



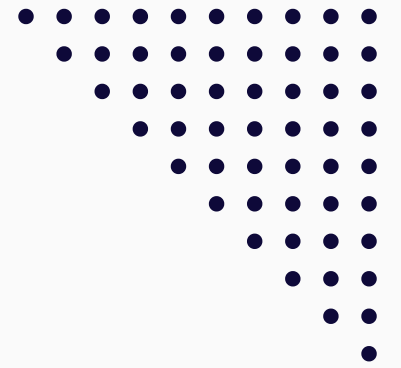
# TATS

Testing Automation Tools Selections  
SoftwareTesting

Presentado Team Playwright:  
Teresa Hernández  
Jorge Sifuentes  
Adalid Guzmán



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH



# Agenda

- 1.Introducción
- 2.Pugh Matrix
- 3.Proof of Concept
- 4.Fortalezas y Debilidades
- 5.Conclusiones





# ¿Dónde estamos?



Nuestra organización ha utilizado Selenium como herramienta principal para la automatización de pruebas. Sin embargo, en la actualidad, hemos enfrentado desafíos relacionados con la escalabilidad, la velocidad de ejecución y la compatibilidad con distintos navegadores.

# Apunta alto

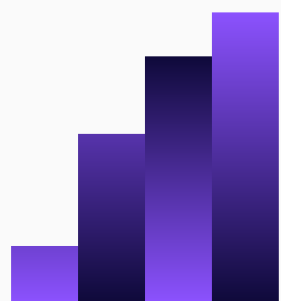
Ante esta situación, hemos decidido explorar alternativas más modernas, como son *Playwright* y *Cypress*.

Para ello se realizó una evaluación empleando la *matriz de Pugh*.

El propósito de esta evaluación, es determinar qué herramienta se adapta mejor a nuestras necesidades actuales y futuras, mediante el uso de esta, que ofrece un enfoque estructurado para evaluar diferentes alternativas basadas en un conjunto de criterios predefinidos

# Pugh Matrix (Resumida)

Topic	Criteria	Peso	Alternatives scoring			Scoring comments	
			Selenium as Baseline	Cypress	Playwright	Cypress	Playwright
	Cross-browser testing	3	S	-	+	Soporta solo Chrome, Edge y Firefox; no Safari ni IE.	Soporta múltiples navegadores modernos, incluyendo Safari.
	Mobile testing friendly (simulation)	1	S	-	+	Limitado a simulaciones básicas de dispositivos móviles.	Proporciona simulación móvil robusta y multiplataforma.
	Speed execution	3	S	+	+	Ejecuta pruebas más rápido debido a su arquitectura en el navegador.	Alta velocidad gracias a su arquitectura moderna.
	Integration with Pipeline	1	S	+	+	Compatible con la mayoría de pipelines CI/CD (Jenkins, GitHub).	Compatible y fácil de integrar con CI/CD.
	Network mocking and interception	1	S	+	+	Ofrece capacidades sólidas de mocking de red.	Soporte avanzado y flexible para interceptar redes.
	Madurez	1	S	-	S	Menos maduro que Selenium en términos de tiempo en el mercado.	Más reciente, pero con características modernas.
TOTAL +			0	3	5		
TOTAL -			0	3	0		
Net value			0	0	5		
Weighted Total +			0	5	9		
Weighted Total -			0	2	0		
Final Weighted Value			0	3	9		
Ranking			3	2	1		



# Resultados



Los resultados de la matriz indican que la herramienta más adecuada en función de nuestros criterios y sus pesos, es Playwright.

Playwright obtiene una puntuación de 30, contra los 9 puntos que obtiene Cypress.

No se considero la ponderación, por la evidente ventaja de Playwright.



# Proof of Concept

Se realiza una prueba de concepto para validar que la herramienta cumple nuestras necesidades y criterios establecidos.

Para ello, se automatizan 2 test cases en donde evalúa diferentes funciones en una aplicación web.



## Playwright

Herramienta de automation.



