**Automatización de una Mini Aplicación E-commerce - E2E con Playwright**

# Descripción del Proyecto

Este proyecto tiene como objetivo automatizar los flujos End-to-End (E2E) de la aplicación SauceDemo utilizando Playwright. Se aplican buenas prácticas de automatización, incluyendo el uso de Page Object Model (POM), fixtures, hooks y generación de reportes.

# ¿Qué es Playwright?

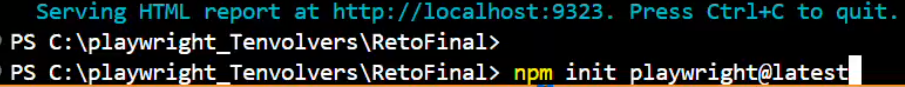
Playwright es un framework de automatización de pruebas desarrollado por Microsoft que permite la prueba de aplicaciones web en múltiples navegadores como Chrome, Firefox y Safari. Soporta pruebas en paralelo, ejecución en modo headless y generación de trazas para depuración.

# Configuración del Proyecto

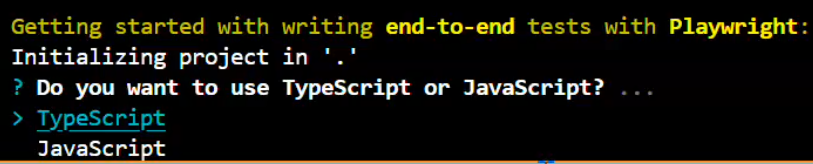
## Primeros pasos, instalación de dependencias

Para ejecutar este proyecto es necesario que luego de descargar la carpeta que lo contiene y abrirla con la ayuda de Visual Studio Code, habras la consola y ejecutes los siguientes comandos para poder instalar las dependencias necesarias para poder ejecutar los test que contiene, para ello el primer comando a ejecutar es:

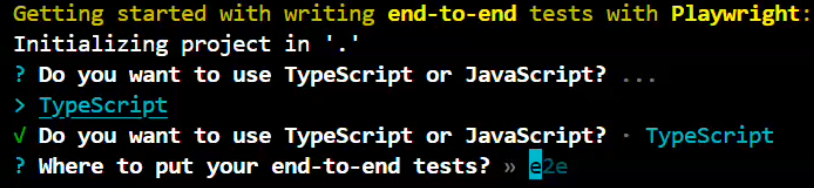
1. npm init playwright@latest



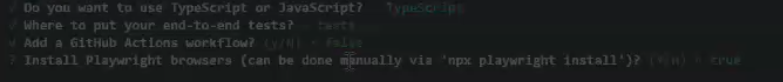
1. Seleccionar TypeScript ya que esta desarrollado en este lenguaje.



1. A continuación, en la pregunta escribes “test”, porque así se llama la carpeta donde están los test a ejecutar, puedes verlo en la estructura del proyecto.



1. Luego te pedirá que, si quieres agregarlo a github, le escribes “false” y presionas “enter”.



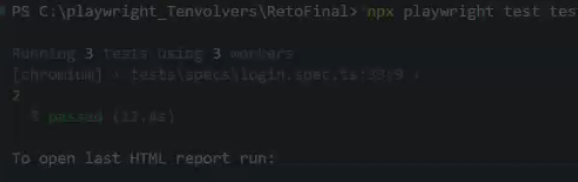
1. Luego te pedirá que, si quieres instalar los navegadores, así que le escribes “true” y presión as “enter”.



1. Luego saldrá que si deseo sobre Escribir la carpeta le decimos que no y presionamos “enter”.



1. Se descargan e instalan los paquetes y queda listo para poder correr los tests.
2. Corro los test.



## Uso

Para ejecutar todas las pruebas:

***npx playwright test***

Para ejecutar una prueba específica, por ejemplo, login:

***npx playwright test tests/specs/login.spec.ts***

# Características

- Automatización E2E  
- Validaciones dinámicas  
- Uso de fixtures personalizados  
- Hooks beforeEach y afterEach  
- Soporte para múltiples navegadores (Chromium, Firefox)  
- Capturas automáticas al fallar  
- Generación de reportes HTML

# Estructura del Proyecto

RETOFINAL/  
├── node\_modules/  
├── playwright-report/  
│ └── index.html  
├── test-results/  
├── tests/  
│ ├── fixture/  
│ │ └── loginFixture.ts  
│ ├── pages/  
│ │ ├── CarritoPages.ts  
│ │ ├── CheckoutStepOne.ts  
│ │ ├── CheckoutStepTwo.ts  
│ │ ├── CompraExitosPages.ts  
│ │ ├── LoginPages.ts  
│ │ ├── LoginValidacionRestricciones.ts  
│ │ └── ProductosPages.ts  
│ └── specs/  
│ ├── escenariosAlternosE2E1.spec.ts  
│ ├── inicioDeSesion.spec.ts  
│ ├── login.spec.ts  
│ └── LoginValidaciónDeRestriccionesDeUsuario.spec.ts  
├── tests-examples/  
│ └── demo-todo-app.spec.ts  
├── .env  
├── .gitignore  
├── package-lock.json  
├── package.json  
├── playwright.config.ts  
└── storageState.json

# Reportes HTML

Se genera un reporte en playwright-report/ accesible en el navegador.

# Autor

Ing. Elber Alexander Ponguta