

## Challenge QA Automation

El candidato deberá automatizar un flujo simple pero representativo usando **Playwright + Cucumber + Node.js**, siguiendo buenas prácticas y documentándolo correctamente.

### Escenario a automatizar

Automatizar el siguiente flujo:

1. Navegar a una página web pública que permita interacción simple (ejemplo: <https://www.saucedemo.com/>).
2. Realizar login con usuario estándar:
  - a. username: standard\_user
  - b. password: secret\_sauce
3. Validar que el login fue exitoso (por ejemplo, que aparece la lista de productos).
4. Agregar un producto al carrito.
5. Validar que el carrito muestra 1 ítem.
6. Cerrar sesión.

### Requerimientos mínimos

El candidato deberá:

#### 1. Construir mini framework con:

- **Page Object Model** (PO por lo menos para LoginPage y ProductsPage).
- **Steps de Cucumber** organizados por funcionalidad.
- **Hooks** (Before/After) para inicializar y cerrar navegador.
- Configuración de Playwright (`playwright.config.ts` o JS).
- Selección adecuada de selectores (no usar siempre XPath genérico).
- Esperas explícitas solo cuando sea estrictamente necesario.

#### 2. Entregar:

- Código en un repositorio (GitHub, GitLab, Zip).
- Archivo `README.md` con:
  - Cómo instalar
  - Cómo ejecutar
  - Breve explicación de decisiones técnicas
  - Posibles mejoras

## CRITERIOS DE EVALUACIÓN

### **1. Arquitectura y Buenas Prácticas (40%)**

- Separación correcta de responsabilidades.
- Reutilización (no repetir lógica en steps).
- Extensibilidad del mini framework.
- Limpieza y orden en los directorios.

## **2. Conocimiento Técnico de Playwright (30%)**

- Uso correcto de `page.locator()`.
- Manejo de `waits` vs. `auto-waiting` natural de Playwright.
- Assertions claras.
- Manejo adecuado de fixtures o hooks.

## **3. Calidad del Código (15%)**

- Legibilidad, naming, comentarios.
- Selectores mantenibles.
- Uso adecuado de helpers o utilities.

## **4. Documentación (10%)**

- README claro.
- Explicación concisa de decisiones.
- Cómo ejecutar el proyecto.

## **5. Estabilidad / Evitar flaky behaviour (5%)**

- No usar `waitForTimeout`.
- Uso apropiado de `expect` y `waits` esperados.