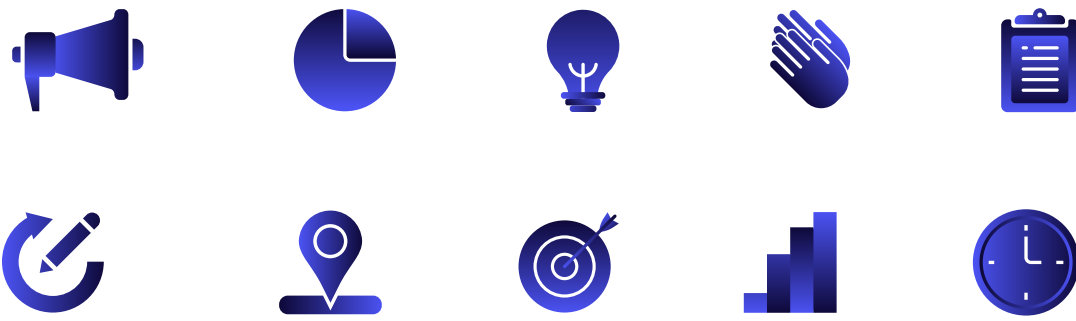
## TypeScript

Lenguaje de programación



## Visual Studio Code

IDE (Entorno de desarrollo empleado)

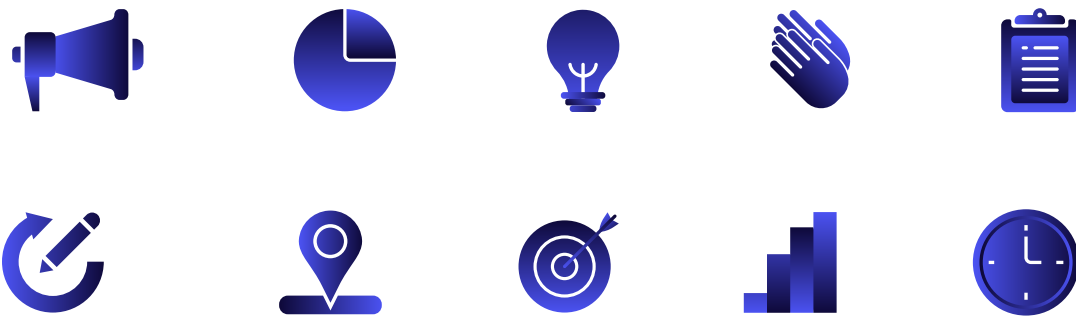


# Test Case 1

1. Go [todemo.oscommerce.com/watch](https://todemo.oscommerce.com/watch).
2. Click “Royal London 41003-03” product.
3. Add to Cart Update quantity to 2.
4. Assert that quantity is 2.
5. Checkout .
6. Sign in (\*).
7. Select “Cash on delivery” payment method and confirm conditions.
8. Complete the payment.
9. Check that the message “Your order has been processed” is shown.

```
TS tc1.spec.ts x
tests > TS tc1.spec.ts > ...
1  import { test, expect } from "@playwright/test";
2
3  test("Test Case 1", async ({ page }) => {
4    await page.goto('https://sqademosatp.net/watch/');
5    await page.locator('#box-1857').getByText('Royal London 41003-').click();
6    await page.waitForTimeout(1000);
7    await page.locator('#btn-cart').getByRole('button', { name: 'Add to Basket' }).click();
8    await page.locator('.qty > .qty-box > .bigger').click();
9    await page.waitForTimeout(1000);
10   await page.locator('.qty > .qty-box > .bigger').click();
11   await page.waitForTimeout(1000);
12   await expect(page.locator('input[name="cart_quantity\\\\[\\\\]"') ).toHaveValue('3');
13   await page.locator('.pop-up-close').click();
14   await page.locator('#cart-box').hover();
15   await page.locator('#cart-box').getByRole('link', { name: 'Checkout' }).click();
16   await page.locator('#login-email_address').click();
17   await page.locator('#login-email_address').fill('adalid.guzman@hotmail.com');
18   await page.locator('#login-password').click();
19   await page.locator('#login-password').fill('Contraseña12');
20   await page.getByRole('button', { name: 'Log in' }).click();
21   await page.waitForTimeout(1000);
22   await page.getByRole('radio', { name: '£102.06 Total' }).check();
23   await page.waitForTimeout(1000);
24   await page.getByRole('radio', { name: 'Cash on Delivery' }).check();
25   await page.waitForTimeout(1000);
26   await page.getByRole('button', { name: 'Confirm and pay' }).click();
27   await expect(page.locator('#box-36757')).toContainText('We\\'ll send you order status update by email');
28 });
29
```

