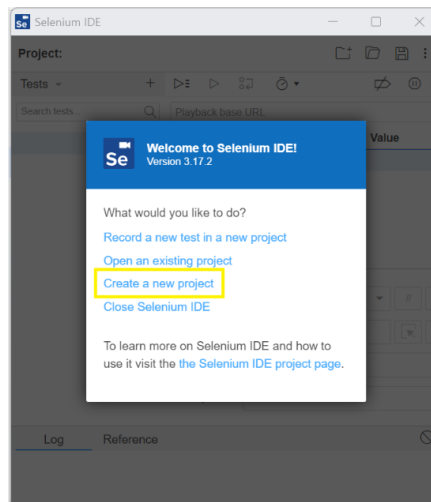


# Cómo utilizar Selenium IDE

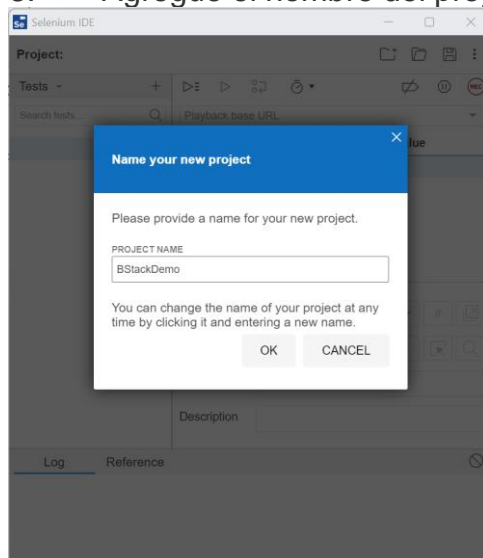
Una vez que tengamos la configuración de Selenium IDE en nuestro respectivo navegador, es decir, Chrome o Firefox, es hora de ver cómo grabar y reproducir scripts en el IDE. Usaríamos la aplicación BStackDemo para probar el proceso de pago.

## Proceso de grabación en Selenium IDE

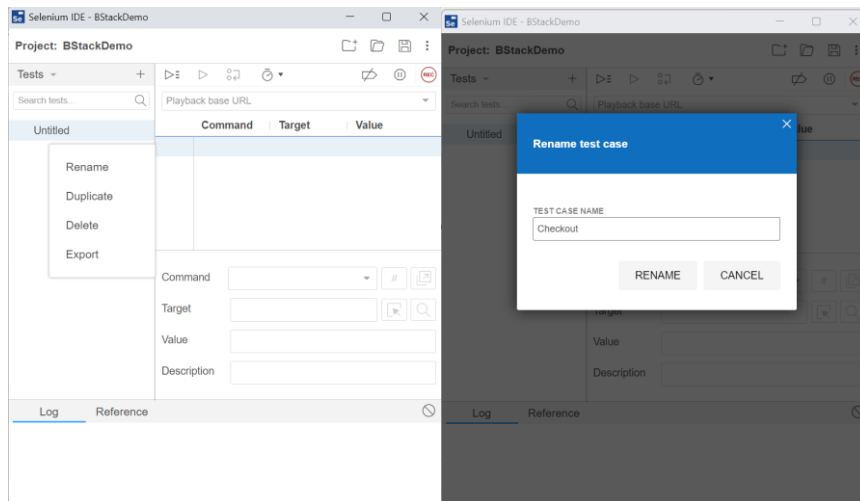
1. Abra el navegador Chrome/Firefox y haga clic en el complemento Selenium presente en la esquina superior derecha del navegador para abrir Selenium IDE.



