys('Selenium PHP');
$element->submit();

$driver->quit();
?>

```

3. Explicación del Código:

- o `require_once('vendor/autoload.php');` Incluye las bibliotecas de Composer.
- o `$host = 'http://localhost:4444/wd/hub';` define el host del servidor de selenium.
- o `$capabilities = DesiredCapabilities::chrome();` Define las características del navegador que selenium va a utilizar.
- o `$capabilities->setCapability('binary', 'C:\xampp\chromedriver.exe');` Define la ruta del chromedriver, solo es necesario si no se encuentra en el path del sistema.
- o `$driver = RemoteWebDriver::create($host, $capabilities);` Crea una instancia del navegador Chrome.
- o `$driver->get('http://www.google.com');` Abre Google en el navegador.
- o `$element = $driver->findElement(WebDriverBy::name('q'));` Encuentra el elemento de búsqueda.
- o `$element->sendKeys('Selenium PHP');` Escribe "Selenium PHP" en el cuadro de búsqueda.
- o `$element->submit();` Envía el formulario de búsqueda.
- o `$driver->quit();` Cierra el navegador.

Paso 6: Iniciar el Servidor de Selenium

1. Descarga del Servidor:

- o Visita el sitio web oficial de Selenium (selenium.dev/downloads) y descarga la última versión del servidor Selenium Standalone Server (archivo .jar).

2. Ejecución del Servidor:

- o Abre la línea de comandos y navega hasta la ubicación donde guardaste el archivo .jar.
- o Ejecuta el siguiente comando:
 - `java -jar selenium-server-<version>.jar`

- Reemplaza `<version>` con el número de versión del archivo.

Paso 7: Ejecución del Script de Prueba

1. Apertura del Navegador:

- Abre tu navegador web y ve a la siguiente URL:
`http://localhost/selenium_php/test.php.`

2. Resultado:

- Si todo está configurado correctamente, se abrirá una nueva ventana de Chrome, se cargará Google, se introducirá "Selenium PHP" en el cuadro de búsqueda y se realizará la búsqueda.

Consideraciones Adicionales

- Asegúrate de que la versión de ChromeDriver coincida con la versión de Chrome.
- Verifica los registros de errores de XAMPP y del servidor de Selenium si tienes problemas.
- Si deseas trabajar con otros navegadores, deberás descargar los controladores correspondientes (GeckoDriver para Firefox, EdgeDriver para Microsoft Edge, etc.).
- Recuerda que la automatización web debe realizarse de forma ética y responsable, respetando los términos de servicio de los sitios web.

Conclusión

La combinación de Selenium, PHP y XAMPP ofrece un entorno de desarrollo poderoso y versátil para la automatización web. A través de esta guía, hemos explorado los pasos necesarios para configurar este entorno, desde la instalación de los componentes esenciales hasta la ejecución de scripts de prueba.

Selenium permite a los desarrolladores automatizar tareas repetitivas, realizar pruebas exhaustivas y extraer datos de la web de manera eficiente. La integración con PHP y XAMPP facilita la creación de aplicaciones web dinámicas y la automatización de procesos complejos.

Es importante recordar que la automatización web debe realizarse de manera ética y responsable, respetando los términos de servicio de los sitios web y protegiendo la privacidad de los usuarios.

Con los conocimientos adquiridos en esta guía, los desarrolladores pueden aprovechar al máximo el potencial de Selenium para mejorar la calidad de sus aplicaciones web y optimizar sus flujos de trabajo.