

## Guía Completa del Proyecto Selenium Grid Demo

Alumno: Luis Orlando Chicas Campos

Este documento describe paso a paso cómo se desarrolló el proyecto selenium-grid-demo para el reto 2, incluye

## 1. Creación de la estructura del proyecto en VSCode

Se creó la siguiente estructura de carpetas y archivos:

```
selenium-grid-demo/  
... config/  
· ... webdriver-simple-config.js  
... tests/  
· ... reliable-chrome-test.js  
... docker-compose.yml  
... package.json
```

Comandos utilizados en CMD:

```
mkdir selenium-grid-demo  
cd selenium-grid-demo  
mkdir config tests  
touch config/webdriver-simple-config.js  
touch tests/reliable-chrome-test.js  
touch docker-compose.yml package.json
```

## 2. Configuración del archivo webdriver-simple-config.js

Este archivo contiene la configuración del WebDriver para Chrome usando Selenium Grid.

Código:

```
const { Builder } = require('selenium-webdriver');
```

```
async function createDriver() {  
  return await new Builder()  
    .usingServer('http://localhost:4444/wd/hub')  
    .forBrowser('chrome')  
    .build();  
}
```

```
module.exports = createDriver;
```

### 3. Desarrollo del caso de prueba reliable-chrome-test.js

Este archivo contiene el caso de prueba para realizar login exitoso en <https://www.saucedemo.com>.

Código:

```
const createDriver = require('../config/webdriver-simple-config');
const { By } = require('selenium-webdriver');

(async function loginTest() {
  const driver = await createDriver();

  try {
    await driver.get('https://www.saucedemo.com/');
    await driver.findElement(By.id('user-name')).sendKeys('standard_user');
    await driver.findElement(By.id('password')).sendKeys('secret_sauce');
    await driver.findElement(By.id('login-button')).click();
    await driver.sleep(3000);
    const currentUrl = await driver.getCurrentUrl();
    console.log('Login exitoso, URL actual:', currentUrl);
  } catch (error) {
    console.error('Error en la prueba:', error);
  } finally {
    await driver.quit();
  }
})();
```

#### 4. Configuración de docker-compose.yml

Este archivo configura el Selenium Grid con el hub y un nodo Chrome.

Código:

version: '3'

services:

selenium-hub:

image: selenium/hub:4.14.0

container\_name: selenium-hub

ports:

- "4444:4444"

chrome:

image: selenium/node-chrome:4.14.0

volumes:

- /dev/shm:/dev/shm

depends\_on:

- selenium-hub

environment:

- SE\_EVENT\_BUS\_HOST=selenium-hub

- SE\_EVENT\_BUS\_PUBLISH\_PORT=4442

- SE\_EVENT\_BUS\_SUBSCRIBE\_PORT=4443

## 5. Creación de package.json

Este archivo define las dependencias y scripts para ejecutar la prueba.

Código:

```
{
  "name": "selenium-grid-demo",
  "version": "1.0.0",
  "description": "Demo Selenium Grid con Chrome",
  "scripts": {
    "test": "node tests/reliable-chrome-test.js"
  },
  "dependencies": {
    "selenium-webdriver": "^4.14.0"
  }
}
```

## 6. Instalación de dependencias

Comando utilizado en CMD para instalar las dependencias del proyecto:

```
npm install
```

## 7. Comandos en CMD para levantar el Grid y ejecutar pruebas

Para levantar el Selenium Grid:

```
docker-compose up
```

Para ejecutar la prueba:

```
npm test
```

Para verificar la UI del Grid en el navegador:

```
http://localhost:4444/ui
```