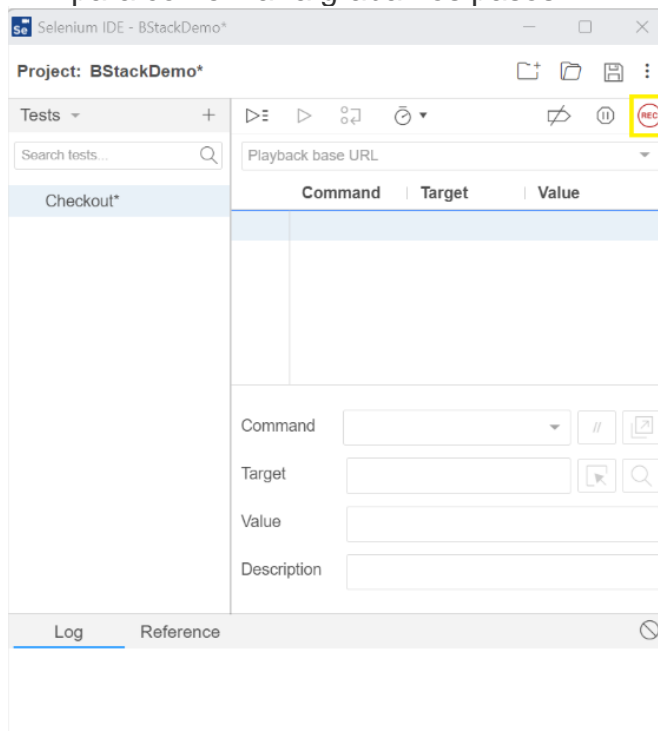
2. Haga clic en Crear un nuevo proyecto.
3. Agregue el nombre del proyecto y haga clic en el botón "Aceptar".



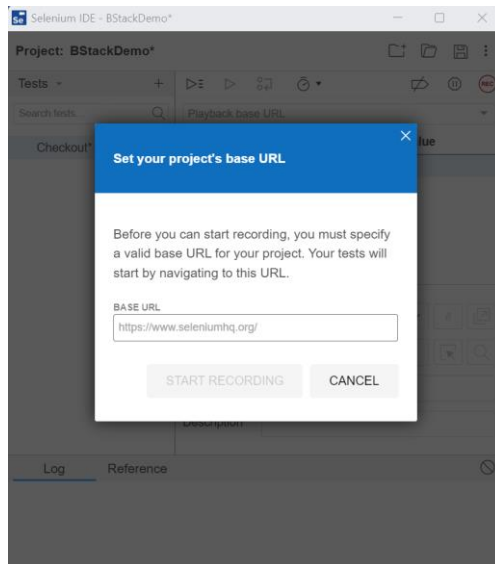
4. Cambie el nombre del caso de prueba sin título a cualquier nombre de caso de prueba apropiado.



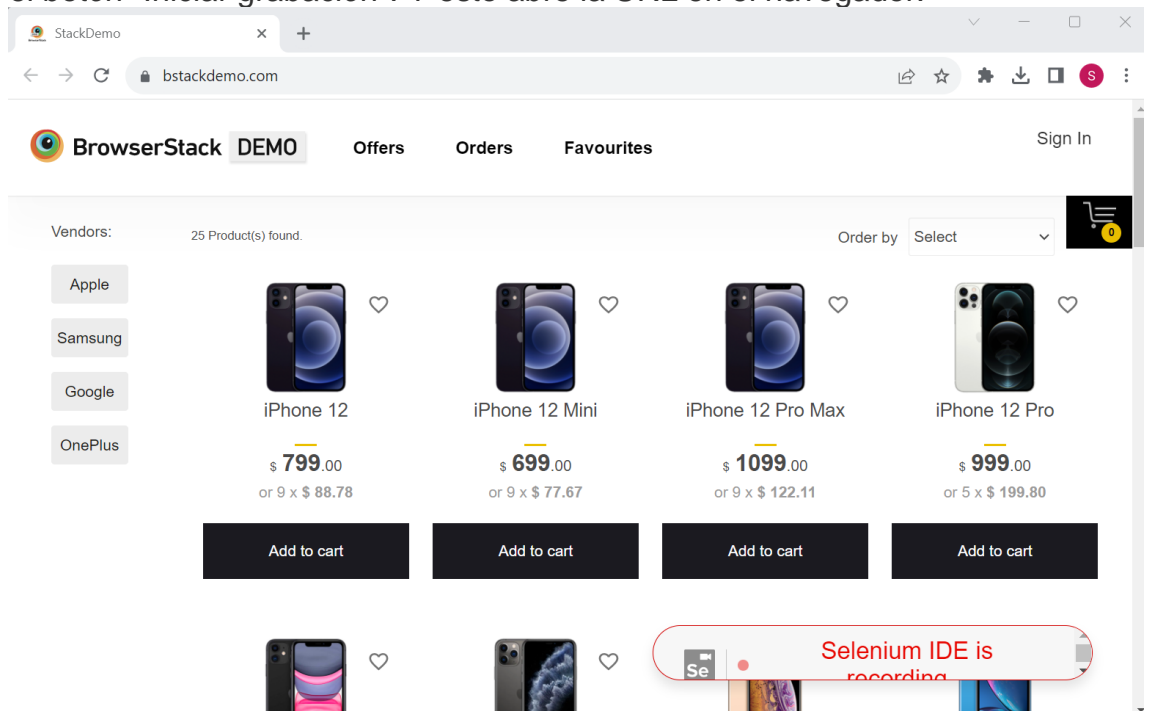
5. Haga clic en el botón Grabar en la esquina superior derecha del IDE para comenzar a grabar los pasos.



