# Guía para realizar una automatización y la creación del entorno

En esta documentación se recogen todos los pasos a realizar para hacer una automatización de una página web, además de la preparación del entorno para poder realizarla de manera correcta como también la realización de documentos como el archivo .env para facilitarnos la ejecución del test de una manera cómoda, correcta y fácil de entender.

***Preparación del entorno:***

Lo primero que hay que hacer es instalar Node.js ya que era esencial para poder usar Playwright, una herramienta de automatización y pruebas end‑to‑end para aplicaciones web, desarrollada por Microsoft y de código abierto y que se ejecutasen las cosas sin ningún tipo de problema. La instalación se debe realizar desde la página oficial:

* <https://nodejs.org>

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Después de instalar correctamente node.js hay que crear la carpeta del proyecto y posteriormente abrir el editor de código que usaremos, en este caso VSCode.

En la consola de comandos dentro de la carpeta creada para nuestro proyecto ejecutamos el siguiente comando para comenzar nuestro proyecto:

* npm create playwright@latest

Este comando genera automáticamente todo el entorno necesario para realizar tests con Playwright. En concreto:

* Crea el package.json (en el caso de que no existiera ya)
* Instala Playwright y sus dependencias
* Descarga los navegadores usados para los test: Chromium, Firefox y WebKit
* Crea una estructura de carpetas con ejemplos
* Configura el archivo playwright.config y scripts útiles

Con este único comando, el entorno queda listo para empezar a crear y ejecutar pruebas automatizadas.

Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

***Prueba y creación del test:***

Una vez con el proyecto creada correctamente ejecutaremos nuestro primer test, en este caso el test de ejemplo que se crea en nuestro proyecto con el siguiente comando:

* npx playwright test

Texto

El contenido generado por IA puede ser incorrecto.

Con esto corroboramos que nuestro proyecto funciona correctamente y sin ningún tipo de problema, por lo que procedemos a la creación de nuestro primer test, en este caso automatizaremos el login de la página Elysium:

* <https://elyp.intechsol-pr.net>

En la terminal del proyecto ejecutamos el siguiente comando:

* npx playwright codegen <URL\_de\_la\_web>

En nuestro caso:

* npx playwright codegen https://elyp.intechsol-pr.net

Este comando abrira Playwright Codegen que es una herramienta que graba las acciones que haces en el navegador y genera el código base para el test.

IMPORTANTE: UNA VEZ EJECUTADO EL COMANDO ANTERIOR LA NUEVA PESTAÑA QUE SE ABRA GRABARA TODAS LAS ACCIONES HECHAS POR EL USUARIO EN DICHA PAGINA POR LO QUE HAY QUE TENER ESPECIAL ATENCION CON LO QUE SE QUIERE GRABAR Y REALIZAR LAS COSAS CORRECTAMENTE PARA EVITAR CODIGO CRUZADO Y ERRONEO.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Una vez ejecutado el código se abrirá DOS NUEVAS VENTANAS en nuestro ordenador, por un lado, nos aparecerá la ventana de crohmium la cual será la responsable de captar todo lo que haga el usuario y será transcrito en código en la otra ventana abierta la cual se trata de Playwright Inspector.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Las primeras líneas de código en aparecer son siempre las importaciones y las funciones correspondientes para realizar las pruebas sin problemas y la primera línea de código de la automatización como tal, la cual se encarga de abrir la url que hayamos puesto.

Continuamos el test poniendo el usuario y la contraseña, en mi caso:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Y comprobamos que se realiza la copia de nuestras acciones en código:

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Se copia automáticamente nuestras acciones en código, en este caso guarda las credenciales con las que hemos rellenado los apartados de username y password. El resultado final de nuestra automatización es este:

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Lo copiamos y lo pegamos en un documento nuevo que crearemos dentro de la carpeta tests.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Debido a que el test contiene datos sensibles es necesario el uso de un archivo .env para mejorar el test.

***Archivo .env:***

Lo primero que haremos será crear un archivo .env el cual permite separar las credenciales sensibles del código fuente. Esto mejora la seguridad, evitando exponer usuarios y contraseñas en el repositorio. También facilita el mantenimiento al poder cambiar datos sin modificar el script. Además, permite reutilizar las mismas variables en distintos entornos (desarrollo, pruebas, producción).

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Creamos nuestras variables de entorno en este caso: BASE\_URL, USER, PASSWORD.

Y ahora lo que tenemos que hacer es ejecutar el siguiente comando para hacer funcional nuestro archivo .env

* npm install dotenv

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Este comando lo que hace es instalar las librerías correspondientes y permite que el archivo .env pueda ser leído.

Después añadimos en nuestro test lo siguiente:

* import dotenv from 'dotenv';
* dotenv.config();

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Estas dos líneas harán que se importen la librería que hemos instalado anteriormente para que el archivo .env pueda ser leído y además permiten extraer la información necesaria de ese archivo.

Una vez hecho todo esto comenzamos a corregir nuestro código para que no contenga datos sensibles.

Texto

El contenido generado por IA puede ser incorrecto.

Convertimos nuestras variables de entorno en constantes para trabajar de manera más sencilla.

Texto

El contenido generado por IA puede ser incorrecto.

Arreglamos todo nuestro código y además añadimos en este caso un catch para detectar posibles errores detectados y una verificación sencilla en este caso una captura de pantalla para corroborar que esta todo correcto.

El resultado final es el siguiente:

import { test, expect } from '@playwright/test';

import dotenv from 'dotenv';

dotenv.config();

if (!process.env.BASE\_URL || !process.env.USER || !process.env.PASSWORD) {

  throw new Error('Faltan variables de entorno: BASE\_URL, USER o PASSWORD');

}

test('test', async ({ page }) => {

const baseUrl = process.env.BASE\_URL!;

const username = process.env.USER!;

const password = process.env.PASSWORD!;

try {

  await page.goto(`${baseUrl}/Login/Index`);

  await page.getByRole('textbox', { name: 'Username' }).click();

  await page.getByRole('textbox', { name: 'Username' }).fill(username);

  await page.getByRole('textbox', { name: 'Password' }).click();

  await page.getByRole('textbox', { name: 'Password' }).fill(password);

  await page.getByRole('button', { name: 'SIGN IN' }).click();

    // Espera a que la web cambie de pantalla

  await page.waitForNavigation();

  // Toma una captura de pantalla de la página después de hacer login y la guarda

  await page.screenshot({ path: 'elysium\_logged\_in.png' });

        await page.waitForTimeout(2000);

    } catch (error) {

      console.error('La prueba falló:', error);

      throw error;

    }

});

A continuación, verificamos nuestro test:

* npx playwright test login.spec.ts

Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

El resultado es exitoso y la captura se realizó correctamente.

Verificamos también el test:

Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Todo se ha realizado correctamente, nuestro test a sacado correctamente la información del archivo .env ha conseguido iniciar sesión correctamente y nos ha verificado que el test es correcto no solo con la confirmación de la consola de que el test ha pasado, sino que además a hechos todo correctamente entre esas cosas la captura de pantalla de la pantalla principal tras el inicio confirmándonos el acceso correcto al entorno de ELYSIUM.