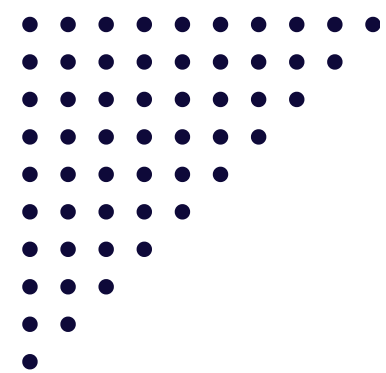




# Test Case 2

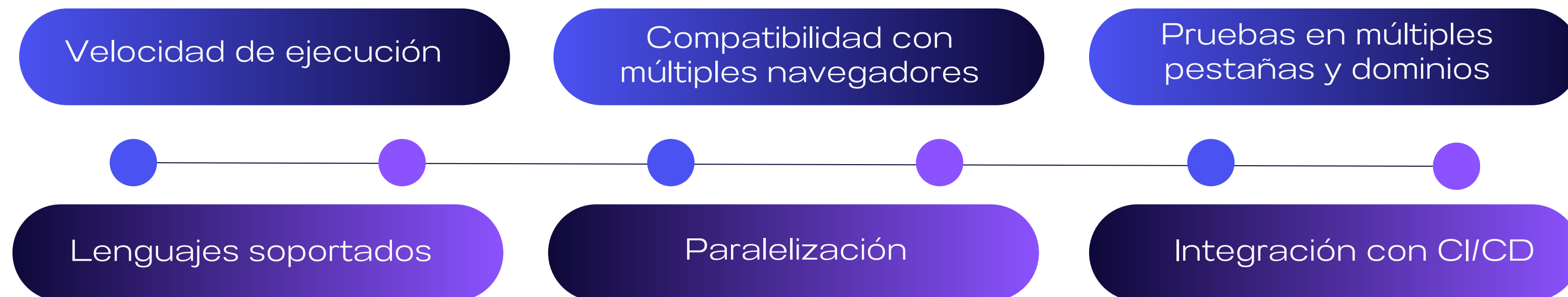
1. Go [todemo.oscommerce.com/watch](https://todemo.oscommerce.com/watch).
2. Click “Citizen Eco-Drive Silver Tone Men” product.
3. Add to Cart Update quantity to 2.
4. Assert that quantity is 3.
5. Checkout .
6. Sign in (\*).
7. Select “Cash on delivery” payment method and confirm conditions.
8. Complete the payment.
9. Check that the message “Your order has been processed” is shown.

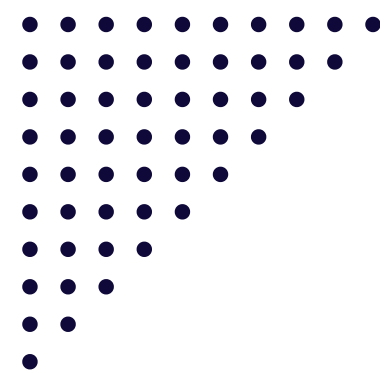
```
TS tc2.spec.ts x
tests > TS tc2.spec.ts > test("Test Case 2") callback
1  import { test, expect } from "@playwright/test";
2
3  test("Test Case 2", async ({ page }) => {
4    await page.goto("https://sqademosatp.net/watch/");
5    await page.getByLabel("Next").click();
6    await page.getByRole("link", { name: "Citizen Eco-Drive Silver Tone" }).first().click();
7    await page.waitForTimeout(1000);
8    await page.locator("#btn-cart").getByRole("button", { name: "Add to Basket" }).click();
9    await page.waitForTimeout(1000);
10   await page.locator(".qty > .qty-box > .bigger").click();
11   await page.waitForTimeout(1000);
12   await page.locator(".qty > .qty-box > .bigger").click();
13   await page.waitForTimeout(1000);
14   await expect(page.locator('input[name="cart_quantity\\[\\"]')).toHaveValue("3");
15   await page.locator(".pop-up-close").click();
16   await page.locator("#cart-box").hover();
17   await page.locator("#cart-box").getByRole("link", { name: "Checkout" }).click();
18   await page.locator("#login-email_address").click();
19   await page.locator("#login-email_address").fill("Juana12345@hotmail.com");
20   await page.locator("#login-email_address").press("Tab");
21   await page.locator("#login-password").fill("JuanaJuana12345");
22   await page.getByRole("button", { name: "Log in" }).click();
23   await page.waitForTimeout(1000);
24   await page.getByLabel("£122.58 Total").check();
25   await page.waitForTimeout(1000);
26   await page.getByLabel("Cash on Delivery").check();
27   await page.waitForTimeout(1000);
28   await page.getByRole("button", { name: "Confirm and pay" }).click();
29   await expect(page.locator("#box-36757")).toContainText("We'll send you order status update by email");
30 });
31
```