Abre la siguiente ventana emergente cuando el proyecto se crea por primera vez.



6. Ingrese la URL base como <https://bstackdemo.com/> y haga clic en el botón "Iniciar grabación". Y esto abre la URL en el navegador.



7. En la página BStackDemo, realice los siguientes pasos para grabar:

- Filtrar Ordenar por "De menor a mayor".
- Haga clic en el modelo de Samsung.
- Haga clic en "Agregar al carrito" para el primer artículo.
- Haga clic en el botón "Pagar".

8. Vaya a IDE y haga clic en el botón Grabar para detener la grabación.

The screenshot displays the Selenium IDE interface for a project named 'BStackDemo\*'. The 'Tests' tab is active, showing a test suite named 'Checkout\*'. The URL bar is set to 'https://bstackdemo.com/'. The test suite contains four steps:

|   | Command         | Target                  | Value                   |
|---|-----------------|-------------------------|-------------------------|
| 1 | open            | https://bstackdemo.com/ |                         |
| 2 | set window size | 1054x656                |                         |
| 3 | click           | css=select              |                         |
| 4 | select          | css=select              | label=Lowest to highest |

Below the steps, there are input fields for Command, Target, Value, and Description, each with a corresponding icon for selection or search.

The 'Log' tab at the bottom shows the execution results:

- 3. click on css=select OK 15:25:37
- 4. select on css=select with value label=Lowest to highest OK 15:25:40
- 5. click on css=.filters-available-size:nth-child(3) .checkmark OK 15:25:43
- 6. click on css=#\31 4 > .shelf-item\_\_buy-btn OK 15:25:46
- 7. click on css=.buy-btn OK 15:25:49
- 'Checkout' completed successfully 15:25:50

9. Todos los pasos se capturan y ejecutan una vez que se detiene la grabación. Guarde el proyecto en el sistema haciendo clic en el botón Guardar en la esquina superior derecha del IDE. (Se guarda en formato .side)

## Proceso de reproducción en Selenium IDE

1. Haga clic en el botón "Ejecutar prueba actual" en Selenium IDE

Selenium IDE - BStackDemo

Project: BStackDemo

Executing ▾

Run current test Ctrl+R ▾

|   | Command | Target  | Value |
|---|---------|---|-------|
|   |         |   | hest  |
| 5 | ✓ click | css=.filters-available-size:nth-child(3) .checkmark |       |
| 6 | ✓ click | css=#31 4 > .shelf-item__buy-btn                    |       |
| 7 | ✓ click | css=.buy-btn  |       |

Command ▾ // [icon]

Target [input] [icon] [icon]

Value [input]

Description [input]

Runs: 1 Failures: 0

Log Reference

|   |    |          |
|---|----|----------|
| 3. click on css=select  | OK | 15:39:25 |
| 4. select on css=select with value label=Lowest to highest      | OK | 15:39:26 |
| 5. click on css=.filters-available-size:nth-child(3) .checkmark | OK | 15:39:28 |
| 6. click on css=#31 4 > .shelf-item__buy-btn                    | OK | 15:39:30 |
| 7. click on css=.buy-btn  | OK | 15:39:32 |
| 'Checkout' completed successfully                               |    | 15:39:32 |

2. Ejecuta los mismos pasos registrados anteriormente y da un resultado Aprobado si todos los pasos se realizan correctamente. En caso de que no pueda interactuar con algún elemento durante la reproducción, la ejecución falla.

