**PlayWright vs Puppeteer:**

Microsoft ha anunciado una nueva herramienta que automatiza las acciones del navegador con nodeJS. Los navegadores compatibles con esta biblioteca son: chromium, WebKit, Firefox y Edge (que se basa en el chromium). Vale la pena mencionar que **playwright** no es un marco de trabajo de Automatización de pruebas en sí mismo, pero puede integrarse con las bibliotecas de prueba (chai, mocha, jazmín, jest, cucumber) y convertirse en un marco de trabajo de automatización de pruebas.

**PlayWright** posee una biblioteca similar que tiene como objetivo hacer lo mismo que **puppeteer** de Google, pero la principal diferencia es que **puppeteer** solo admite chromium, mientras que **playwright** admite varios navegadores diferentes.

Navegar por la API no es muy sencillo porque no es fácil encontrar las funciones que estamos buscando (parte de esto es porque no hay muchos ejemplos en línea).

El objetivo principal con **playwright** es mejorar las pruebas automatizadas de la interfaz de usuario eliminando la fragilidad, mejorando la velocidad de ejecución y ofreciendo información sobre el funcionamiento del navegador.

Al crear **playwright** la idea inicial fue tomar como base lo que hace Puppeteer e incorporar lo que este hace en la API, por ejemplo, la emulación de dispositivo / agente de usuario, el se configura de manera coherente en el nivel de .**BrowserContext** para permitir escenarios de varias páginas.

Por otra parte, **playwright** también apunta a ser nativo dentro de la nube. En lugar de una sola página, la abstracción de BrowserContext ahora es central para la operación de la biblioteca.

Las librerías de **playwright** no son muy maduras y, con suerte, Microsoft trabajará muy duro para mejorarlas. Si se tuviera que desarrollar un script de pruebas web muy simple, **playwright** es una excelente opción. Pero, considerando los escenarios realmente complejos que tenemos en nuestro trabajo considerando el control de calidad, definitivamente es mejor elegir otros marcos de trabajo en lugar de **playwright.**

Aquí se presenta una tabla comparando las diferencias entre las tres herramientas utilizadas en el taller:

|  | **Cypress** | **Protractor** | **Puppeteer** | **PlayWright** |
| --- | --- | --- | --- | --- |
| Interfaz gráfica | Sí | No | No | **No** |
| Generador de configuración | Sí | Sí (Con angular CLI) | No | **No** |
| Locators para AngularJS | No | Sí | No | **No** |
| Integración con Test frameworks | No, interno (sintaxis jasmine) | Jasmine, Mocha, Cucumber | Principalmente todos | **Principalmente todos** |
| Sistemas operativos | Mac, Linux, Windows | Mac, Linux, Windows | Mac, Linux, Windows | Mac, Linux, Windows |
| Grabación de pruebas | Sí | No | No | **No** |
| Selenium backend | No | Sí | No | **No** |
| Time-traveling | Sí | No | No | **No** |