# Principales fortalezas usando Playwright

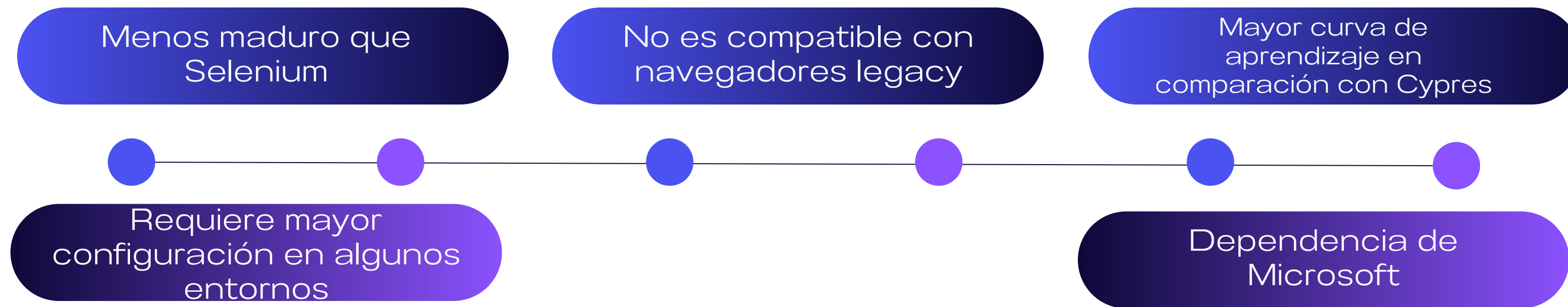
Aquí tenemos una lista de las principales fortalezas en automatización de pruebas, que respaldan el uso de Playwright y definen el rumbo de la propuesta:



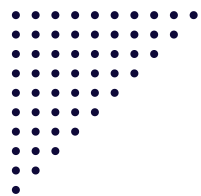


# Principales debilidades usando Playwright

Aquí tenemos una lista de las principales debilidades en automatización de pruebas







# Conclusiones

Las herramientas de control de calidad se utilizan para determinar, medir, analizar y proponer soluciones a los problemas identificados que interfieren con el rendimiento de los procesos de la organización, ayudando a mejorar los indicadores de calidad.

La matriz de Pugh se presenta como una herramienta que nos ha permitido poder evaluar diferentes herramientas para realizar las actividades de Calidad, en este caso la matriz de Pugh nos ofrece un enfoque estructurado para evaluar diferentes alternativas basadas en un conjunto de criterios predefinidos, permitiendo a todos tomar decisiones informadas y objetivas.

En Playwright se posiciona como una herramienta superior para pruebas automatizadas en comparación con Cypress debido a su flexibilidad, velocidad, escalabilidad y cobertura avanzada de navegadores. Los usuarios que optan por Playwright buscan maximizar la eficiencia de sus pruebas en aplicaciones modernas y complejas.

Durante el desarrollo de esta práctica con Visual Studio Code y Playwright, hemos comprobado que esta combinación facilita la creación y ejecución de test cases. Playwright ofrece herramientas visuales útiles, como reportes automáticos, que permiten identificar y analizar errores de manera clara y rápida. Además, su interfaz intuitiva y su capacidad para automatizar pruebas contribuyen a que el proceso de validación sea más eficiente, accesible y comprensible, lo que optimiza el desarrollo y la calidad del software.