3. Se capturan registros para cada paso junto con el tiempo necesario para ejecutar cada paso.

Runs: 1 Failures: 0

Description [input]

Log Reference

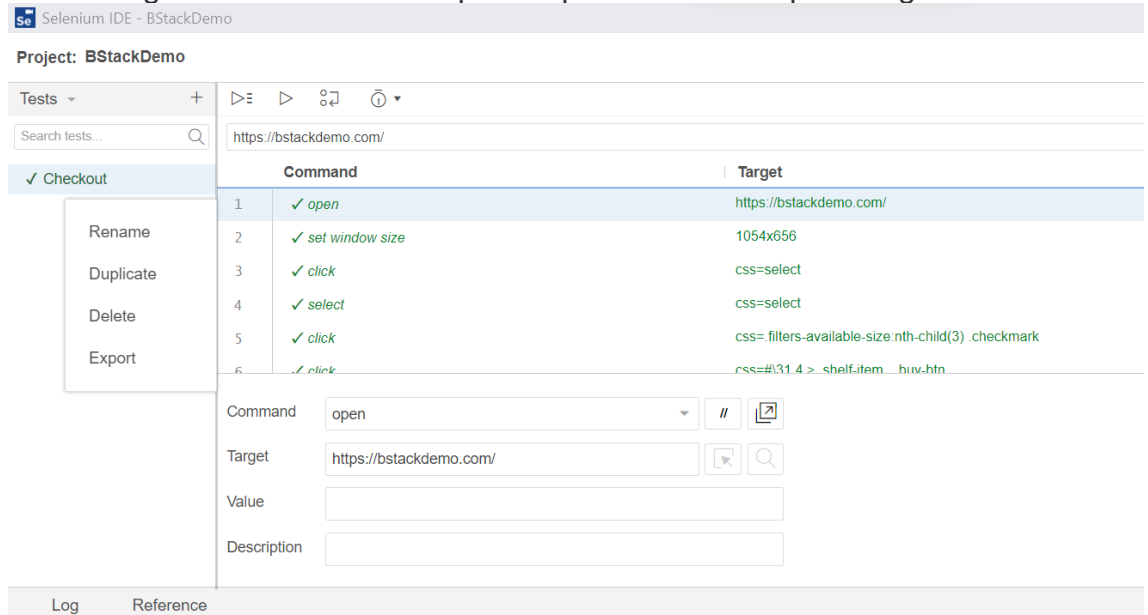
Running 'Checkout'

|   |    |          |
|---|----|----------|
| 1. open on https://bstackdemo.com/                              | OK | 15:39:21 |
| 2. setWindowSize on 1054x656                                    | OK | 15:39:21 |
| 3. click on css=select  | OK | 15:39:23 |
| 4. select on css=select with value label=Lowest to highest      | OK | 15:39:25 |
| 5. click on css=.filters-available-size:nth-child(3) .checkmark | OK | 15:39:26 |
| 6. click on css=#31 4 > .shelf-item__buy-btn                    | OK | 15:39:28 |
| 7. click on css=.buy-btn  | OK | 15:39:30 |
| 'Checkout' completed successfully                               |    | 15:39:32 |

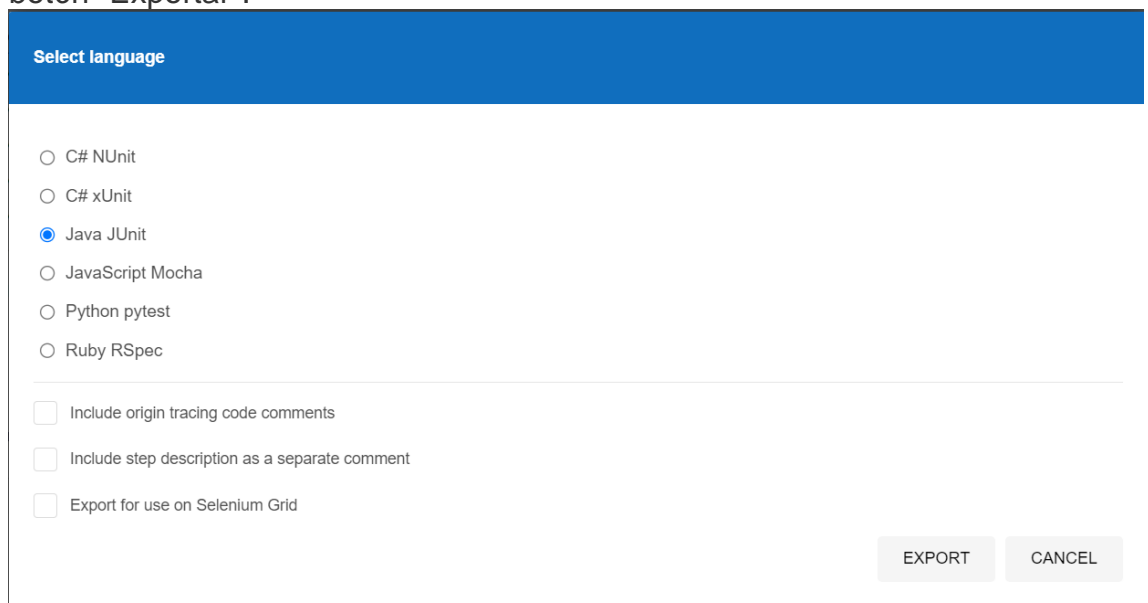
# Exporte el TestCase usando Selenium IDE

Testcase se puede exportar en diferentes lenguajes de programación desde Selenium IDE siguiendo los pasos a continuación:

1. Haga clic en el botón "Exportar" para el caso de prueba grabado.



2. Esto abre una ventana emergente para Seleccionar idioma. Elija cualquier idioma preferido. (por ejemplo: Java JUnit) y haga clic en el botón "Exportar".



3. Guarde el archivo en el sistema local y luego ábralo usando el Bloc de notas o cualquier editor compatible para verificar el script de

prueba.

A screenshot of the Selenium IDE interface showing a test script named 'CheckoutTest'. The script is written in Java and uses Selenium WebDriver to interact with a web application. It includes imports for java.net.URL, WebDriver, Map, JavaScriptExecutor, and WebElement. The script has a setUp() method to initialize the driver and vars, a tearDown() method to quit the driver, and a checkout() test method that performs several actions: getting the URL, setting window size, clicking a select element, clicking a dropdown option, clicking a checkmark, and clicking buy buttons. The interface shows a menu with File, Edit, and View options.

```
import java.net.URL;
public class CheckoutTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void checkout() {
        driver.get("https://bstackdemo.com/");
        driver.manage().window().setSize(new Dimension(1054, 656));
        driver.findElement(By.cssSelector("select")).click();
        {
            WebElement dropdown = driver.findElement(By.cssSelector("select"));
            dropdown.findElement(By.xpath("//option[. = 'Lowest to highest']")).click();
        }
        driver.findElement(By.cssSelector(".filters-available-size:nth-child(3) .checkmark")).click();
        driver.findElement(By.cssSelector("#\\31 4 > .shelf-item__buy-btn")).click();
        driver.findElement(By.cssSelector(".buy-btn")).click();
    }
}
```

## Resumen

Selenium IDE es una de las herramientas de grabación y reproducción más accesibles de Selenium Tool Suite, que no requiere ninguna configuración particular. Selenium IDE tiene complementos para los navegadores Firefox y Chrome. Selenium IDE viene con un amplio conjunto de comandos impulsados por Selenese y le permite registrar y probar diferentes interacciones de una aplicación web con el navegador. También existen limitaciones que deben tenerse en cuenta antes de elegir Selenium IDE.